

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**SEGMENTAÇÃO DO VENTRÍCULO ESQUERDO EM
IMAGENS ECOCARDIOGRÁFICAS**

JUNIER CAMINHA AMORIM

ORIENTADORA: Prof^ª. Dra. JULIANA FERNANDES CAMAPUM

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.DM-358/08

BRASÍLIA/DF: DEZEMBRO-2008

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENERGIA ELÉTRICA**

**SEGMENTAÇÃO DO VENTRÍCULO ESQUERDO EM
IMAGENS ECOCARDIOGRÁFICAS**

JUNIER CAMINHA AMORIM

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA ELÉTRICA DA FACULDADE DE
TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA
ELÉTRICA.**

APROVADA POR:

**Prof^a JULIANA FERNANDES CAMAPUM, Ph.D. (EnE/UnB)
(ORIENTADORA)**

**Prof RICARDO LOPES DE QUEIROZ, Ph.D. (EnE/UnB)
(EXAMINADOR INTERNO)**

**Prof ANTONIO DE CASTRO NUNO SANTA ROSA, Dr. (CIC/UnB)
(EXAMINADOR EXTERNO)**

BRASÍLIA/DF, 12 DE DEZEMBRO DE 2008.

FICHA CATALOGRÁFICA, REFERÊNCIA BIBLIOGRÁFICA

AMORIM, JUNIER CAMINHA.

Segmentação do Ventrículo Esquerdo em Imagens Ecocardiográficas. [Distrito Federal] 2008.

xv, 108p., 210x297mm (EnE/FT/UnB, Mestre, Engenharia Elétrica, 2008).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

- | | |
|----------------|------------------------|
| 1. Algoritmo | 4. Filtros |
| 2. DICOM | 5. Watershed |
| 3. Segmentação | 6. Ventrículo Esquerdo |
| I. EnE/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

Amorim, J. C. (2008). Segmentação do Ventrículo Esquerdo em Imagens Ecocardiográficas. Dissertação de Mestrado, publicação PPGENE.DM – 358/08. Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 108p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Junier Caminha Amorim.

TÍTULO DA DISSERTAÇÃO: Segmentação do Ventrículo Esquerdo em Imagens Ecocardiográficas.

GRAU: Mestre.

ANO: 2008.

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente com propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta monografia de pós-graduação pode ser reproduzida sem a autorização por escrito do autor.

Junier Caminha Amorim
SGAN 912, Mód. C, Bloco H, apt 15
CEP 70.790-120, Brasília – DF

DEDICATÓRIA

Dedico este trabalho aos meus queridos pais, Flávio Cerqueira Amorim e Maria Zilmar Caminha Amorim, que sob a fé e proteção Divinas, proporcionaram-me todas as condições para prosseguir na árdua e bela caminhada da vida e que, sem tais passos iniciais, seria impossível estar concluindo o presente trabalho. Pela confiança e apoio constantes em todas as minhas decisões, trazendo sempre uma palavra de conforto e incentivo em qualquer situação.

Por acreditarem em Deus e, baseado nisto, terem firmado bases morais e educacionais sólidas, sendo decisivos para a minha formação como homem e profissional.

Ao meu filho, Pedro Henrique, que após chegar em minha vida, tornou-se fonte inspiradora através do seu olhar doce e forte, e do seu sorriso inocente.

AGRADECIMENTOS

À minha orientadora Prof^a. Juliana Fernandes Camapum que, com simplicidade, competência e profissionalismo, me aceitou como orientando e guiou a execução dos trabalhos com paciência, amizade e incentivo constantes, sendo de fundamental importância para a conclusão do referido trabalho. Ensinou-me, através do exemplo, que o conhecimento não deve alimentar a arrogância sobre o próximo mas sim, a capacidade de contribuir com o crescimento daqueles que nos procuram.

Ao meu amigo Helton Garcia pelas orientações e apoio dados que facilitou meu ingresso no curso de mestrado.

À Dra. Roseanne de Fátima Ramos Almeida, Ecocardiografista do Hospital de Base de Brasília, que dedicou horas extras do seu trabalho prestando-me o total apoio em termos técnicos e científicos, propiciando material e conhecimento teórico necessários para a conclusão desta monografia.

À minha amiga Michelle Camilo Guedes, técnica em enfermagem lotada na Pediatria do Hospital de Base de Brasília, por toda a ajuda moral e técnica que me foi dispensada.

E, agradeço a todas as pessoas que diretamente ou indiretamente colaboraram para a conclusão, com êxito, desta dissertação de mestrado.

RESUMO

Este trabalho tem por objetivo apresentar a estrutura e a lógica de um algoritmo aplicado na segmentação do ventrículo esquerdo de ecocardiografias codificadas sob o padrão DICOM (*Digital Imaging and Communications in Medicine*) obtidas através de equipamentos adequados.

Além de identificar fases, melhorias e resultados do algoritmo criado, é realizado um estudo comparativo desses resultados com os obtidos por meio da aplicação de outros algoritmos de mesma finalidade sobre o mesmo conjunto de imagens. Os fatores de comparação tais como o tempo de processamento e a qualidade do resultado são identificados através da comparação das áreas de segmentação entre as imagens tratadas pelo profissional da área de saúde e pelo algoritmo.

Um conjunto de operações morfológicas sobre as imagens foi aplicado, além de algoritmos de corte e preparação das mesmas. Filtros de suavização, tais como o Gaussiano e o de Lee[4], além de sucessivas operações de abertura e fechamento, foram utilizados com o propósito de eliminar ruídos e prepará-las para a segmentação *watershed*.

Neste contexto, deu-se um foco especial às etapas do pré-processamento, pela sua importância em preparar os quadros que serão manipulados pelo algoritmo de segmentação.

ABSTRACT

This work aims at presenting the structure and logic of an algorithm applied to the segmentation of the left ventricle of echocardiographies encoded in the DICOM (Digital Imaging and Communications in Medicine) standard, obtained through appropriate equipment.

Besides identifying stages, improvements and results of the created algorithm, we conducted a comparative study of these results with those obtained through the application of other algorithms with the same purpose on the same set of images. The comparison of factors such as the processing time and result quality are identified by comparing the areas of segmentation between the images processed by the medical professional and the algorithm.

A number of morphological operations on the images was applied, as well as algorithms, cutting and preparing them. Smoothing filters, such as Gaussian and Lee [4], and successive operations of opening and closing, were used for the purpose of eliminating noise and preparing them for watershed segmentation.

In this context, was a particular focus on stages of pre-processing was given, for its importance in preparing the frames wich will be handled by the segmentation algorithm.

SUMÁRIO

1. INTRODUÇÃO.....	16
1.1 - MOTIVAÇÃO	17
1.2 - OBJETIVO	18
1.3 – A DINÂMICA CARDÍACA.....	19
1.4 - APLICAÇÃO MÉDICA	23
1.5 - TRABALHOS RELACIONADOS	26
1.6 - ORGANIZAÇÃO DA DISSERTAÇÃO.....	29
2. O ALGORITMO	31
2.1 - INTRODUÇÃO	31
2.2 – VISÃO GERAL.....	31
2.3 – PREPARAÇÃO DAS IMAGENS	33
2.4 – PRÉ-PROCESSAMENTO.....	34
2.4.1 – Agrupamento das Imagens.....	35
2.4.2 – Recorte das Imagens	37
2.4.3 – Cálculo do Limiar de Otsu	38
2.4.4 – Suavização	38
2.4.5 – Binarização da Imagem.....	40
2.5 – SEGMENTAÇÃO.....	41
2.5.1 – Aplicação da Erosão	41
2.5.2 – Transformada <i>Watershed</i>	44
2.5.2.1 – Gradiente Multiescalar	44
2.5.2.2 – Distância Euclidiana.....	45
3. TECNOLOGIAS UTILIZADAS E ARQUITETURA.....	48
3.1 - INTRODUÇÃO	48
3.2 – TECNOLOGIAS UTILIZADAS	48
3.2.1 – JAVA	48
3.2.2 – MATLAB.....	50
3.2.3 – Eclipse.....	50
3.2.4 – Ferramentas Auxiliares	50
3.3 – CONEXÃO DA APLICAÇÃO JAVA COM O MATLAB	51
3.4 – AMBIENTE DE EXECUÇÃO	52
3.5 – ARQUITETURA IMPLEMENTADA.....	53
3.5.1 – Casos de uso do aplicativo	53
3.5.2 – Diagrama de classes	56
3.5.3 – Diagrama de Componentes	58
4. ANÁLISE COMPARATIVA	61
4.1 – ÍNDICES DE COMPARAÇÃO.....	61
4.2 – SELEÇÃO DAS IMAGENS	62
4.3 – TESTE DO PRÉ-PROCESSAMENTO.....	63
4.4 – TESTE DA SEGMENTAÇÃO	64

4.5 – APLICAÇÃO DOS ÍNDICES DE COMPARAÇÃO NOS RESULTADOS DA SEGMENTAÇÃO.....	67
4.6 – COMPARAÇÃO COM RESULTADOS OBTIDOS POR OUTROS AUTORES	70
5. CONCLUSÃO E TRABALHOS FUTUROS.....	72
REFERÊNCIAS BIBLIOGRÁFICAS	74
ANEXO A.....	77
PADRÕES UTILIZADOS.....	77
A.1 - INTRODUÇÃO	77
A.2 – ARQUIVOS DICOM.....	77
A.2.1 - Histórico	77
A.2.2 - Estrutura.....	78
A.2.3 – Manipulação dos arquivos DICOM	79
A.3 – ECOCARDIOGRAFIA	80
A.3.1 – Introdução	80
A.3.2 – Ecocardiografia Transtorácica	80
A.3.3 – Ecocardiografia Doppler	81
A.3.4 – O Exame Ecocardiográfico	82
A.4 – IMAGENS INDIVIDUAIS.....	83
A.4.1 - Visão Geral	83
A.5 – PROGRAMAÇÃO ORIENTADA A OBJETOS	84
A.5.1 - Visão Geral	84
A.5.2 - Conceitos	84
A.6 – MATLAB	86
A.6.1 - Visão Geral	86
A.6.2 – Manipulação dos Arquivos	86
ANEXO B.....	87
FUNDAMENTOS TEÓRICOS	87
B.1 - INTRODUÇÃO.....	87
B.2 - HISTÓRICO	87
B.3 – IMAGEM DIGITAL.....	88
B.4 – OPERAÇÕES MORFOLÓGICAS	91
B.4.1 – Elemento Estruturante	91
B.4.2 – Dilatação e Erosão.....	92
B.4.3 – Abertura e Fechamento	93
B.4.4 – Afinamento.....	96
B.5 – FILTROS	97
B.5.1 – Filtro de Mediana	97
B.5.2 – Filtro de Média	98
B.5.3 – Filtro Gaussiano	99
B.5.4 – Filtro de Lee	101
B.6 – SEGMENTAÇÃO	101
B.6.1 – Limiarização.....	102
B.6.2 – Limiarização de Otsu.....	104
B.6.3 – Gradiente Morfológico	105
B.6.4 – Watershed.....	106

LISTA DE TABELAS

<i>Tabela 3.1 – Configuração utilizada e configuração mínima da máquina que executa o algoritmo.....</i>	<i>53</i>
<i>Tabela 3.2 – Resumo dos casos de uso.....</i>	<i>54</i>
<i>Tabela 3.3 – Descrição do caso de uso “Abrir Arquivo DICOM”</i>	<i>55</i>
<i>Tabela 3.4 – Notações de relacionamentos.....</i>	<i>57</i>
<i>Tabela 3.5 – Funções das principais classes.....</i>	<i>58</i>
<i>Tabela 3.6 – Descrição do diretório <code>br.com.matlab.files</code>.....</i>	<i>60</i>
<i>Tabela 4.1 – Resultados do Índice de Correlação (IC) para imagens de alta e baixa qualidade.....</i>	<i>67</i>
<i>Tabela 4.2 – Resultados do índice Erro Percentual (EP) para imagens de alta e baixa qualidade</i>	<i>68</i>
<i>Tabela 4.3 – Resultados do índice Somatório do Erro (SE) para imagens de alta e baixa qualidade.....</i>	<i>69</i>
<i>Tabela 4.4 – Índices de comparação obtidos para imagens de alta e baixa qualidade</i>	<i>70</i>
<i>Tabela 4.5 – Índices de comparação para dois algoritmos: Reis[39] e o deste trabalho.....</i>	<i>71</i>
<i>Tabela 4.6 – Índices de comparação para dois algoritmos: Silva[36] e o deste trabalho.....</i>	<i>71</i>

LISTA DE FIGURAS

Figura 1.1 – Ecocardiografia com destaque para o VE.....	17
Figura 1.2 – Visão geral do coração com suas câmaras identificadas.....	20
Figura 1.3 – O coração em corte de quatro câmaras	20
Figura 1.4 – Fluxo sanguíneo no coração e nos grandes vasos, durante o ciclo cardíaco.	23
Figura 1.5 – Planos ecocardiográficos.....	24
Figura 1.6 – Janelas ecocardiográficas.....	25
Figura 1.7 – Ecocardiografias bidimensionais transtorácicas: (a) Imagens de corte apical de 4 câmaras e; (b) De corte paraesternal do VE.	25
Figura 1.8 – Captura de imagens ecocardiográficas: (a) Posicionamento do transdutor para a captura de imagem apical de 4 câmaras; (b) captura de imagem paraesternal transversal VE.	26
Figura 2.1 – Diagrama representativo do algoritmo proposto, identificando as três etapas.	32
Figura 2.2 – Interface gráfica da aplicação com os campos manipulados pelo usuário.....	33
Figura 2.3 – Agrupamento das imagens extraídas do arquivo DICOM.	35
Figura 2.4 – Algumas imagens(quadros) retiradas dos arquivos DICOM antes do agrupamento.....	37
Figura 2.5 – Algumas imagens (quadros) retiradas dos arquivos DICOM após agrupamento.	37
Figura 2.6 – Recorte da imagem: (a) Linha tracejada marcando a região da imagem original extraída pelo aplicativo; (b) Arquivo no formato BITMAP criado apartir da região recortada.....	38
Figura 2.7 – Resultado da Suavização: (a)Imagem original; (b)Imagem suavizada com filtro Gaussiano ($\sigma = 1,5$)	39
Figura 2.8 – Histogramas de análise: (a)Histograma indicando a presença de ruídos na imagem; (b)Histograma de uma imagem mais homogênea.	40
Figura 2.9 – Elementos estruturantes utilizados na erosão: (a)Elemento para imagens de corte apical de 4 câmaras; (b)Elemento para imagens de corte paraesternal transversal.	42
Figura 2.10 – Região de Interesse de uma imagem manipulada pela aplicação.	42
Figura 2.11 – Movimento da válvula mintral e sua influência na segmentação.....	43
Figura 2.12 – Paredes do ventrículo esquerdo: (a) Septal; (b) Apical; e (c) Lateral.....	43
Figura 2.13 – Resultado do Gradiente Multiescalar: (a)Imagem binária do ROI; (b)Resultado do Gradiente sobre o ROI.	45
Figura 2.14 – Distância Euclidiana sobre a ROI.....	46
Figura 2.15 – Segmentação watershed: (a)Imagem resultante; (b)Sobreposição do contorno do VE na imagem do quadro original.	47
Figura 3.1 – Compilação e interpretação de programas escritos em JAVA.....	49
Figura 3.2 – Interação do JAVA com o MATLAB através da API JMatLink.....	51
Figura 3.3 – Diagrama de Casos de Uso.....	54
Figura 3.4 – Diagrama de Classes.....	56
Figura 3.5 – Estrutura de pacotes.....	59
Figura 3.6 – Estrutura de pastas do aplicativo.....	59
Figura 4.1 – Resultados da segmentação após pré-processamento: (a) Sem agrupamento de imagens; (b) Com agrupamento de imagens.	63
Figura 4.2 – Alguns resultados obtidos com o algoritmo proposto ao aplicar 4 erosões.....	64
Figura 4.3– Alguns resultados obtidos com o algoritmo proposto usando o Gradiente Multiescalar seguido de 6 erosões.....	65
Figura 4.4 – Comparação entre a segmentação automática (linha contínua) e a manual (linha pontilhada) utilizando o Gradiente Multiescalar e 6 erosões.....	66
Figura 4.5 – Comparação entre a segmentação semi-automática (linha contínua) e a manual(linha pontilhada) utilizando o filtro de Lee.	66
Figura 4.6 – Comparação entre a segmentação automática de alta qualidade e a manual.....	70
Figura A.1 – O arquivo .dcm com seus quadros e as informações adicionais (metadados).....	79
Figura A.2 – Duas das quatro posições padronizadas para colocar o transdutor transtorácico: janela supra- esternal (b) e janela sub-costal (a)[34].....	82
Figura B.1- Representação de imagens digitais (a) modelo intensidade e (b) indexada.....	89
Figura B.2- Representação de imagens digitais (a) modelo RGB e (b) binária.....	90
Figura B.3 - Exemplos de elementos estruturantes, (a) com conectividade de 4 e (b) com conectividade de 8.	91
Figura B.4 – Exemplo de dilatação da imagem A com o elemento estruturante B	92
Figura B.5 – Exemplo de erosão da imagem A com o elemento estruturante B.....	93
Figura B.6 – Imagem original (a) e após a operação da abertura (b).....	94

Figura B.7 – Imagem original com ruído (a) e após a operação da abertura (b).	94
Figura B.8 – Exemplos de Abertura e fechamento. (a) elemento estruturante, (b) imagem original, (c) abertura e (d) fechamento[22].	95
Figura B.9 - Imagem original (a) e após a operação da fechamento (b).	95
Figura B.10 - Esqueletização através de uma operação de afinamento: (a) elemento estruturante J, (b) elemento estruturante K, (c) esqueletização utilizando os operadores J e K[24].	96
Figura B.11 – Aplicação de máscara 3x3 sobre a imagem para filtro de mediana.	98
Figura B.12 – Aplicação de máscara 3x3 sobre a imagem para filtro de média.	98
Figura B.13 – Filtro de média aplicado sobre imagem com ruído (a), com máscara 3x3, gerando a imagem suavizada (b).	99
Figura B.14 – Forma da distribuição Gaussiana com média zero e desvio σ em 2-D	100
Figura B.15 – Imagem original (a); Resultado da aplicação do Filtro de Lee com mascara 3x3 (b); Resultado da aplicação do Filtro de Lee com mascara 5x5 (c);	101
Figura B.16 – Histograma particionado por 1(a) e por 2(b) limiares.	103
Figura B.17 - Imagem original (a) e após calcular o gradiente(b).	106
Figura B.18 - Linhas divisórias, mínimos e bacias de captação de um relevo topográfico.	106
Figura B.19 - Segmentação por Watershed: a) imagem original, b) vista topográfica, c) e d) duas etapas do enchimento, e) resultado de enchimento adicional, f) início do preenchimento completo de dois vales (foi construído entre estes vales, uma pequena barragem), g) barragens maiores, h) Watershed final[4].	107
Figura B.20 – Imagem original (a) e supersegmentação com watershed (b).	108

LISTA DE SÍMBOLOS, NOMENCLATURAS E ABREVIACÕES

ANSI American National Standards Institute
API Application Programming Interface
PACS Picture Archiving and Communication Systems
ASCII American Standard Code for Information Interchange
DICOM Digital Imaging and Communication in Medicine
GPL General Public License
BMP Bitmap File Format
Bpp Bits por Pixel
HTTP Hypertext Transfer Protocol
ID Identifier
E/S Operação de Entrada/Saída
IEEE Institute of Electrical and Electronics Engineers
ISO International Standards Organization
FSB Front Side Bus
HD Hard Disk
ITU-T International Telecommunications Union – Telecommunications Standardization Sector
CPU Central Processing Unit
TDE Transformada de Distância Euclidiana
JPEG Joint Photographic Experts Group
UML Unified Modeling Language
ECG Eletrocardiograma
VE Ventrículo Esquerdo
SO Sistema Operacional
EP Erro Percentual
SE Somatório dos Erros
IC Índice de Correlação
IA Inteligência Artificial
NEMA National Electrical Manufacturers Association
POO Programação Orientada a Objetos
JVM JAVA Virtual Machine

ROI Region of Interest

QCP Quadros por Ciclo

SM Segmentação Manual

SA Segmentação Semi-Automática

CASE Computer-Aided Software Engineering

GLOSSÁRIO

DICOM File: o arquivo DICOM é um arquivo que é formatado de acordo com a parte PS3.10 do padrão. Ele contém um cabeçalho de informação do meta arquivo (*File Meta Information Header*) seguido por um conjunto de dados DICOM (*DICOM Data Set*), formatado apropriadamente. Informalmente um DICOM *file* pode ser um conjunto de dados (*Data Set*) sem o cabeçalho.

DICOM File Format: o formato do arquivo DICOM é um meio para encapsular em um arquivo o conjunto de dados (*Data Set*) que representa uma Instância SOP de um objeto de informação DICOM (*DICOM Information Object*).

Pixel: abreviatura para *Pixel Element*. É uma representação gráfica de um elemento de imagem, sendo o menor elemento para um dado dispositivo de exibição, ao qual pode-se atribuir uma cor.

Freeware: um tipo de permissão de uso de *software*, na qual o usuário pode utilizar o *software*, porém não tem acesso ao seu código.

General Public License: a GPL é uma Licença Pública GNU que permite que o usuário de um *software* de código aberto possa alterar o código, mas não pode comercializar o que foi desenvolvido utilizando este código. Ela está orientada principalmente para proteger a livre distribuição, modificação e uso de *software*, isto é, ela declara que o *software* protegido por esta licença e livre é protegido de apropriações que restrinjam as liberdades do usuário.

Método: é um conceito utilizado em orientação a objeto e define qual a operação a ser executada sobre determinado objeto.

Open Source: são *softwares* que possuem o código aberto, isto é, o usuário do *software* tem a liberdade de alterá-lo para atender as suas necessidades.

Bug: são defeitos ou deficiências, totais ou parciais, encontrados nos *softwares* ou em parte deles.

Release: é uma versão estável de um *software* que é disponibilizado ao cliente/público para uso.

1. INTRODUÇÃO

As tecnologias ligadas à informática disseminaram-se por todas as áreas do conhecimento humano, facilitando as atividades do dia-a-dia e tornando a vida mais versátil e simples. Mas ao mesmo tempo que ocorre a simplificação das tarefas, a complexidade de se operar os novos equipamentos e as exigências de conhecimentos nas áreas tecnológicas aumentam consideravelmente.

Tais inovações alcançaram, de maneira surpreendente, a área médica, e trouxeram novos recursos que auxiliam consideravelmente o trabalho do médico em diagnosticar uma determinada doença e/ou anomalia. Mas para explorar ao máximo os recursos trazidos por esta evolução tecnológica, o profissional da área de saúde precisa estar atualizado e familiarizado com o manuseio destas ferramentas.

Nesse contexto, busca-se desenvolver ferramentas e tecnologias que auxiliam a intervenção humana nas operações médicas críticas realizadas pelos médicos, além de gerenciar todo o conhecimento e resultados gerados a cada diagnóstico.

Nas áreas ligadas ao tratamento cardíaco, um recurso largamente utilizado para realizar o diagnóstico mais preciso é a ecocardiografia. Ela utiliza as ondas sonoras com frequências superiores a 20 KHz, ultra-som[1], para identificar diversas anomalias ligadas ao coração, tais como o sopro cardíaco, sintomas de palpitação, síncope, infarto do miocárdio, insuficiência cardíaca e anomalias congênitas, viabilizando o diagnóstico precoce e aumentando as possibilidades de tratamento dessas disfunções.

Através das imagens de um ecocardiograma, o médico consegue avaliar o funcionamento do órgão, bem como identificar possíveis anomalias morfológicas e funcionais que representam sinais de doenças mais graves. O fluxo sanguíneo também pode ser analisado, identificando sua direção e intensidade, utilizando o efeito Doppler[2].

Nesse tipo de diagnóstico, a identificação da câmara cardíaca denominada de ventrículo esquerdo – VE assume uma importância significativa, pois, a partir dela, o profissional de saúde pode tirar diversas conclusões a respeito do quadro clínico do paciente. Dimensões, volume e outros dados biométricos desta parte do coração servem de indícios certos de anomalias no funcionamento cardíaco. A Figura 1.1 mostra uma ecocardiografia que exhibe as quatro câmaras cardíacas, com o destaque do ventrículo esquerdo.

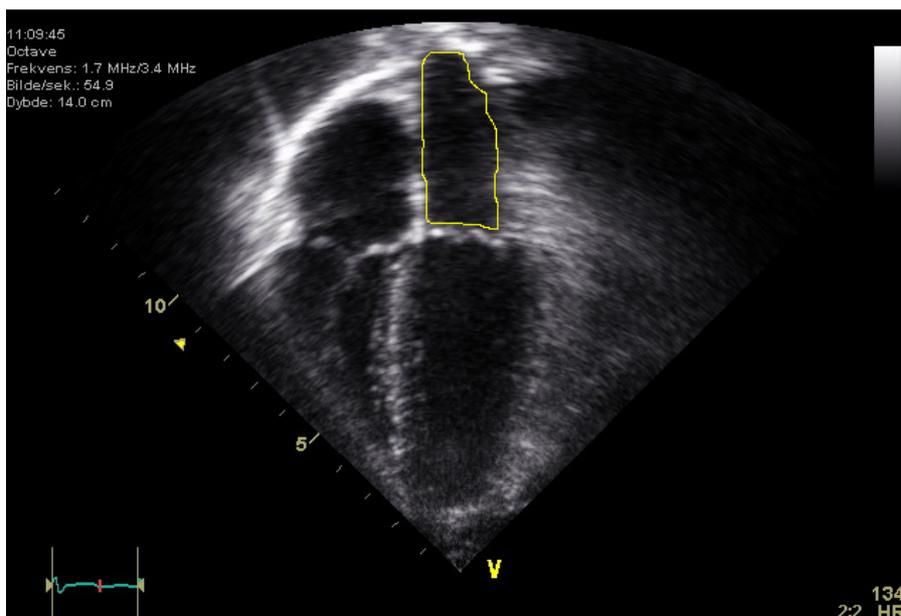


Figura 1.1 – Ecocardiografia com destaque para o VE

As imagens manipuladas estão no formato DICOM (*Digital Imaging and Communication in Medicine*) [3], com extensão “.dcm”. Tal padrão foi criado por uma iniciativa conjunta da Faculdade Americana de Radiologia (*American College of Radiology – ACR*) e da Associação Nacional de Fabricantes Elétricos (*National Electrical Manufacturers Association – NEMA*). Ele define o formato dos arquivos gerados pelos dispositivos que produzem as diversas modalidades de imagens médicas digitais. Além disso, especifica o protocolo de comunicações para a transmissão dessas imagens e informações relacionadas, permitindo a conexão entre dispositivos de hardware e software.

Devido à padronização de formatos, pode-se construir uma aplicação que manipula essas imagens, executa algoritmos de segmentação, e interage com diversos equipamentos conferindo flexibilidade e auxiliando o médico na realização do diagnóstico.

1.1 - MOTIVAÇÃO

A tecnologia existente nas áreas médicas nos traz diversas facilidades e melhorias na precisão dos diagnósticos e tratamentos. Os artefatos computacionais estão cada vez mais presentes, dividindo o espaço com o bisturi e o estetoscópio, auxiliando, e muito, na realização do tratamento de novas e antigas anomalias de saúde. O grande número de

modelos de equipamentos médicos é uma dificuldade, pois carrega, de maneira implícita, a complexidade de padronização.

O padrão DICOM veio para solucionar este problema, pois ele tem como objetivo a padronização dos arquivos de imagens geradas pelos vários fabricantes de equipamentos de imagens médicas. Essas são transmitidas através de uma rede de computadores, permitindo dessa forma o desenvolvimento de sistemas que possam interagir com equipamentos de fabricantes distintos. Esses sistemas são conhecidos como Sistemas de Comunicação e Arquivamento de Imagens – *Picture Archiving and Communication Systems* – PACS.

No caso da análise de ecocardiografias, em que a identificação do ventrículo esquerdo (VE) é de suma importância para o diagnóstico, normalmente, aplica-se a segmentação manual realizada pelo médico com o auxílio do próprio equipamento de captura das imagens cardíacas. Transformar esta etapa em um procedimento que dependa menos da interferência humana é de primordial importância. Quanto maior a necessidade de interação, maior é a possibilidade de erros na tarefa, além de tornar a ferramenta mais limitada.

1.2 - OBJETIVO

Este trabalho tem como objetivo principal a criação de um mecanismo com considerável grau de automatismo para realizar a segmentação do ventrículo esquerdo de ecocardiografias cardíacas. Aplica-se um algoritmo que reúne e adapta técnicas consagradas podendo trazer resultados mais eficientes para esta tarefa de auxílio ao médico.

Além desse, busca-se realizar a comparação dos resultados de diversos algoritmos aplicados sobre o mesmo conjunto de imagens, através da análise da segmentação automática em relação à segmentação realizada pelo profissional de saúde, de maneira a identificar o algoritmo mais eficiente.

Ademais, serão testadas diversas técnicas de pré-processamento das imagens, para realizar a quantificação do impacto desse procedimento sobre o restante do processo de segmentação da imagem, buscando alinhar o melhor pré-processamento com o melhor algoritmo de segmentação.

O objetivo deste trabalho não é criar um algoritmo totalmente inédito, mas sim, identificar partes comuns de funcionamento da maioria dos algoritmos estudados e agrupar, no algoritmo proposto, as partes de melhor desempenho. E nesse contexto, especial atenção foi dispensada ao pré-processamento.

1.3 – A DINÂMICA CARDÍACA

Eleger o órgão mais importante do corpo humano é uma tarefa arriscada e, certamente, resultaria em uma classificação, no mínimo, questionável, já que para funcionar de maneira harmoniosa e perfeita, a participação de cada parte do organismo, por menor que seja, tem uma importância significativa no concerto geral da dinâmica do nosso corpo.

De qualquer forma, o papel do coração nesse cenário é destacado pelo fato de que, anomalias em seu funcionamento, acarretam em sérios danos em outros órgãos, comprometendo o organismo como um todo. Além disso, ele é o responsável por transportar os nutrientes necessários à sobrevivência das células, através do bombeamento sanguíneo, uma das funções básicas do sistema cardiovascular.

O coração situa-se no interior da caixa torácica, em uma cavidade entre os pulmões denominada mediastino, e é formado por dois pares de câmaras, chamadas de átrios e ventrículos, de onde partem e chegam as artérias e veias que conduzem o sangue às diversas partes do organismo. Uma visão geral do órgão, identificando suas câmaras, pode ser vista na Figura 1.2 que mostra uma seção longitudinal do mesmo.

Esse órgão realiza, através de movimentos involuntários, uma média em torno de 70 bpm (batimentos por minuto)[1]. Devido ao fato de que o ventrículo esquerdo é o responsável pela grande circulação durante a diástole, impulsionando o sangue arterial rico em oxigênio e nutrientes para todas as partes do corpo, por intermédio da artéria Aorta, seu esforço de trabalho é maior. Isso o torna, de maneira natural, ligeiramente mais hipertrofiado do que as demais câmaras cardíacas, facilitando o trabalho de segmentação automática.

Quando as paredes do coração são abertas, verifica-se que a cavidade cardíaca apresenta o septo, subdividindo-a em quatro câmaras ou cavidades, como mostra a Figura 1.3. O septo horizontal, denominado de septo átrio-ventricular, divide o coração em duas

porções, superior e inferior. A porção superior apresenta um septo sagital - septo interatrial, que a divide em duas câmaras: átrios direito e esquerdo. Cada átrio possui um apêndice, chamado de aurícula.

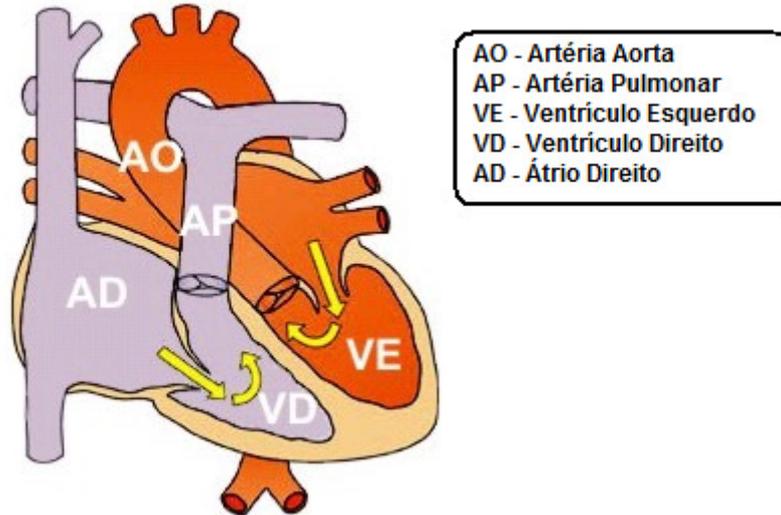


Figura 1.2 – Visão geral do coração com suas câmaras identificadas

A porção inferior apresenta também um septo sagital - septo interventricular, que a divide em duas câmaras: ventrículos direito e esquerdo. O septo átrio-ventricular possui dois orifícios, um à direita e outro à esquerda, chamados de óstios átrio-ventriculares direito e esquerdo. Eles possibilitam a comunicação do átrio direito com o ventrículo direito e do átrio esquerdo com o ventrículo esquerdo.

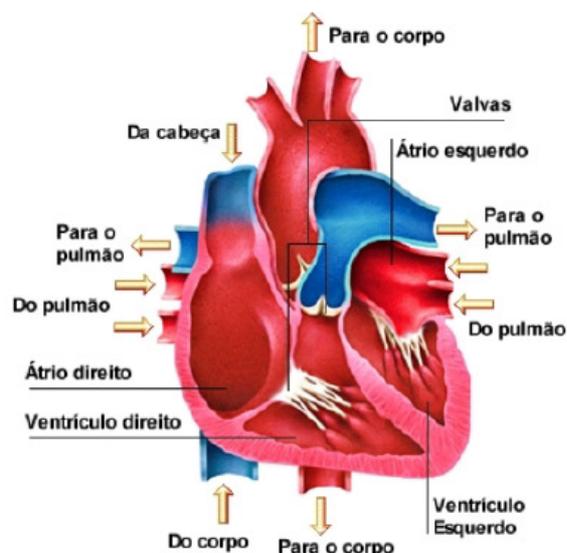


Figura 1.3 – O coração em corte de quatro câmaras

Os óstios átrio-ventriculares são providos de dispositivos que permitem a passagem do sangue somente do átrio para o ventrículo: são as valvas átrio-ventriculares. A valva é formada por uma lâmina de tecido conjuntivo denso, recoberta em ambas as faces pelo endocárdio. Essa lâmina é descontínua, apresentando subdivisões incompletas, que recebem o nome de válvulas ou cúspides. A valva átrio-ventricular direita possui três válvulas e recebe a denominação de valva tricúspide. A valva átrio-ventricular esquerda apresenta duas válvulas e chama-se valva mitral. Quando ocorre a sístole (contração) ventricular, a tensão nessa câmara aumenta consideravelmente, o que poderia provocar uma reversão da valva para o átrio, e, conseqüentemente, refluxo do sangue para ele. Porém, tal fato não ocorre porque cordas tendíneas prendem a valva a músculos papilares, os quais são projeções do miocárdio nas paredes internas do ventrículo.

O ciclo cardíaco pode ser dividido nas seguintes etapas (Figura 1.4):

- **Fim da Diástole:** Ao fim da diástole, as válvulas mitral e tricúspide, entre os átrios e os ventrículos, estão abertas, e as válvulas aórtica e pulmonar estão fechadas. O sangue flui para o coração durante a diástole, enchendo os átrios e ventrículos. A velocidade do enchimento diminui à medida que os ventrículos se distendem, e especialmente quando a frequência cardíaca é baixa. Em seguida, as cúspides das válvulas atrioventriculares – AV deslocam-se em direção à posição de fechamento, a fim de evitar o refluxo sanguíneo, e com esta conformação, a pressão nos ventrículos permanece baixa.
- **Sístole Atrial:** A contração do átrio impulsiona um volume de sangue adicional para os ventrículos, porém cerca de 70% do enchimento ventricular ocorre passivamente durante a diástole. Os orifícios da veia cava superior e inferior e das veias pulmonares são estreitados pela contração do músculo atrial que os circunda. E a inércia do sangue que se movimenta para o coração tende a mantê-lo no seu interior. Entretanto, existe algum refluxo de sangue para as veias durante a sístole atrial.
- **Contração Ventricular Isovolumétrica:** No início da sístole ventricular, as válvulas mitral e tricúspide estão fechadas. Inicialmente, o músculo ventricular se encurta relativamente pouco, mas a pressão intraventricular aumenta rapidamente à medida que o miocárdio pressiona o sangue no interior do ventrículo. Esse período de contração ventricular isovolumétrica dura cerca de 0,05 segundos, até que as pressões nos ventrículos esquerdo e

direito excedam as pressões na aorta e na artéria pulmonar, e as válvulas aórtica e pulmonar se abram. Durante a contração isovolumétrica, as válvulas AV formam um abaulamento em direção aos átrios, causando pequena e rápida elevação da pressão atrial.

- **Ejeção Ventricular:** Quando as válvulas aórtica e pulmonar abrem, começa a fase de ejeção ventricular. Inicialmente a ejeção é rápida, tornando-se mais lenta à medida que a sístole progride. A pressão intraventricular aumenta até um máximo e depois diminui um pouco, antes do término da sístole ventricular. No fim da sístole, a pressão aórtica excede a ventricular, porém, durante curto período, o *momentum* impulsiona o sangue para frente. As válvulas AV são puxadas para baixo pela contração do músculo ventricular, e a pressão atrial diminui. Durante o repouso, cada ventrículo ejeta cerca de 70-90 ml de sangue em cada contração. O volume diastólico final ventricular é de cerca de 130 ml. Assim, no final da sístole, cerca de 50ml de sangue permanece em cada ventrículo (volume sistólico final ventricular) e, a porcentagem do volume diastólico final ventricular que é ejetada em cada contração, a fração de ejeção, é cerca de 65%.

- **Relaxamento Ventricular Isovolumétrico:** Quando o músculo ventricular está completamente contraído, a pressão ventricular em declínio diminui mais rapidamente. Este é o período da protodiástole. Dura cerca de 4 segundos e termina quando o *momentum* do sangue ejetado é superado e as válvulas aórtica e pulmonar fecham. Provocam-se vibrações transitórias no sangue nas paredes dos vasos sanguíneos. Após o fechamento das válvulas, a pressão continua a diminuir rapidamente durante o período de relaxamento ventricular isovolumétrico. Este termina quando a pressão ventricular atinge valor abaixo da pressão atrial e as válvulas AV abrem, permitindo o enchimento dos ventrículos. Inicialmente o enchimento é rápido, para depois diminuir quando a contração ventricular seguinte se aproxima. A pressão atrial continua a se elevar após o final da sístole ventricular até a abertura das válvulas AV. Depois, diminui e aumenta outra vez vagarosamente até a próxima sístole atrial.

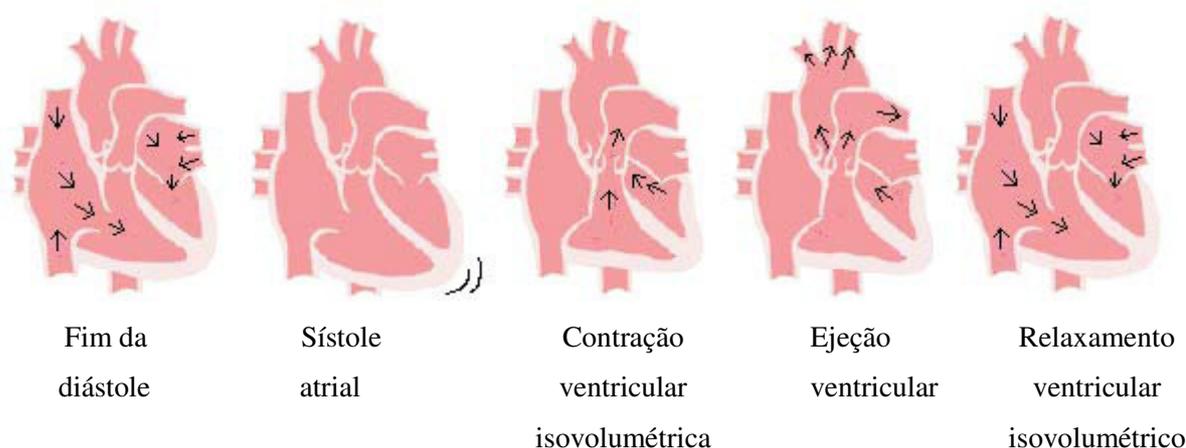


Figura 1.4 – Fluxo sanguíneo no coração e nos grandes vasos, durante o ciclo cardíaco.

1.4 - APLICAÇÃO MÉDICA

A avaliação do volume, da massa e da função mecânica do ventrículo esquerdo, como bomba ou como músculo, requeria o uso de métodos como o cateterismo cardíaco, com os riscos, desconfortos e despesas relacionados a esse procedimento. Além disso, ele é impraticável para o estudo rotineiro ou seriado dos pacientes devido à frequência necessária para a avaliação de efeitos de drogas, momento cirúrgico, técnicas cirúrgicas, ou mesmo para o melhor conhecimento da história natural de certas cardiopatias. A ecocardiografia surge como um método não-invasivo, que utiliza ultra-som (2,0-7,0 MHz), capaz de fornecer dados valiosos a respeito do tamanho e da função sistólica e diastólica do ventrículo esquerdo, sem apresentar os mesmos inconvenientes do cateterismo cardíaco.

A ecocardiografia se divide em transtorácica, transesofágica e fetal. Ela apresenta atualmente diversas modalidades. Iniciou-se com a ecocardiografia unidimensional, seguida pelo método bidimensional. Posteriormente, surgiram as modalidades unidimensional e bidimensional com Doppler convencional, e, finalmente, o Doppler colorido e a ecocardiografia 3D. Este trabalho baseia-se na manipulação de imagens de ecocardiogramas unidimensionais transtorácicas que apresenta a imagem cardíaca projetada em um dos planos mostrados na Figura 1.5 dependendo do posicionamento do equipamento de captura.

Partindo do conhecimento detalhado do funcionamento cardiovascular, potenciais doenças são identificadas através de anomalias em estruturas cardíacas. A segmentação do ecocardiograma transtorácico, aliada às informações obtidas em um eletrocardiograma –

ECG, permitem detectar anomalias tais como a comunicação interventricular[2] e a estenose aórtica. Ambas têm como característica principal a hipertrofia do ventrículo esquerdo e pressão excessiva nas paredes das artérias e veias ligadas ao coração.

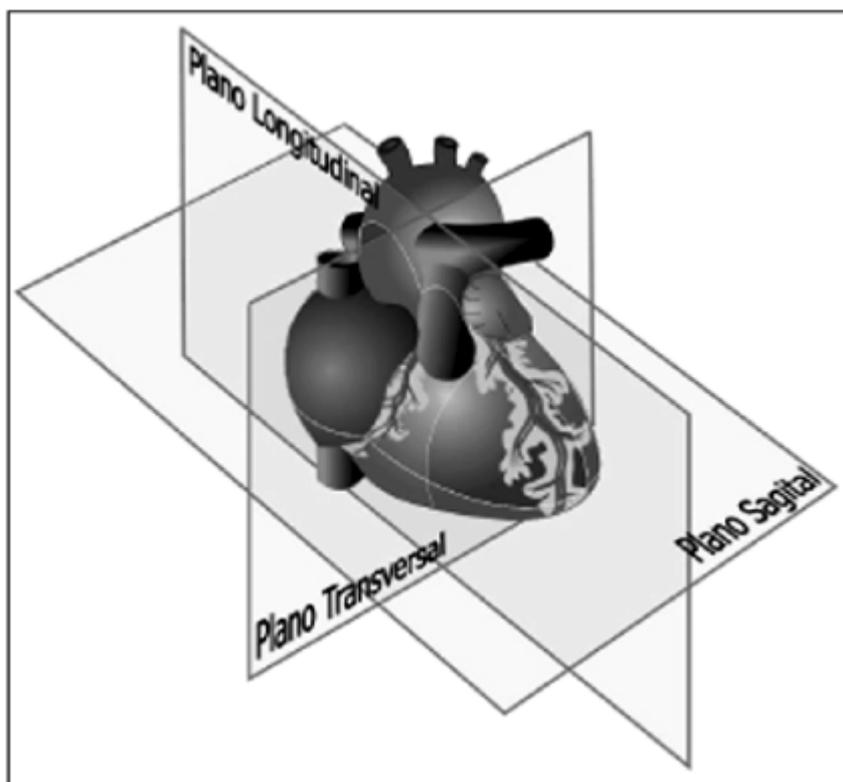


Figura 1.5 – Planos ecocardiográficos.

Tendo como premissa que a segmentação manual, realizada pelo profissional de saúde, não é precisa, pode-se fornecer uma segmentação anterior realizada matematicamente. Essa é obtida através do algoritmo proposto neste trabalho, de modo que venha a ser ajustada e/ou refinada e auxiliada de maneira significativa no fornecimento de um diagnóstico mais preciso.

Esse conjunto de segmentações automáticas obtidas pelo sistema pode ser agrupado em uma base de dados de modo a formar uma biblioteca de informações, que poderá ser consultada nos exames futuros. Com isso, o sistema aprenderia com os seus próprios resultados, incrementando a qualidade das futuras segmentações e, em consequência, poderia auxiliar na identificação de anomalias cardíacas concentradas sobre o VE. Esse aprendizado automático seria possível com a implementação de um sistema especialista que utilize inteligência artificial em sua estrutura[19], e que trabalharia em conjunto com o algoritmo proposto como um sistema autônomo ou como um componente de *software* agregando funcionalidades.

Para o trabalho de segmentação, utilizam-se imagens de cortes de diversas janelas ecocardiográficas dependendo do interesse médico, do equipamento utilizado e do diagnóstico que é buscado. Os nomes destes cortes são de acordo com a janela de captura do ecocardiograma[34]. Na Figura 1.6 podemos ver as posições das diferentes janelas ecocardiográficas, alterando o foco da ressonância.

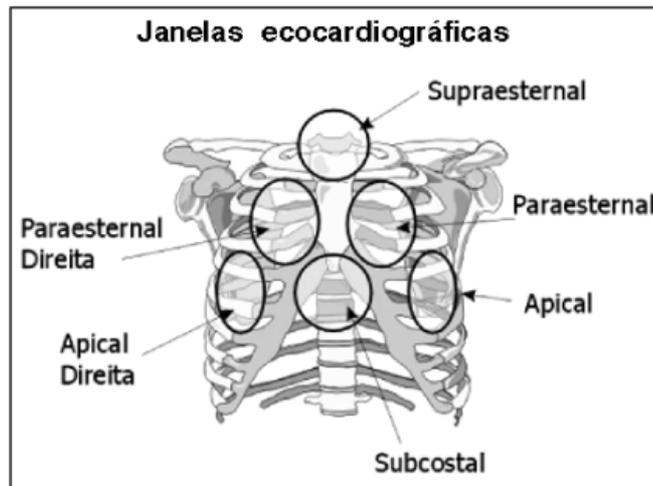
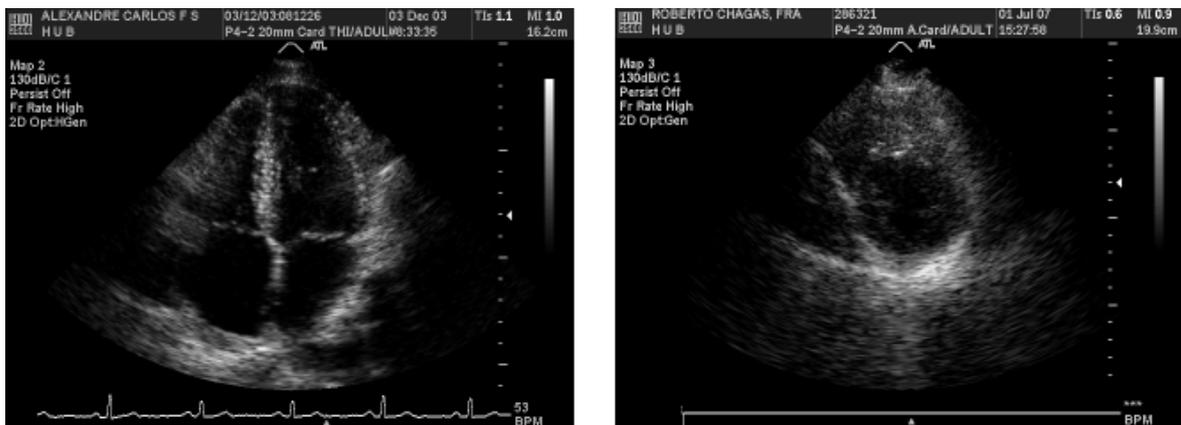


Figura 1.6 – Janelas ecocardiográficas.

Observa-se nas Figuras 1.7(a) e 1.7(b) que a imagem do corte paraesternal fornece uma visão longitudinal do ventrículo esquerdo, posicionado no centro do ecocardiograma, já na imagem de corte apical, o ventrículo esquerdo aparece na parte superior direita da imagem, destacando-se pelo seu tamanho ligeiramente maior do que das demais câmaras cardíacas.



(a)

(b)

Figura 1.7 – Ecocardiografias bidimensionais transtorácicas: (a) Imagens de corte apical de 4 câmaras e; (b) De corte paraesternal do VE.

Quando o foco é alterado, um ângulo de visão diferenciado é mostrado, o que possibilita a obtenção de outras informações sobre o órgão.

O presente trabalho manipula imagens de corte apical (4 câmaras). A Figura 1.8(a) ilustra o posicionamento do transdutor para a obtenção deste tipo de imagem.

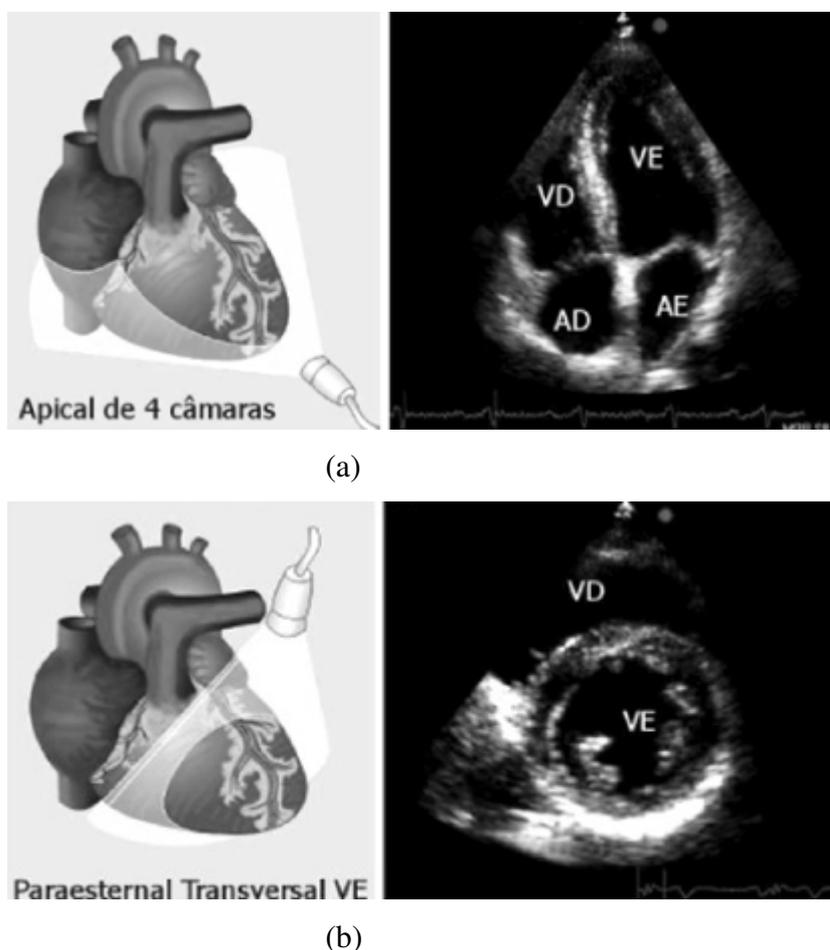


Figura 1.8 – Captura de imagens ecocardiográficas: (a) Posicionamento do transdutor para a captura de imagem apical de 4 câmaras; (b) captura de imagem paraesternal transversal VE.

1.5 - TRABALHOS RELACIONADOS

Através da informatização da medicina, várias atividades anteriormente realizadas exclusivamente pelo médico, hoje são executadas por um computador ou robô de maneira autônoma ou servindo de instrumentos de apoio no aprimoramento do diagnóstico identificado.

Diversos trabalhos vêm sendo realizados na área de segmentação de imagens com foco em imagens médicas e, mais especificamente, na segmentação do VE. O detalhe a se considerar é que a maioria dos trabalhos que conduziram a alguma implementação e/ou produto final, exigiam um considerável grau de interação humana no processo.

Uma restrição existente em todos os trabalhos pesquisados é a especialização do método em imagens de corte paraesternal transversal do ventrículo esquerdo (Eixo Curto – EC) ou apical de 4 câmaras (Eixo Longo – EL), sendo que a qualidade do resultado cai consideravelmente nos algoritmos aplicáveis em ambos os tipos de imagem. Um exemplo dessa especialização é identificado nos algoritmos propostos por Macchiavello[37] e Prashant[3]. O processo de detecção automática do VE utilizado por Prashant[3] foi baseado na busca radial e na suavização temporal, enquanto que Macchiavello[37] utilizou *watershed*. Em ambos, a identificação do VE ocorre sem intervenção humana, mas aplica-se apenas a seqüências de imagens ecocardiográficas paraesternais e de qualidade elevada, previamente selecionadas. O passo mais importante é a identificação do Ponto Central do Ventrículo Esquerdo – *Left Ventricle Center Point* – LVCP com as demais operações baseadas nesse ponto.

Na segmentação do VE propriamente dita, o algoritmo *watershed*[4] é o mais utilizado pela sua simplicidade e eficácia. Porém, antes de aplicar o *watershed* na imagem, filtros e máscaras são usados de modo a reduzir o ruído nessa, evitando supersegmentação. Um limiar muito utilizado nas filtragens é o de *Otsu*[5], definindo os resultados de binarização das imagens.

Interno a este pré-processamento encontram-se os filtros de suavização compostos, principalmente, pelo Gaussiano[3] e pelo de *Lee*[3]. Este último reduz o tipo de ruído mais encontrado, o *speckle*[6]. Comparações de resultados obtidos com diferentes filtros indicaram que o custo computacional da operação é muito elevado para alguns filtros, tais como o de Lee, e o ganho nos resultados é muito reduzido.

Os principais trabalhos existentes tratam o pré-processamento das imagens, mas sem dar a devida importância a essa etapa. Um pré-processamento bem realizado elimina a exigência de operações mais elaboradas nos passos seguintes, reduzindo o custo de processamento e melhorando o resultado final da segmentação. Nesse primeiro tratamento, o alvo é o ruído existente nas imagens de ecocardiografia, que causa distorções na segmentação por *watershed*[4] e no cálculo do gradiente.

No trabalho de Chai[7], a imagem foi decomposta através de *wavelets* sucessivas após operações de filtragem morfológica multi-escalar e geodésicas. Assim, o ruído foi

eliminado para evitar a supersegmentação na aplicação do *watershed*. A inserção de *wavelets* no algoritmo proposto não proporcionou uma melhora significativa na precisão da segmentação automática, causando ainda, uma sobrecarga de processamento. Tais comparações serão realizadas na seção 4.5 deste trabalho. Essa sobrecarga computacional pode ser reduzida através do uso de componentes de software prontos que realizam essas operações de maneira otimizada.

Em nenhum dos trabalhos pesquisados e bibliografias consultadas, fazia-se referência ao modo de aquisição das imagens. Baseado no fato de que os arquivos no formato DICOM são um agrupamento de imagens sequenciais que retratam uma janela de tempo do funcionamento cardíaco, a seleção das imagens, ou combinação dessas, que serão usadas na segmentação pode conduzir a resultados distintos. Esse conjunto de imagens diferentes umas das outras precisa ser agrupado de modo sistemático, levando em consideração a sua semelhança *intraframe*, ou distância *interframe*[3], reunindo as características favoráveis e eliminando parte dos ruídos e/ou falhas.

A maioria dos artigos pesquisados, implementam apenas provas de conceito, utilizando recursos específicos da plataforma Matlab[8], e ficam limitados quanto à utilização integrada com outros softwares e/ou equipamentos.

As implementações de diversos recursos de complexidade matemática maior, tais como a decomposição por *wavelet*, a aplicação do filtro de Lee e o cálculo do limiar de Otsu para binarização de imagens, em outras linguagens de programação, apresentaram consideráveis diferenças de desempenho e qualidade dos resultados.

Um trabalho de segmentação de imagens por *watershed* implementado em JAVA[9][10] e testado em imagens de boa qualidade consumiu muito processamento e tempo de execução. O programa fez uso da biblioteca JAVA de Manipulação de Imagens – *Java Advanced Imaging API* – JAI[11] que otimiza os cálculos realizados, mas como a linguagem é de alto nível e orientada a objetos, o processamento matemático da máquina ficou comprometido pois as operações matemáticas a serem realizadas precisam ser acessadas através de comandos mais complexos que exigem a execução de inúmeros passos intermediários. Entretanto, apesar de buscar a implementação de um algoritmo que realize a segmentação do ventrículo esquerdo, nota-se uma deficiência na comparação dos diversos pré-processamentos e na capacidade desse software ser acoplado ou reutilizado.

Os trabalhos de Cheng[5], Chai[7], Peccini[9] e Shin[20] forneceram subsídios para que houvesse o aperfeiçoamento do processo de segmentação e a busca de um componente de software com um maior grau de reusabilidade e escalabilidade[12].

Assim, este trabalho busca gerar um componente de software que realize, de forma automatizada, a segmentação do VE, selecionando e agrupando as imagens mais adequadas extraídas dos arquivos no padrão DICOM. Tal componente será composto pela junção e adaptação de diversos passos de algoritmos consagrados e a inserção de operações novas de maneira a otimizar os resultados obtidos. Serão realizadas comparações desses com trabalhos já realizados neste centro universitário.

1.6 - ORGANIZAÇÃO DA DISSERTAÇÃO

O presente trabalho está dividido e organizado de acordo com os seguintes capítulos:

CAPÍTULO 1: INTRODUÇÃO

Esse capítulo apresenta a motivação para a realização deste trabalho, seus objetivos, a sua aplicação na medicina, os trabalhos relacionados com ele e a organização geral da dissertação.

CAPÍTULO 2: O ALGORITMO

Esse capítulo apresenta a descrição do algoritmo proposto, tratando as suas diversas etapas, especificando as operações realizadas bem como os elementos de entrada e os resultados de cada processo.

Será feito uso de fluxogramas para esquematizar as diferentes etapas do algoritmo, detalhando as operações realizadas dentro de cada caixa do diagrama.

CAPÍTULO 3: ARQUITETURA E TECNOLOGIAS UTILIZADAS

Esse capítulo apresenta a arquitetura do *software* implementado, identificando seus componentes e módulos internos, as tecnologias utilizadas em sua implementação e a utilização de frameworks e componentes de terceiros, tais como o JMatLink[13], que realiza a comunicação entre a aplicação JAVA e o núcleo do MATLAB.

O algoritmo proposto foi implementado utilizando-se o paradigma da orientação a objetos[14], dividindo-se os diversos módulos em camadas de modo a proporcionar

elevado grau de manutenibilidade e reusabilidade do software como um todo ou de apenas parte dele.

CAPÍTULO 4: ANÁLISE COMPARATIVA

Esse capítulo apresenta uma comparação dos resultados obtidos com a aplicação de alguns trabalhos implementados, salientando a influência do pré-processamento nesses resultados.

Os intercâmbios dos diversos pré-processamentos podem gerar resultados distintos para um mesmo conjunto de imagens, e essa diferença auxilia na identificação do melhor conjunto de etapas para a realização de uma segmentação mais aprimorada.

CAPÍTULO 5: CONCLUSÕES E TRABALHOS FUTUROS

Nesse capítulo, mostram-se as conclusões obtidas com a realização deste trabalho. E, apresentam-se sugestões e/ou planejamento de realização de trabalhos futuros de modo a aperfeiçoá-lo ou incrementá-lo através da integração com outros processos.

REFERÊNCIA BIBLIOGRÁFICA

ANEXO A: PADRÕES UTILIZADOS

Essa parte do trabalho apresenta os padrões utilizados no projeto no que diz respeito a formatos de arquivos e imagens, para escrita e leitura, bem como as ferramentas de apoio utilizadas no projeto.

ANEXO B: FUNDAMENTOS TEÓRICOS

Nesse anexo tratamos de toda a teoria necessária para a realização deste trabalho. Os algoritmos, métodos e processos são detalhados e seus pontos críticos são evidenciados.

As justificativas das escolhas feitas neste trabalho são apresentadas de maneira científica, descrevendo as tarefas e adaptações realizadas, calcadas em análises e pesquisas.

2. O ALGORITMO

2.1 - INTRODUÇÃO

Os vários estudos realizados na área de Processamento Digital de Imagens apontam soluções que utilizam algumas das técnicas que são apresentadas nos anexos deste trabalho, mas não focam na problemática do pré-processamento e o quanto isso pode influenciar no resultado final.

Com um bom resultado no pré-processamento, pode-se aumentar o ganho da segmentação em até 35% [4], e esta diferença pode ser facilmente identificada ao analisar a saída dos algoritmos de segmentação após intercambiar seus módulos de pré-processamento.

Esse tratamento prévio no algoritmo proposto procurou aproveitar as características dos batimentos cardíacos, no que diz respeito às imagens capturadas. Explorando a repetição dos movimentos de diástole e sístole do coração [26], pode-se obter imagens com um grau maior de similaridade em posições iguais dos ciclos cardíacos. A semelhança contribui para a obtenção do resultado de junção e seleção de imagens mais confiáveis, otimizando o pré-processamento.

A ênfase dada nessa preparação prévia tem como objetivo chegar à etapa de segmentação com um grupo de imagens que reúna características favoráveis à aplicação do algoritmo *watershed*.

2.2 – VISÃO GERAL

O conjunto de passos realizados pelo algoritmo proposto é mostrado na Figura 2.1 e dá uma visão geral da manipulação das imagens. Sendo o ventrículo esquerdo o alvo principal da segmentação, concentramos as operações de segmentação sobre a área de maior probabilidade de encontrá-lo nas imagens cardíacas obtidas no plano sagital (Figura 1.6). Essa estratégia contribuiu para isolar a influência ruim da qualidade de visualização das demais câmaras cardíacas sobre a segmentação do ventrículo esquerdo.

O algoritmo está dividido em três etapas: preparação das imagens, pré-processamento e segmentação. Os tópicos a seguir detalharão cada uma dessas etapas.

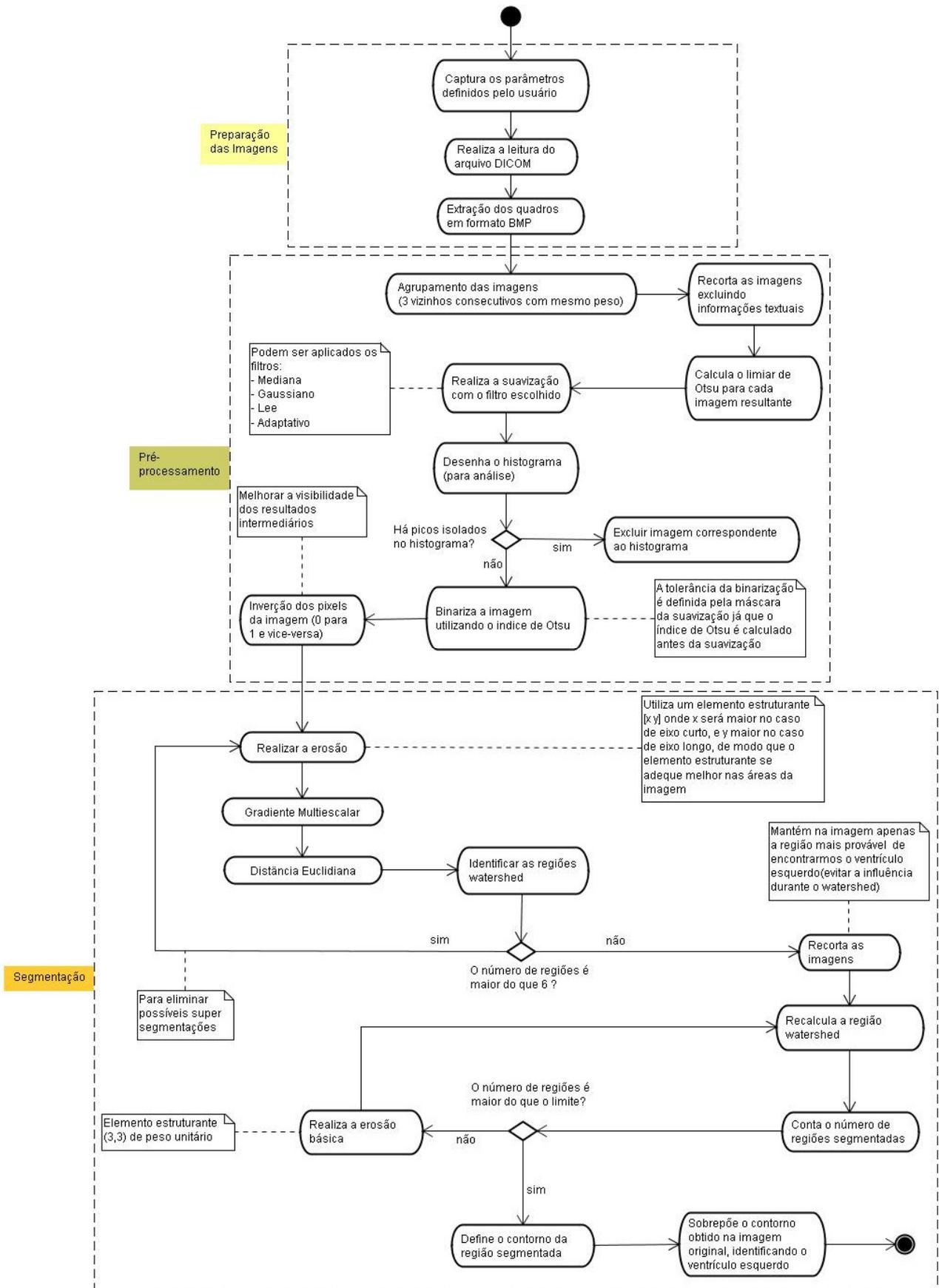


Figura 2.1 – Diagrama representativo do algoritmo proposto, identificando as três etapas.

2.3 – PREPARAÇÃO DAS IMAGENS

Inicialmente, a aplicação captura os parâmetros que foram definidos pelo usuário na interação com sua interface. A Figura 2.2 mostra a interface da aplicação onde estão presentes os campos que podem ser alterados pelo usuário e que influenciam na segmentação das imagens, além dos botões de execução de tarefas.

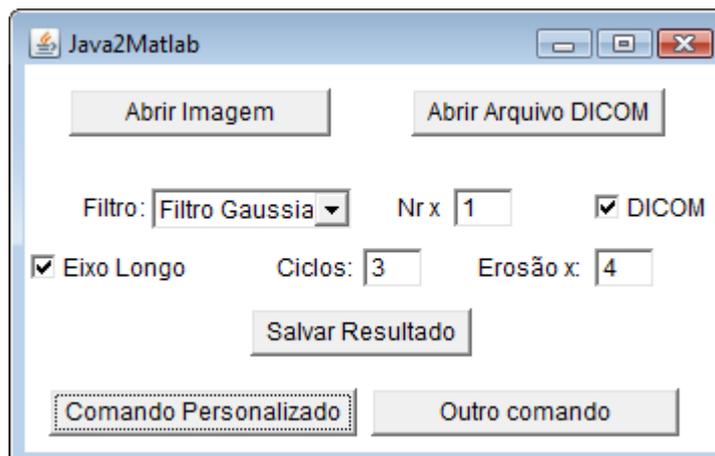


Figura 2.2 – Interface gráfica da aplicação com os campos manipulados pelo usuário.

O algoritmo implementado é capaz de manipular tanto imagens de corte apical de quatro câmaras quanto imagens de corte paraesternal transversal do ventrículo esquerdo, através da adequação das dimensões dos elementos estruturantes utilizados nas operações morfológicas básicas. Para que isto ocorra, a marcação “Eixo Longo” na interface informa que serão imagens paraesternais.

Apesar da aplicação aceitar imagens de corte paraesternal, o seu desempenho é significativamente melhor em imagens de corte apical de quatro câmaras. Essas foram o objeto de estudo do presente trabalho e sobre o qual foram feitos ajustes visando a otimização da segmentação realizada.

Os diversos elementos que compõem a Figura 2.2 são: o campo “Filtro” define o tipo de filtro a ser utilizado; o campo “Nr” informa a quantidade de repetições da aplicação do filtro; a marcação “DICOM” informa à aplicação que o arquivo a ser operado está no padrão DICOM; o campo “Ciclos” informa quantos ciclos cardíacos a seqüência de imagens que compõem o arquivo DICOM engloba; por fim, o campo “Erosão x” informa quantas vezes a operação de erosão deve ser aplicada sobre as imagens.

Ao clicar em “Abrir Arquivo DICOM”, o usuário poderá escolher o arquivo (.dcm) nos diretórios locais e a aplicação realizará a leitura do mesmo, identificando os dados

constantes no cabeçalho do arquivo, e extrai cada um dos quadros que o compõe salvando-os no formato BITMAP – BMP.

Imagens BITMAP podem ser manipuladas diretamente, através do botão “Abrir Imagem”. O botão “Salvar Resultado” possibilita salvar os resultados em outro local que não o pré-estabelecido pela aplicação. Os botões “Outro Comando” e “Comando Personalizado” permitem ao usuário escolher um comando da lista implementada pela aplicação ou digitar um comando pertencente ao MATLAB diretamente na aplicação, respectivamente.

Dessas funcionalidades, a principal é a “Abrir Arquivo DICOM”, pois a partir dela todo o processo de segmentação é iniciado automaticamente, ocorrendo a execução de todas as fases citadas na Figura 2.1.

Para os arquivos DICOM utilizados neste trabalho, a extração resultou em noventa quadros consecutivos que englobam três ciclos cardíacos, com suas diástoles e sístoles, e cada um desses quadros foi convertido para o formato BMP com representação de 24 *bits* por *pixel* - *bpp* (24 *bpp*).

2.4 – PRÉ-PROCESSAMENTO

Esta etapa é composta pelas seguintes fases: agrupamento das imagens consecutivas, em posições simétricas, nos diferentes ciclos; recorte das imagens; cálculo do limiar de Otsu; aplicação da suavização sobre cada uma das imagens; desenho do histograma; binarização e inversão dos *pixels*. A seguir será comentada cada uma dessas fases.

O principal ganho alcançado pelo algoritmo foi o bom grau de eficiência obtido nas etapas que compõem o pré-processamento. Nessa fase do algoritmo, realiza-se o agrupamento dos diversos quadros extraídos do arquivo DICOM compondo uma única imagem. Esse agrupamento será apresentado na seção a seguir.

A importância do pré-processamento torna-se evidente à medida que a exclusão de ruídos e identificação adequada de bordas proporcionam uma aplicação do algoritmo *watershed* adequada reduzindo a probabilidade de ocorrer uma supersegmentação.

2.4.1 – Agrupamento das Imagens

A preparação das imagens, quer sejam agrupadas ou isoladas, para realizar o pré-processamento é fundamental para o sucesso das etapas posteriores do algoritmo. Partindo dessa premissa, o agrupamento das imagens extraídas dos arquivos DICOM explorou a alta probabilidade de simetria destas em momentos equivalentes nos diversos ciclos do batimento cardíaco.

Os arquivos DICOM utilizados agrupam 90 quadros consecutivos que englobam, aproximadamente, 3 ciclos cardíacos compostos de uma sístole e uma diástole. Nestes casos, o número de quadros por ciclo – QPC será igual a 30. A quantidade de ciclos está parametrizada no sistema e pode ser alterada através da interface com o usuário, de acordo com as características dos arquivos DICOM manipulados.

O agrupamento é realizado através da fusão de 9 imagens, para o caso de 3 ciclos cardíacos, sendo 3 imagens consecutivas de cada ciclo (veja Figura 2.3).

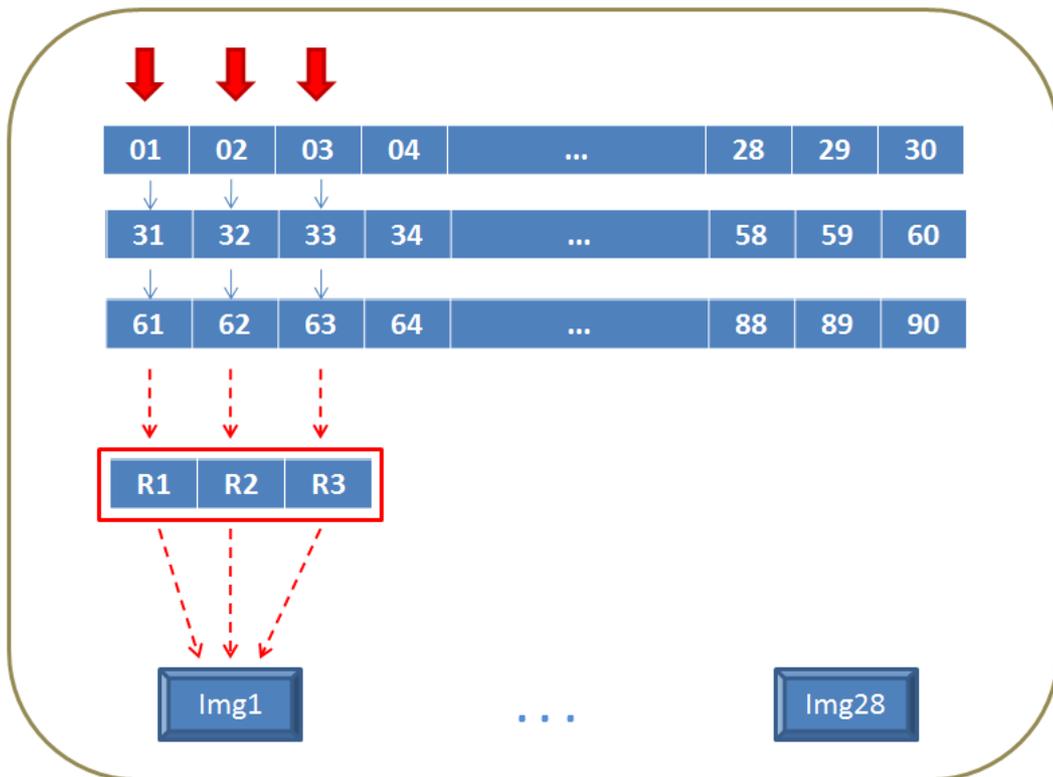


Figura 2.3 – Agrupamento das imagens extraídas do arquivo DICOM.

Assim, o agrupamento busca as imagens na posição k de cada ciclo, tomando como resultado os *pixels* correspondentes com maior nível de cinza. Essa etapa é repetida no quadro vizinho antecedente e posterior e, ao final, capturam-se os *pixels* de maior

intensidade dentre os de mesma posição nas três imagens obtidas, resultando na imagem final deste agrupamento.

Em um arquivo contendo 90 quadros, através deste processo obtém-se 28 imagens finais, já que as imagens nos extremos (índices 1 e 90) são excluídas com o agrupamento dos quadros 1,2 e 3, e 88,89 e 90, respectivamente, gerando as resultantes *img1* e *img28*.

As Equações 2.1, 2.2 e 2.3 representam a obtenção das imagens resultantes *R1*, *R2* e *R3* da Figura 2.3. O número de quadros vizinhos, igual a 3, foi heurísticamente adotado com base em testes feitos com o método apresentado por *Cheng*[5]. A adoção de um número maior de quadros vizinhos causa um borramento na imagem resultante já que a diferença entre as imagens torna-se cada vez maior com o maior distanciamento do quadro central do agrupamento.

$$f_{R1_k}(x, y) = \max\{f_{k-1}(x, y), f_{k-1+QPC}(x, y), f_{k-1+2.QPC}(x, y)\} \text{ para } k = 2,3,\dots,(QPC - 1) \quad (2.1)$$

$$f_{R2_k}(x, y) = \max\{f_k(x, y), f_{k+QPC}(x, y), f_{k+2.QPC}(x, y)\} \text{ para } k = 2,3,\dots,(QPC - 1) \quad (2.2)$$

$$f_{R3_k}(x, y) = \max\{f_{k+1}(x, y), f_{k+1+QPC}(x, y), f_{k+1+2.QPC}(x, y)\} \text{ para } k = 2,3,\dots,(QPC - 1) \quad (2.3)$$

Sendo: QPC = Número de Quadros por Ciclo;

k = Índice que representa a posição da imagem (quadro) dentro de um ciclo;

O arquivo resultante final *Img1*, apresentado na Figura 2.3, é obtido através da Equação 2.4 representado por $f_{R_k}(x, y)$.

$$f_{R_k}(x, y) = \max\{f_{R1_k}(x, y), f_{R2_k}(x, y), f_{R3_k}(x, y)\} \text{ para } k = 2,3,\dots,(QPC - 1) \quad (2.4)$$

A Figura 2.4 mostra duas imagens retiradas de um dos arquivos DICOM antes do agrupamento apresentado nesta seção, e a Figura 2.5 mostra o resultado do agrupamento das imagens desse mesmo arquivo DICOM. Nota-se o ganho na identificação das paredes cardíacas.

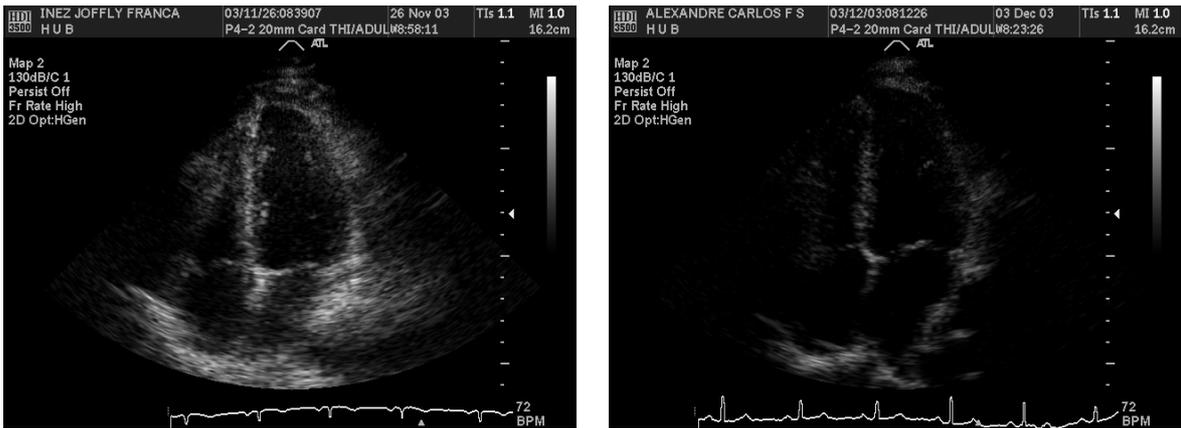


Figura 2.4 – Algumas imagens(quadros) retiradas dos arquivos DICOM antes do agrupamento.

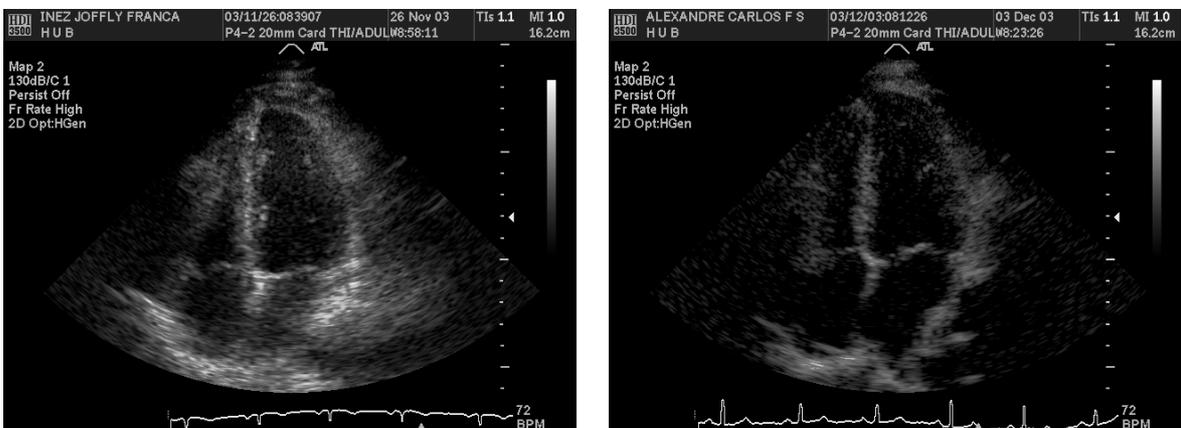


Figura 2.5 – Algumas imagens (*quadros*) retiradas dos arquivos DICOM após agrupamento.

2.4.2 – Recorte das Imagens

Os quadros que compõem o arquivo DICOM carregam em sua estrutura, além da imagem ecocardiográfica, outras informações tais como dados do paciente e do exame ou diagnóstico. Essas informações são escritas nas imagens próximo de suas bordas, e esses símbolos poderiam influenciar, negativamente, nas operações matemáticas.

Antes de realizar qualquer operação sobre as imagens, é necessário recortá-las de modo a retirar essas informações. Tais informações são textos inseridos em modo gráfico nos quadros que compõem o arquivo DICOM. As coordenadas X e Y de corte das imagens, tanto as de corte apical quanto as de corte paraesternal, ficam parametrizadas pois o posicionamento dessas informações é padronizado. Esse corte é realizado numa área de

390x390 pixels. A região de corte na imagem original é mostrada na Figura 2.6 através da linha vermelha pontilhada, feita manualmente, para limitar essa área.

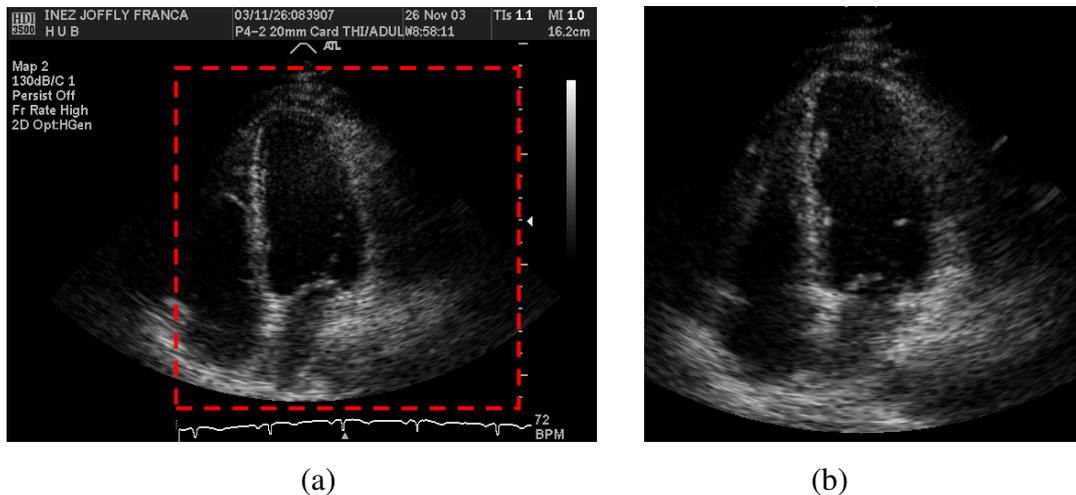


Figura 2.6 – Recorte da imagem: (a) Linha tracejada marcando a região da imagem original extraída pelo aplicativo; (b) Arquivo no formato BITMAP criado a partir da região recortada.

2.4.3 – Cálculo do Limiar de Otsu

Para cada imagem criada a partir do corte apresentado anteriormente, é calculado o Limiar de Otsu, apresentado e detalhado na seção B.6.2, que será utilizado na operação de binarização dessas imagens (*thresholding*). Esses índices são armazenados em uma estrutura vetorial, conservando a ligação à imagem que o originou.

Esse tipo de limiar foi escolhido por ser de simples implementação e por fornecer um resultado mais independente de variações locais. Isso porque seu cálculo é baseado em propriedades globais, conferindo estabilidade e constância à medida tomada.

2.4.4 – Suavização

Para esta fase, a aplicação utiliza o parâmetro definido pelo usuário na interface que escolhe o tipo de filtro que será aplicado dentre os disponíveis (Gaussiano, de Lee, de Mediana ou Customizado[5]). O objetivo é eliminar ruídos das imagens, de modo que a binarização não inclua, nas imagens binárias, regiões e *pixels* correspondentes a esses ruídos.

Para cada um dos filtros é utilizada uma máscara adequada. Para o Filtro Gaussiano, utilizou-se uma máscara de dimensão cinco e desvio padrão 1,5 (σ). O Filtro de Lee calcula o valor de um determinado *pixel* baseado na análise de seus vizinhos, de acordo com o explicado na seção B.5.4. O Filtro de Mediana realiza o cálculo da mediana dos *pixels* vizinhos com uma máscara unitária de dimensão igual a 5% da dimensão da imagem original.

Já o Filtro Adaptativo, proposto por Cheng[5] utiliza duas máscaras, sendo a maior igual a 5% da dimensão da imagem e a menor igual à metade da maior, sendo que a máscara maior é aplicada nos *pixels* cujo nível de cinza é maior que o valor do Limiar de Otsu e a máscara menor sobre os demais *pixels*.

Dos filtros utilizados, o que apresentou a melhor relação custo, no que se refere a tempo de processamento, *versus* benefício ou qualidade do resultado foi o Gaussiano, sendo este o utilizado durante o ajuste fino do algoritmo. A Figura 2.7 mostra o resultado obtido com a aplicação desse tipo de filtro.

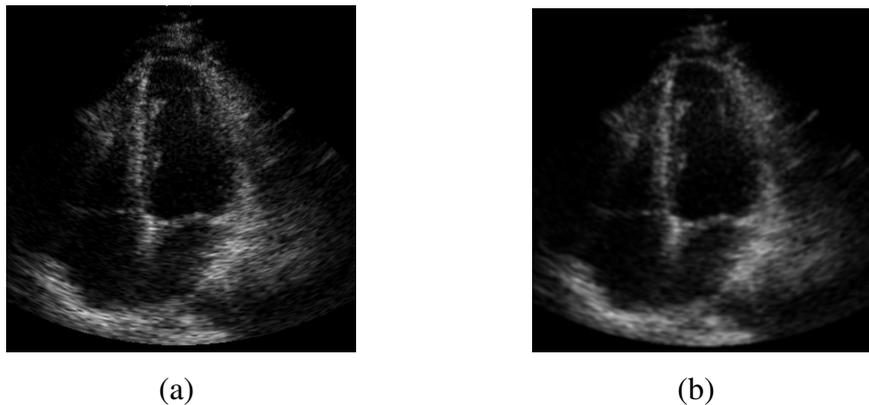


Figura 2.7 – Resultado da Suavização: (a)Imagem original; (b)Imagem suavizada com filtro Gaussiano ($\sigma = 1,5$)

Para algumas imagens, o filtro de Lee trouxe melhores resultados, apesar do maior tempo de processamento necessário. Isso foi devido ao fato do ruído aleatório presente neste tipo de imagem ser muito semelhante ao ruído *speckle*.

Para fins comparativos, durante o processamento e após as filtrações das imagens, o programa dá a opção de exibir o histograma. Através de uma análise da sua distribuição, pode-se identificar imagens que não apresentarão bons resultados na segmentação, sendo previamente excluídas do rol de imagens a serem trabalhadas. No histograma da Figura 2.8(a), podemos identificar picos que representam uma quantidade excessiva de *pixels* na imagem que apresentam os valores intermediários de nível de cinza, identificando forte

indício de ruídos significativos. Essas imagens, consideradas de baixa qualidade, são excluídas antes do pré-processamento.

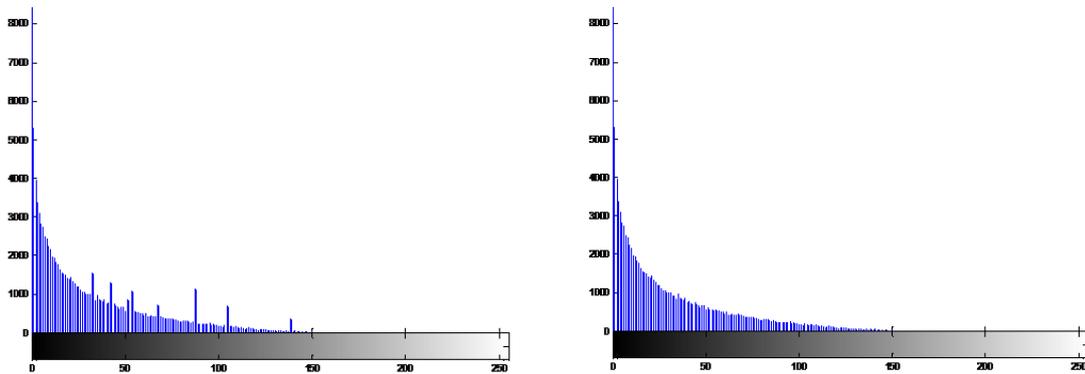


Figura 2.8 – Histogramas de análise: (a)Histograma indicando a presença de ruídos na imagem; (b)Histograma de uma imagem mais homogênea.

2.4.5 – Binarização da Imagem

Nesta fase a imagem digital $f(x,y)$ em nível de cinza, quantizada em 24 bits, é binarizada de acordo com o Limiar de Otsu (l) calculado antes da suavização, de acordo com a regra apresentada pela Equação 2.5. O uso dos limiares anteriores à suavização termina por aumentar a tolerância do *thresholding*, garante melhores resultados na binarização e transfere para a suavização o ajuste fino do nível destes limiares.

O resultado dessa binarização não é salvo em arquivo de modo a agilizar as fases seguintes evitando operações extras de Entrada/Saída – E/S.

$$f(x, y) = \begin{cases} 1, & \text{se } f(x, y) \geq l \\ 0, & \text{se } f(x, y) < l \end{cases} \quad (2.5)$$

Para melhorar a visualização dos resultados finais e da lógica matemática utilizada, inverte-se os valores dos *pixels* de modo a dar destaque à região do ventrículo esquerdo, deixando-o com valor 1 (branco), melhorando a visibilidade.

2.5 – SEGMENTAÇÃO

Nesta etapa é realizado um conjunto de operações a fim de identificar o ventrículo esquerdo. A fase de erosão da imagem foi colocada na etapa de segmentação pelo fato de trabalhar em conjunto com a transformada *watershed* para identificar o número de regiões existentes na imagem.

2.5.1 – Aplicação da Erosão

Para evitar uma super-segmentação da imagem ao aplicar o *watershed*, realizam-se, de maneira iterativa e automática, erosões sucessivas de modo a eliminar ruídos e deformações na imagem. O ponto de parada destas interações é atingir um número de regiões distintas, obtido com a transformada *watershed*, que seja compatível com o resultado desejado. Este resultado esperado deverá ser menor do que 6 pois para uma segmentação ideal teríamos que encontrar quatro regiões correspondentes às câmaras cardíacas[27] e mais uma correspondente à região externa envoltória. Essa operação é realizada sobre cada quadro extraído do arquivo DICOM, excluindo-se as imagens que não apresentam resultados aproveitáveis.

A erosão faz uso de dois elementos estruturantes distintos dependendo da marcação ou não, pelo usuário, da caixa “Eixo Longo” na interface gráfica da aplicação (Figura 2.2). Para o caso de imagens apicais de quatro câmaras, o elemento estruturante apresenta maior extensão na vertical com mais linhas do que colunas. Já no caso de imagens de corte paraesternal transversal do ventrículo esquerdo, o elemento estruturante apresenta mais colunas do que linhas, sendo mais extenso na horizontal. Essa variação torna-se útil na adequação do elemento dentro das regiões que caracterizam as câmaras cardíacas.

A Figura 2.9 mostra as dimensões dos elementos estruturantes para os dois casos citados. No cálculo da dimensão variável, utiliza-se 2,5% da largura ($x_{máx}$) ou 2,5% da altura ($y_{máx}$) da imagem original $f(x,y)$, para imagens de corte paraesternal e apical, respectivamente. O número máximo de linhas e colunas dos elementos estruturantes utilizados estão representados por n e m , respectivamente.

Após a erosão concluída, recortamos a imagem binarizada de modo a identificar a Região de Interesse – Region Of Interest – ROI. Essa região engloba a menor área de mais alta probabilidade de encontrarmos o ventrículo esquerdo. Esse procedimento é útil para contornar o problema da baixa qualidade de uma boa parte das imagens obtidas por

intermédio do equipamento ecocardiográfico, pois as operações matemáticas de tratamento e segmentação da imagens se concentraram sobre a ROI, buscando desconsiderar os ruídos existentes no restante da imagem e que influenciam, de modo negativo, sobre o resultado da segmentação. Pode-se observar, na Figura 2.10, a ROI correspondente a uma das imagens operadas.

$$\begin{array}{ccc}
 \left| \begin{array}{ccc} e_{11} & e_{12} & e_{13} \\ \vdots & \ddots & \vdots \\ e_{n1} & e_{n2} & e_{n3} \end{array} \right| & & \left| \begin{array}{ccc} e_{11} & \cdots & e_{1m} \\ e_{21} & \ddots & e_{2m} \\ e_{31} & \cdots & e_{3m} \end{array} \right| \\
 n = x_{\max} \cdot 2,5\% & & m = y_{\max} \cdot 2,5\% \\
 \text{(a)} & & \text{(b)}
 \end{array}$$

Figura 2.9 – Elementos estruturantes utilizados na erosão: (a)Elemento para imagens de corte apical de 4 câmaras; (b)Elemento para imagens de corte paraesternal transversal.



Figura 2.10 – Região de Interesse de uma imagem manipulada pela aplicação.

Um ponto de preocupação, e que influencia na área segmentada, é o posicionamento da válvula mitral, responsável por ligar o VE com o átrio esquerdo. Com a seqüência de imagens agrupadas, devido à soma dos pixels dos diversos quadros que traduzem o movimento cardíaco, a região próxima desta válvula aparece borrada causando a junção do ventrículo esquerdo com o átrio esquerdo. A Figura 2.11 mostra o momento de abertura desta válvula.

A solução adotada para contornar este problema foi inserir no algoritmo uma subrotina que conta o número de regiões existentes na imagem, de modo a identificar uma possibilidade de ter ocorrido uma ligação entre as duas câmaras. Essa contagem ocorre

dentro da execução do algoritmo e não ocasiona sobrecarga sensível no tempo de execução do mesmo.

O programa, após a junção das imagens e a binarização do resultado, realiza sucessivas operações morfológicas de modo a ajustá-la e refiná-la, excluindo ruídos. Se mesmo após essas operações, a contagem de regiões identificar uma possível junção das duas câmaras, a imagem é automaticamente desprezada para evitar a realização de uma segmentação equivocada.

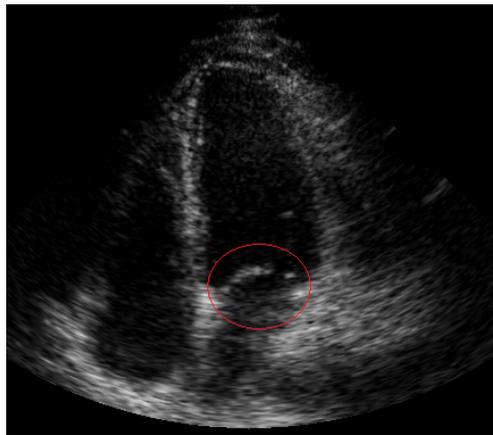


Figura 2.11 – Movimento da válvula mitral e sua influência na segmentação

Sob questões práticas de análise médica, essa região do VE não é tratada com extrema importância pelo médico. As informações mais importantes são a identificação das paredes septal, apical e lateral do VE, de acordo com o que é mostrado na Figura 2.12. Entretanto, a identificação da região onde encontra-se a válvula mitral torna-se necessária para que a execução da segmentação possa ocorrer corretamente, sem a junção do VE e do átrio esquerdo.

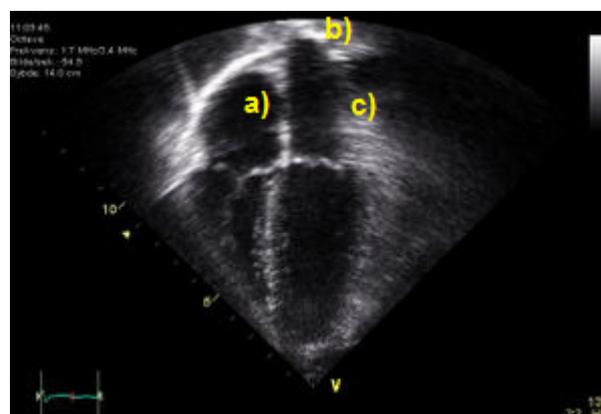


Figura 2.12 – Paredes do ventrículo esquerdo: (a) Septal; (b) Apical; e (c) Lateral.

Foi identificado que a perda de imagens devido à união das câmaras cardíacas com a abertura da válvula mitral, após a aplicação desse tipo de pré-processamento, não passou dos 20% das imagens extraídas do arquivo DICOM. Esse percentual ultrapassa os 50% para o caso da segmentação sem o pré-processamento adequado, confirmando a importância de tal fase do processamento digital.

2.5.2 – Transformada *Watershed*

A transformada *watershed* é aplicada sobre a distância Euclidiana calculada a partir do gradiente da imagem binarizada[4]. Para melhorar o resultado dessa operação sobre a imagem a ser segmentada, utilizou-se o Gradiente Multiescalar ao invés do Gradiente Morfológico simples.

2.5.2.1 – Gradiente Multiescalar

No momento em que realizamos as operações sobre as imagens binárias, preparando-as para submetê-las à segmentação *watershed*, nosso objetivo é identificar os delineadores ou bordas que demarcam a região do VE. Essas bordas indicam uma mudança abrupta no valor dos *pixels* daquela região da imagem[22].

Para identificar essas bordas é utilizado o algoritmo do gradiente morfológico (ρ) sobre a imagem (f). A operação executada por este algoritmo é representada na Equação 2.6, sendo B o elemento estruturante, \oplus a operação morfológica de dilatação e \ominus a operação de erosão. Detalhes dessas operações podem ser vistas na seção B.4.2.

$$\rho(f, B) = (f \oplus B) - (f \ominus B) \quad (2.6)$$

Essa forma de encontrar o gradiente não consegue englobar bordas adjacentes e ruídos próximos das bordas. Para contornar esse problema decidiu-se adotar o algoritmo do gradiente multiescalar (MG), descrito na Equação 2.7, que realiza sucessivos cálculos de gradiente morfológico com diferentes elementos estruturantes[4].

$$MG = \frac{1}{n} \sum [\rho(f, B_i) \ominus B_{i-1}] \quad (2.7)$$

Onde f é a representação matricial da imagem e B_i é o elemento estruturante definido por uma matriz de dimensões $(2i+1, 2i+1)$ de elementos unitários, onde i atende ao intervalo $0 \leq i \leq n$. Pela Equação 2.3, temos as operações sendo realizadas sobre um conjunto finito de elementos estruturantes quadrados.

O número de iterações representado pelo valor inteiro n pode variar, mas na prática observa-se que valores muito maiores do que 15 provocam a degradação total da imagem. No algoritmo proposto, utilizou-se o intervalo de 1 a 10. A Figura 2.13 mostra um exemplo do resultado obtido, durante a primeira iteração, na aplicação do Gradiente Multiescalar sobre uma das imagens binárias manipuladas pela aplicação.

Com este operador, as bordas com diversas distâncias são identificadas já que os elementos estruturantes (B_i) assumem diversos tamanhos e podem posicionar-se entre uma e outra. Com isso, o gradiente máximo é conservado no local correto da imagem.

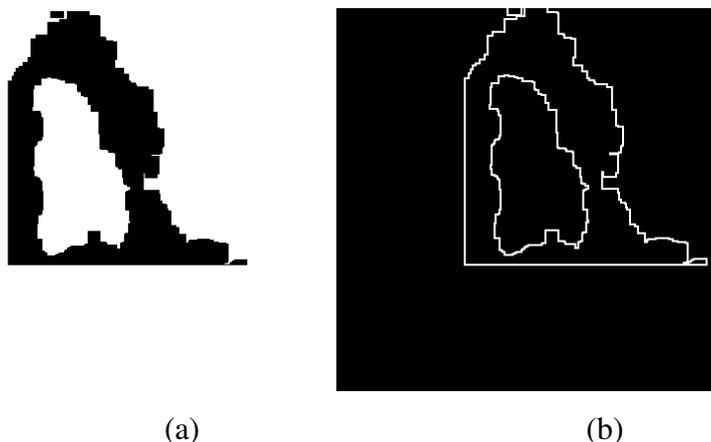


Figura 2.13 – Resultado do Gradiente Multiescalar: (a)Imagem binária do ROI; (b)Resultado do Gradiente sobre o ROI.

2.5.2.2 – Distância Euclidiana

Para melhorar o resultado da aplicação do *watershed* sobre a imagem a ser segmentada, utilizou-se a distância euclidiana [32]. Assim, é possível identificar o ventrículo esquerdo sem que haja interferência das falhas presentes na imagem binária.

Em uma imagem binária, a transformada de distância euclidiana (TDE) gera uma imagem, chamada de mapa de distâncias euclidianas, onde o valor em cada ponto é a menor distância euclidiana deste ponto com o fundo da imagem.

Essa distância entre dois *pixels* $p(x,y)$ e $q(i,j)$ é representada através da Equação 2.8:

$$dist(p,q) = \sqrt{(i-x)^2 + (j-y)^2} \quad (2.8)$$

Onde (x,y) e (i,j) são as coordenadas dos pontos p e q , respectivamente. O ponto p representa os pontos de valor 1 (frente) e q os pontos de valor 0 (fundo). A aplicação da transformada *watershed* sobre o mapa de distâncias euclidianas gera uma imagem de regiões segmentadas mais fiel à realidade.

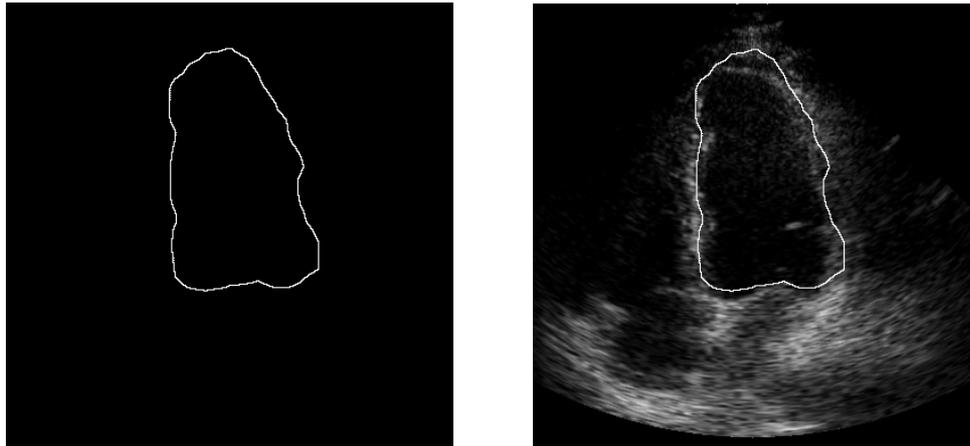
A Figura 2.14 mostra o resultado do cálculo da distância euclidiana sobre a ROI segmentada anteriormente. A região em preto representa o restante da imagem original que foi excluída ao calcular a área de interesse.



Figura 2.14 – Distância Euclidiana sobre a ROI.

2.5.2.3 – Aplicação da Transformada *Watershed*

A aplicação da transformada *watershed* sobre o mapa de distâncias Euclidianas particiona a imagem (ROI) em duas regiões, fundo e objeto. O objeto representa o ventrículo esquerdo (veja Figura 2.15). O contorno do VE é colocado sobre a imagem original do quadro que gerou essa informação. Tal junção proporciona a identificação do ventrículo esquerdo sobre a ecocardiografia original.



(a)

(b)

Figura 2.15 – Segmentação *watershed*: (a)Imagem resultante; (b)Sobreposição do contorno do VE na imagem do quadro original.

3. TECNOLOGIAS UTILIZADAS E ARQUITETURA

3.1 - INTRODUÇÃO

Para a implementação do algoritmo foram utilizadas tecnologias atuais e de grande escalabilidade e robustez, que possibilitam a integração com outros sistemas e o acoplamento de novos módulos, inserindo novas funcionalidades. O considerável grau de manutenibilidade[31] que a implementação possui garante fácil alteração e inserção de funcionalidades, adequando o sistema às necessidades.

O cenário criado utiliza a linguagem JAVA[10] como plataforma principal, responsável por interagir com o usuário, via interface gráfica, e realizar a conexão com os demais componentes e ferramentas utilizadas. Para a implementação das funções matemáticas temos o MATLAB como ambiente de execução e provedor de uma linguagem própria para tal tarefa. A comunicação entre a aplicação JAVA e as bibliotecas do MATLAB é feita através da biblioteca *JMatLink*[13].

Além disso, ferramentas de auxílio no desenvolvimento, tais como o ambiente integrado Eclipse e o JUDE, para tratar dos diagramas no padrão *Unified Modeling Language* – UML, foram utilizados e serão descritos nos itens a seguir, assim como toda a arquitetura adotada na solução implementada.

3.2 – TECNOLOGIAS UTILIZADAS

Algumas tecnologias foram utilizadas na implementação do algoritmo, integrando funcionalidades e complementando recursos específicos. A seguir temos as características das principais tecnologias e ferramentas utilizadas.

3.2.1 – JAVA

O aplicativo foi implementado utilizando a plataforma JAVA[10], sob o paradigma da programação orientada a objetos (POO) e buscando-se a componentização de modo a viabilizar o reuso futuro de partes do aplicativo(componentes).

Essa linguagem, lançada oficialmente em 1995 [10], é considerada de alto nível e não é executada diretamente. Ao compilar o código-fonte obtemos um arquivo em outra estrutura de codificação chamada de *bytecode*. Esta precisa ser interpretada para que gere os resultados esperados.

A interpretação dos arquivos *bytecode* é realizada pela Máquina Virtual JAVA – JAVA Virtual Machine – JVM, que se comporta como uma camada entre a aplicação e o Sistema Operacional, tornando os programas escritos em *JAVA* independentes de plataforma. Podemos identificar na Figura 3.1 que para cada Sistema Operacional – SO temos uma JVM específica que executa os programas em *JAVA*. Devido à necessidade de serem interpretados, os programas desenvolvidos nessa linguagem consomem mais tempo de execução do que os de linguagens que geram executáveis diretos em linguagem de máquina, tal como é a linguagem C.

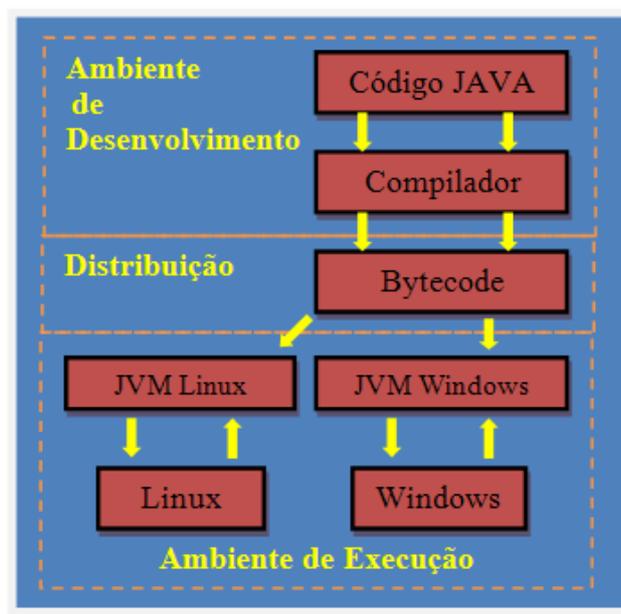


Figura 3.1 – Compilação e interpretação de programas escritos em JAVA

Diversos componentes da *API JAVA* foram utilizados de modo a fornecer funcionalidades adequadas para realizar as operações do algoritmo, bem como fazer a interface gráfica com o usuário. A versão 1.6 da linguagem foi utilizada neste trabalho. Na parte gráfica, para a confecção da interface com o usuário (Figura 2.2), foram utilizadas classes constantes nos pacotes AWT e SWING[29], de modo a facilitar a interação com a aplicação.

3.2.2 – MATLAB

O MATLAB, um sistema especialista para realização de cálculos numéricos baseado em matrizes, é utilizado como ferramenta de apoio e como fornecedor de componentes para cálculo matemático. Pelo fato de utilizar a Linguagem C como nativa para execução dos seus programas, o MATLAB torna-se mais ágil na execução dos algoritmos do que linguagens de mais alto nível, como é o caso do JAVA. Isso atende às necessidades de operações matemáticas complexas diversas.

Para reduzir o tempo gasto em processamento, tornando a implementação aplicável no ambiente real, as funcionalidades prontas de processamento digital de imagens existentes no MATLAB (em sua versão 7.0 – Release 14) foram utilizadas de maneira integrada à aplicação *JAVA* através de chamadas a métodos nativos.

Funções matemáticas para a realização de erosões, dilatações, suavizações e o próprio *watershed*, são reutilizadas por sua confiabilidade e precisão. Ademais, elas serviram como ambiente de testes e simulações anteriores à implementação na aplicação.

3.2.3 – Eclipse

O ambiente integrado de desenvolvimento (IDE) utilizado para a realização deste trabalho foi o Eclipse[16]. Ele reúne todos os recursos necessários para agilizar a programação, compilação e execução do código, além de manipular, de maneira adequada, a linguagem adotada.

Os seus recursos de depuração de código (*debug*) foram amplamente utilizados para identificar e corrigir erros de lógica e de implementação. A versão 3.2 desta ferramenta atendeu a todas as necessidades.

3.2.4 – Ferramentas Auxiliares

Algumas ferramentas auxiliares, chamadas de ferramentas CASE – *Computer-Aided Software Engineering*, foram utilizadas durante este trabalho. Elas são aplicativos utilizados para auxiliar as tarefas dasqueles envolvidos no desenvolvimento de sistemas computacionais.

Uma delas foi o JUDE[40], de uso pessoal livre e que foi utilizada na confecção dos diversos diagramas UML apresentados neste trabalho, além da confecção de fluxogramas e outros gráficos. De uso simples e intuitivo, essa ferramenta ainda oferece recursos como exportação dos diagramas criados em formato de arquivo gráfico e no padrão *XML Metadata Interchange* – XMI, possibilitando a importação por outras ferramentas que possuam a mesma finalidade e recursos.

Outras ferramentas foram utilizadas neste trabalho, tais como o BOUML[41] também para trabalhar com modelos UML e realizar engenharia reversa do código, e o Microsoft VISIO[42] para geração de gráficos e modelos.

3.3 – CONEXÃO DA APLICAÇÃO JAVA COM O MATLAB

Pelo fato dos programas codificados na linguagem *JAVA* serem interpretados, causando sobrecarga no tempo de processamento ao manipular um grande volume de operações matemáticas, adotou-se a estratégia de utilizar chamadas diretas aos componentes do MATLAB, acessando todas as suas funcionalidades e otimizando todo o processamento digital das imagens.

A integração entre a aplicação e o MATLAB é mostrada na Figura 3.2. Ela é feita através do JMatLink[13], uma biblioteca *JAVA* que possibilita a comunicação entre programas em *JAVA* e as funcionalidades do MATLAB através de chamadas a métodos nativos. A comunicação é feita por intermédio de um canal (*connection*) aberto com o MATLAB. Como esse canal não precisa estar iniciado, o consumo de memória e outros recursos da máquina não têm um aumento considerável.

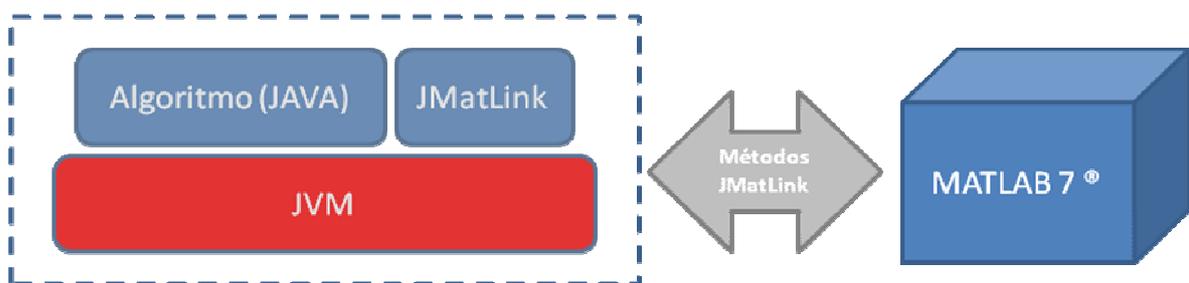


Figura 3.2 – Interação do JAVA com o MATLAB através da API JMatLink

Infelizmente, o projeto JmatLink foi descontinuado em dezembro de 2005, sob a versão 1.3.0. Este *release* apresentou alguns *bugs* de funcionamento com as versões 6 e 7 do MATLAB na classe “JMatLink.java” da biblioteca.

O defeito na API que mais impactou na implementação do algoritmo foi no método *engGetArray(String cmd)* que retorna um *array* de valores com dimensão *double*, resultante da execução no MATLAB da linha de comando representada pela *String cmd*. Esse método não consegue fazer o redimensionamento de valores do MATLAB para o JAVA devido à dimensão dos números manipulados não ser suportada pela linguagem. Esse problema foi identificado no uso da versão 7.0 do MATLAB.

Essa integração trabalha sob o conceito de conexão (*stream*). Ao estabelecer uma com o MATLAB, ela pode ou não, a critério do programador, ser identificada por um número. Conexões não identificadas são causadoras de um outro problema que foi mitigado através do uso do número de identificação da conexão em todas as chamadas a métodos nativos. Essa mesma conexão permanece aberta durante toda a execução do aplicativo. Esses e outros pequenos defeitos na biblioteca foram resolvidos ou contornados e o pacote JmatLink foi incorporado à implementação do algoritmo.

3.4 – AMBIENTE DE EXECUÇÃO

Ao trabalhar com processamento digital de imagens, manipula-se uma grande quantidade de dados, ocupando consideráveis fatias de recursos do *hardware* utilizado, tal como memória e CPU. A execução do programa deste trabalho também é influenciada por essas características ligadas a *hardware* e a desempenho.

A Tabela 3.1 apresenta a configuração da máquina utilizada na execução do programa, e a configuração ideal necessária para otimizar, ao máximo, o processamento matemático da aplicação, reduzindo o tempo de resposta. O item Front Site Bus – FSB foi inserido nesta tabela pois têm grande influência sobre a velocidade final de processamento, já que é ele que define a frequência de comunicação com o barramento de memória, placa de vídeo e outros componentes situado na ponte norte da placa mãe.

A integração com o MATLAB é um fator otimizador, mas, ao mesmo tempo, exige que esse software esteja instalado e seus módulos de operações matemáticas sejam carregados em memória no momento da manipulação das imagens. Uma máquina com boa

capacidade de processamento e quantidade de memória é necessária para que a execução do algoritmo seja feita em um tempo que permita seu uso prático como ferramenta de auxílio ao médico.

Tabela 3.1 – Configuração utilizada e configuração mínima da máquina que executa o algoritmo.

Recurso	Configuração utilizada	Configuração mínima
Processador	Duo 2 Core – 1.6 GHz	Pentium 4 – 2.6 GHz
Memória	2 Gb	1 Gb
Sistema Operacional	Windows Vista	Windows 2000, XP ou Vista
Hard Disk – HD	250 Gb	40 Gb
Front Side Bus – FSB	800 MHz	400 MHz
Vídeo	512 Mb (1280 x 800)	128 Mb (1024 x 768)

Para a implementação e testes, foi utilizada uma máquina PC com os recursos descritos na Tabela 3.1. A configuração mínima desejável para que haja aplicabilidade ao software também é descrita nesta mesma tabela. O processamento paralelo (*multithread* [33]) não é previsto e não foi implementado, mas, devido às tecnologias utilizadas, é possível distribuir o processamento mediante a modificação do sistema e isto ocasionaria um ganho significativo no tempo de execução do sistema.

Apesar de ter sido utilizada a versão 7.0 do MATLAB, todas as funcionalidades e recursos implementados e utilizados são inteiramente compatíveis com a versão 6.

3.5 – ARQUITETURA IMPLEMENTADA

A seguir será feita uma explicação detalhada da arquitetura utilizada na implementação do algoritmo. Os diagramas UML de casos de uso, de pacotes e de classes são utilizados para representar a modelagem feita para a aplicação.

3.5.1 – Casos de uso do aplicativo

Apesar da implementação não constituir um sistema completo, que envolva funcionalidades mais completas, como por exemplo, a comunicação com o equipamento

ecocardiográfico, podemos identificar na Figura 3.3 as funcionalidades básicas que são executadas por quem manipula o aplicativo.

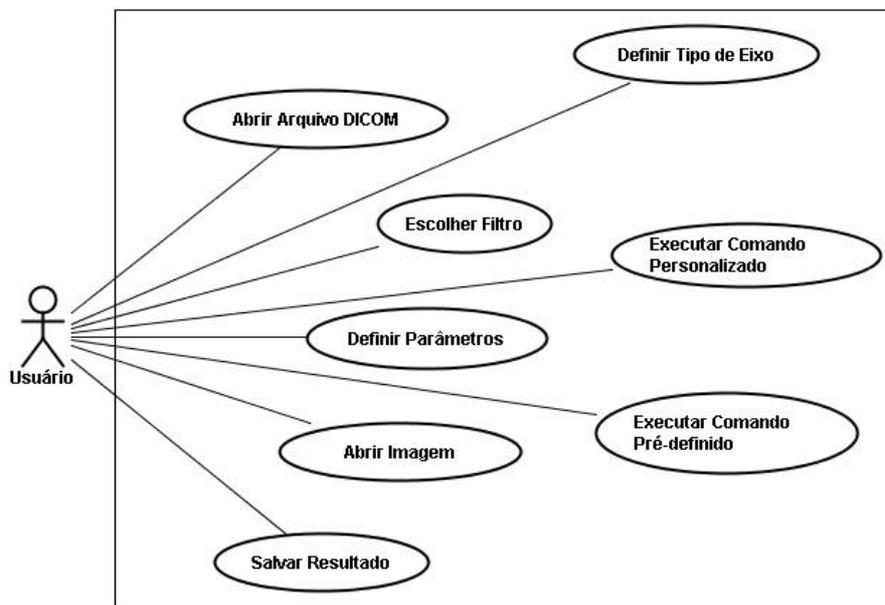


Figura 3.3 – Diagrama de Casos de Uso

Uma descrição resumida destes casos de uso pode ser vista na Tabela 3.2.

Tabela 3.2 – Resumo dos casos de uso

Caso de Uso	Descrição
Escolher Filtro	O usuário pode escolher o filtro a ser utilizado no pré-processamento(Gaussiano, de Lee, Mediana ou Adaptativo)
Definir Tipo de Eixo	Define se a imagem é de corte apical de 4 câmaras (EL) ou paraesternal transversal (EC) (o melhor aproveitamento do algoritmo é em imagens apicais)
Definir Parâmetros	Define os parâmetros de cálculo solicitados na interface da aplicação.
Abrir Arquivo DICOM	Extração das imagens sequenciais constantes do arquivo DICOM. Essa função inicia todo o processamento de segmentação.
Abrir Imagem	Realiza o processamento sobre uma imagem específica no formato BITMAP.
Salvar Resultado	Possibilita salvar as imagens resultantes da segmentação em um diretório definido pelo usuário.
Executar Comando Personalizado	Permite realizar uma linha de comando do MATLAB.
Executar Comando Pré-definido	Permite executar um dos comandos implementados pela aplicação.

Dessas funcionalidades, a principal é a “Abrir Arquivo DICOM”, pois a partir dela todo o processo de segmentação é iniciado automaticamente, ocorrendo a execução de

todas as fases citadas no Capítulo 2. Na Tabela 3.3 temos a descrição para esse caso de uso.

Tabela 3.3 – Descrição do caso de uso “Abrir Arquivo DICOM”

<p>Caso de Uso: Abrir Arquivo DICOM Ator: Usuário Pré-condições: Parâmetros definidos na interface (tipo de filtro e de eixo, número de repetições da erosão e da suavização, número de ciclos que englobam o arquivo DICOM) Pós-condições: Resultados da segmentação salvos na pasta /files/resultado Trigger: Usuário clica no botão “Abrir arquivo DICOM”</p>	
Fluxo Normal	<ol style="list-style-type: none"> 1. O usuário clica em “Abrir arquivo DICOM” 2. O sistema exibe tela para escolha do arquivo 3. O usuário navega nos diretórios locais e escolhe arquivo no padrão DICOM 4. O usuário clica em “Abrir” [E01] 5. O sistema captura os valores definidos na interface gráfica 6. O sistema estabelece comunicação com o MATLAB [E02] 7. O sistema realiza as operações iniciais e exibe as imagens dos resultados intermediários, além dos respectivos histogramas 8. O sistema realiza a segmentação e armazena o resultado no diretório /files/resultado [E03] 9. O sistema exibe as imagens dos resultados finais na tela. 10. Fim de caso de uso
Fluxo de Exceção	<p>E01 – Arquivo não está no formato DICOM</p> <ul style="list-style-type: none"> E01.1 – O sistema verifica que a codificação do arquivo não é correspondente ao DICOM E01.2 – O sistema exibe mensagem de erro E01.3 – Fim de caso de uso <p>E02 – Problemas ao estabelecer comunicação com as bibliotecas do MATLAB</p> <ul style="list-style-type: none"> E02.1 – O sistema exibe mensagem de erro E02.3 – Fim de caso de uso <p>E03 – Erro ao criar arquivos (Erro de ES)</p> <ul style="list-style-type: none"> E03.1 – O sistema não consegue gravar as imagens nos diretórios definidos; E03.2 – O sistema exibe mensagem de erro E03.3 – Fim de caso de uso

Um objetivo secundário desta implementação é disponibilizar uma arquitetura adequada e escalável de modo que suporte o acoplamento de novos aplicativos e/ou funcionalidades com relativa facilidade, permitindo, inclusive, a comunicação direta com os equipamentos médicos de captura de imagens.

3.5.2 – Diagrama de classes

O diagrama, no padrão UML, que descreve a estrutura e a relação dos elementos que compõem o sistema é o Diagrama de Classes. Tal diagrama pode ser visto, de maneira resumida, na Figura 3.4.

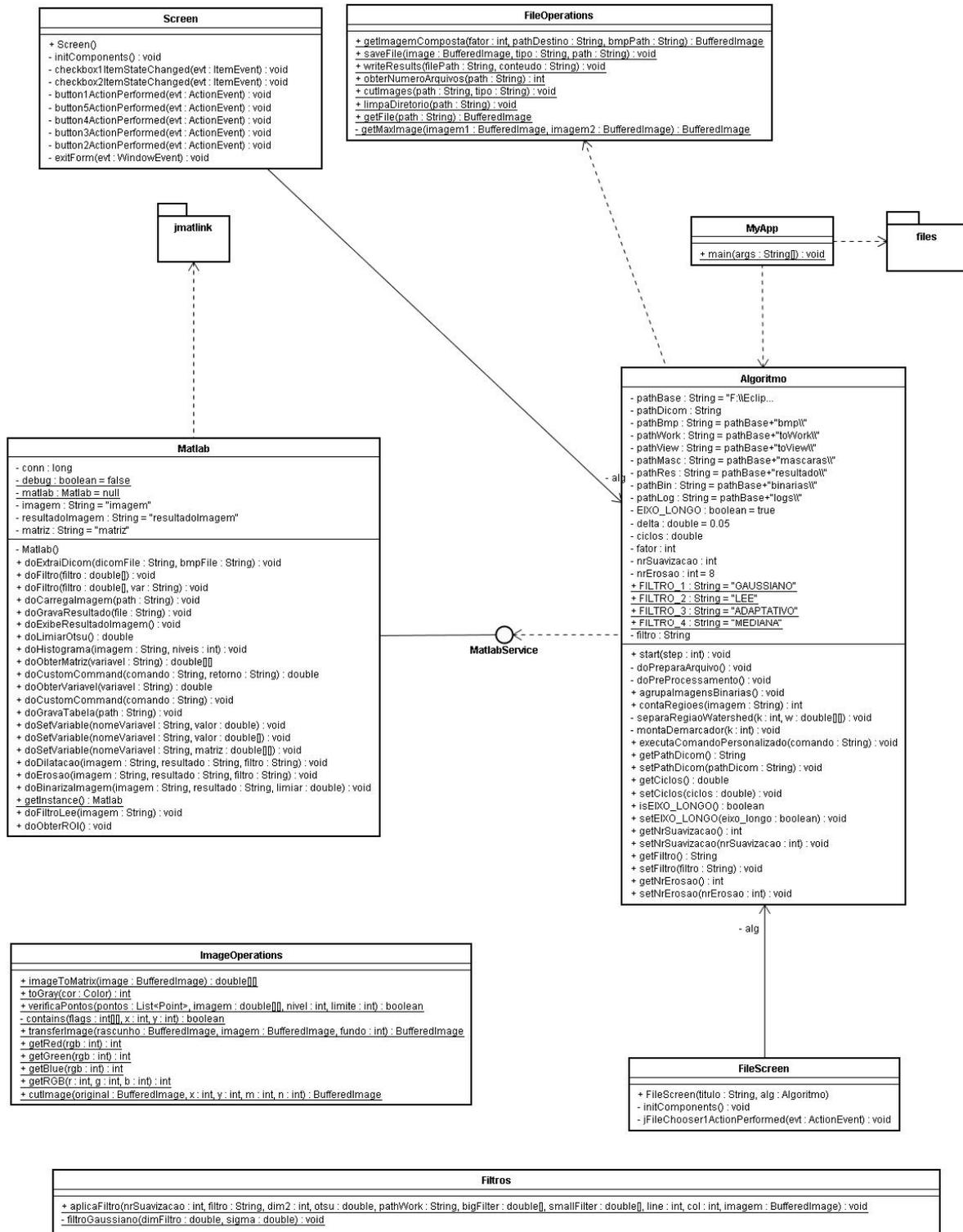


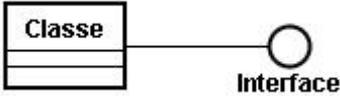
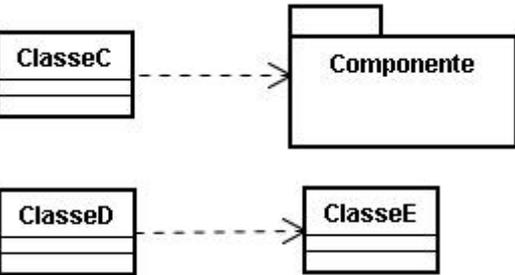
Figura 3.4 – Diagrama de Classes

Nele são identificadas as principais *interfaces* e classes concretas do sistema. O uso de *interfaces* contribui para o baixo acoplamento entre as partes da aplicação e facilita o ajuste com novos componentes.

Esse diagrama transcreve os relacionamentos e comunicações existentes entre as entidades do sistema, sob os aspectos estáticos. As setas pontilhadas indicam uma dependência entre os elementos que a utilizam. As linhas contínuas indicam uma associação, ou seja, a existência de uma ou várias instâncias no interior de outra. Os círculos representam *interfaces* e as linhas contínuas que as ligam indicam qual é a classe concreta que a implementa.

A Tabela 3.4 exemplifica, de maneira resumida, alguns dos relacionamentos existentes no diagrama mostrado.

Tabela 3.4 – Notações de relacionamentos

Símbolo	Significado
	Este relacionamento indica que a Classe implementa a Interface (Todos os métodos de Interface deverão ser implementados)
	Define uma associação simples direcional entre a ClasseA e a ClasseB. No caso, temos que a ClasseA possui uma instância da ClasseB como atributo.
	A seta pontilhada indica dependência. No primeiro caso temos a dependência da ClasseC com o pacote Componente, e no segundo temos a ClasseD dependendo da ClasseE. Essa dependência indica o uso de algum atributo ou método da classe ou componente.

Além dos relacionamentos, o diagrama de classes mostra os atributos e métodos (funções) de cada uma das entidades. Com isso, temos todas as informações necessárias para a confecção dos diagramas de seqüência, quando são necessários, e sua conseqüente implementação.

A Tabela 3.5 resume as funções das principais classes apresentadas no diagrama da Figura 3.5.

Tabela 3.5 – Funções das principais classes

Classe	Função
FileOperations	Reúne todas as operações realizadas com arquivo (abertura, escrita, conversão, etc.)
ImageOperations	Reúne todas as operações realizadas com imagens (conversão, amostragem, etc.)
Matlab	Implementa a interface MatlabServices e concentra todas as funções que são executadas pelas bibliotecas do MATLAB.
FileScreen	Classe responsável por solicitar a escolha do arquivo a ser aberto
Screen	Interface gráfica principal do programa
jMatLink	Pacote com as bibliotecas de conexão com o MATLAB
files	Pacote de armazenamento das imagens
MyApp	Executável do programa

3.5.3 – Diagrama de Componentes

Em UML, os diagramas de componentes mostram os elementos reutilizáveis de software e sua interdependência. Um componente é formado por um conjunto de classes que encontram-se implementadas em seu interior, podendo depender, funcionalmente, das classes de outro componente. Os componentes são identificados como pacotes e o diagrama pode representar, além de componente, subsistemas.

No diagrama de componentes também é possível mostrar a configuração de um sistema de software mostrando, graficamente, a dependência entre os diversos arquivos que o compõem.

O objetivo de dividir um sistema em componentes é possibilitar o reuso futuro em outros sistemas e facilitar a manutenção e o desenvolvimento, concentrando-a apenas no interior do componente. No caso da aplicação em questão, buscou-se concentrar as classes em pacotes de acordo com a similaridade de funções.

A arquitetura da implementação do algoritmo está dividida em pacotes de modo a facilitar a manutenção e o incremento do mesmo. O diagrama de componentes mostrado na

Figura 3.5 exibe a estrutura adotada e a Figura 3.6 mostra a estrutura de pastas utilizadas pela aplicação.

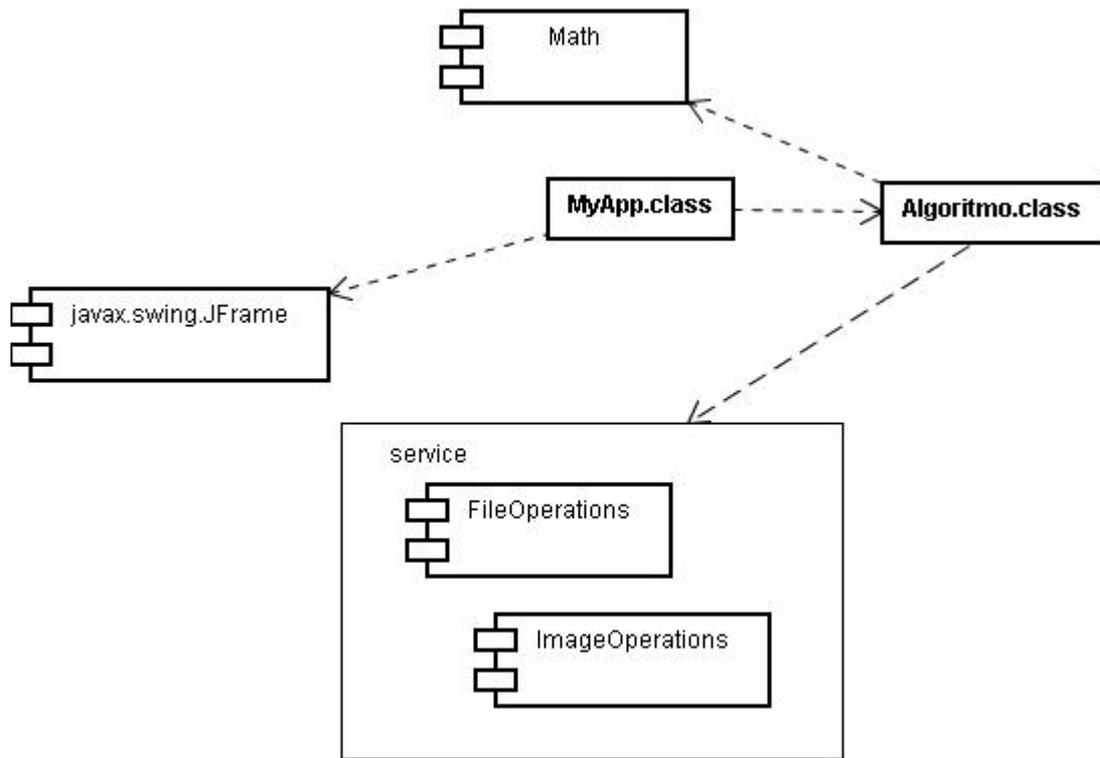


Figura 3.5 – Estrutura de pacotes

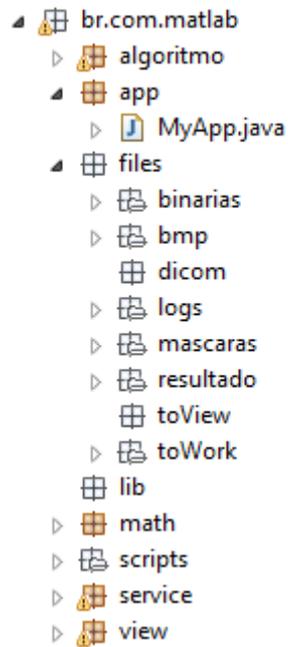


Figura 3.6 – Estrutura de pastas do aplicativo

O pacote *Math* reúne as classes que realizam operações com as bibliotecas do MATLAB. O pacote *service* reúne as classes de serviço utilizadas nas operações comuns à várias ações, no que diz respeito a arquivos e imagens. A classe *Algoritmo* concentra a gerência de todas as etapas do algoritmo proposto, realizando as chamadas necessárias aos componentes e classes adequados. A classe *MyApp* é a executável da aplicação e faz uso da classe gráfica *JFrame* do pacote SWING da API JAVA.

Além de mostrar a estrutura de pastas, a Figura 3.7 detalha o pacote *br.com.matlab.files*, onde as imagens e os resultados parciais e totais são disponibilizados para análise e manipulação pelo aplicativo. A Tabela 3.6 descreve quais são os arquivos armazenados em cada subpasta do pacote *files*.

O mapeamento dos diretórios supracitados está inserido no código, mas de fácil alteração através de um caminho básico (*path*). Havendo a necessidade, essa gravação dos arquivos pode ser feita em qualquer lugar devidamente mapeado e montado. A alteração deste caminho, por estar no código, exige uma recompilação do mesmo.

Tabela 3.6 – Descrição do diretório *br.com.matlab.files*

Caso de Uso	Descrição
binarias	Armazena as imagens resultantes da binarização
bmp	Armazena as imagens em format <i>bitmap</i> de cada quadro extraído do arquivo DICOM
dicom	Armazenamento dos arquivos DICOM
logs	Arquivos texto que guardam informações do processamento matemático
maskas	Imagens com o delineamento do VE
resultado	Resultado final da segmentação com a imagem original e o delineamento do VE
toView	Arquivos temporarios para debug
toWork	Arquivos temporários de resposta das bibliotecas (<i>engine</i>) da MATLAB

4. ANÁLISE COMPARATIVA

A proposta deste trabalho, além de apresentar uma solução viável para a segmentação do ventrículo esquerdo em imagens de ecocardiograma, é mostrar de maneira comparativa o grau de influência do pré-processamento na operação das imagens.

A seguir serão mostrados os resultados da comparação entre a segmentação manual do profissional de saúde e a segmentação semi-automática deste trabalho. Esse também foi comparado com algoritmos de outros autores para avaliação do seu desempenho. Essa comparação foi alcançada aplicando três índices de comparação: o Índice de Correlação – IC, o Erro Percentual – EP e o Somatório dos Erros – SE.

4.1 – ÍNDICES DE COMPARAÇÃO

Para avaliar a eficiência do algoritmo proposto, é feita a comparação dos resultados obtidos de três formas: Erro Percentual – EP, Índice de Correlação – IC e Somatório dos Erros – SE. Os índices relativos ao EP, SE e ao IC podem ser obtidos através da aplicação das Equações 4.1, 4.2 e 4.3 respectivamente[35].

$$EP = \left(\frac{MX}{M} \right) \times 100 \quad (4.1)$$

$$SE = \left(\frac{MX}{M} \right) \times 100 + \left(\frac{AX}{M} \right) \times 100 \quad (4.2)$$

$$IC = \left(\frac{P_1}{P_2} \right) \times 100 \quad (4.3)$$

onde M é o número de *pixels* que representa a área obtida com a segmentação manual, MX representa a quantidade de *pixels* que pertencem à região de segmentação manual e não pertencem à área de segmentação automática, e AX representa o número de *pixels* que pertencem exclusivamente à segmentação automática e não fazem parte da segmentação manual. Além disso, P_1 e P_2 representam a menor e a maior área, respectivamente, entre as áreas da segmentação manual e automática.

O IC indica a correlação linear entre as áreas obtidas no modo automático com as áreas obtidas através da segmentação manual. O EP indica o percentual de diferença entre as áreas, e o SE representa a soma do percentual de áreas não coincidentes.

Para o fator *IC*, a meta é atingir o valor 1. Nesse caso, as áreas da segmentação automática e manual possuem exatamente a mesma dimensão. Isso, porém, não indica a precisão do processo, pois as posições das áreas podem ser diferentes. É preciso os outros dois indicadores para mensurar a precisão da segmentação, e quanto menores melhor o resultado expresso por eles.

4.2 – SELEÇÃO DAS IMAGENS

O algoritmo foi aplicado sobre um total de 14 arquivos DICOM, utilizando o filtro Gaussiano e o filtro de Lee nas operações de suavização. Desses arquivos, 5 não foram segmentados devido à quantidade excessiva de ruídos em seus quadros. Os 9 arquivos restantes foram operados e geraram um total de 47 imagens segmentadas, fornecendo uma média de 5 imagens para cada arquivo DICOM.

A análise, pelo médico colaborador deste trabalho, das imagens segmentadas levou à exclusão de 50% do total, pelo fato da qualidade da imagem original não ser suficiente para o trabalho de segmentação manual. Nos trabalhos diários do médico, esta exclusão das imagens de características insuficientes é uma atividade normal, tendo em vista a dificuldade de captura das mesmas, muito dependente da qualidade do equipamento ecocardiográfico utilizado e de outros fatores externos. Do universo de imagens operadas, conseguiu-se a execução completa da análise (segmentação pelo algoritmo e pelo médico) sobre um conjunto de 14 imagens (arquivos BITMAP de 24 bits), oriundas de 9 arquivos DICOM que representaram 64% dos exames disponíveis para a realização deste trabalho. Estas 14 imagens foram agrupadas em dois conjuntos, alta qualidade (8 arquivos) e baixa qualidade (6 arquivos), de acordo com a classificação feita pelo profissional de saúde.

É importante ressaltar que a precisão necessária na segmentação do VE, sob a ótica médica, resume-se a pontos específicos, não abrangendo todo o perímetro ventricular. Para exemplificar este aspecto podemos citar a precisão, não tão exigida, na identificação da região próxima à válvula mitral.

4.3 – TESTE DO PRÉ-PROCESSAMENTO

A comprovação do ganho obtido na segmentação a partir do pré-processamento das imagens com o método de agrupamento sugerido e descrito na seção 2.4 foi verificada da seguinte forma:

- Aplicação do pré-processamento sem agrupamento de imagens, seguido da operação de segmentação por *watershed*. (Figura 4.1(a))
- Aplicação do pré-processamento com o método de agrupamento seguido da mesma operação de segmentação da seção anterior. (Figura 4.1(b))

A Figura 4.1 fornece uma amostra dos resultados obtidos com essa experiência. A seção 2.4 descreve o método de segmentação utilizado.

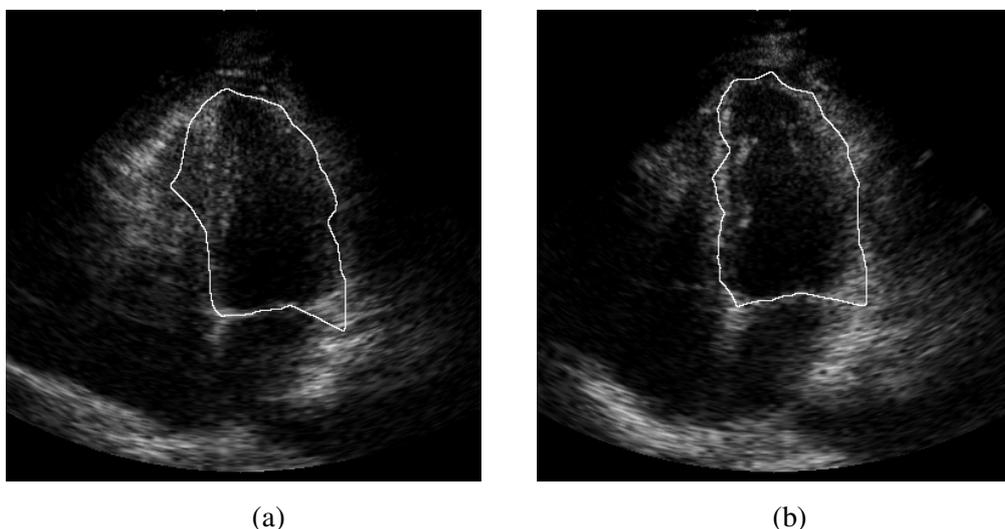


Figura 4.1 – Resultados da segmentação após pré-processamento: (a) Sem agrupamento de imagens; (b) Com agrupamento de imagens.

Apesar da baixa qualidade das imagens originais, é perceptível a diferença no resultado alcançado pela segmentação precedida por um pré-processamento adequado, aproximando-se do resultado obtido pelo médico. Na Figura 4.1(b), foram agrupados cinco quadros consecutivos.

Para um primeiro teste, o filtro Gaussiano foi aplicado apenas uma vez sobre a imagem. A erosão, que exclui pontos indesejados de ligação na imagem binária, foi realizada quatro vezes. Esse teste foi executado sobre um conjunto de cinco arquivos DICOM e confirmou o ganho obtido com o agrupamento por similaridade.

Um aspecto interessante a ser observado é que o aumento do número de aplicações dos diversos filtros (Gaussiano, de Lee, Adaptativo ou de Mediana) acarretou ganho até

um certo limiar. A partir desse valor, ocorreu apenas o aumento no tempo de processamento e, em alguns casos, perda de informações significativas na imagem. Além disso, em imagens com elevado grau de ruído, o excesso de suavização causou a eliminação da fronteira do ventrículo esquerdo, tornando o resultado impreciso.

4.4 – TESTE DA SEGMENTAÇÃO

Serão apresentadas várias séries de segmentações com alterações no pré-processamento quanto ao filtro de suavização escolhido e o número de erosões e suavizações aplicados.

A primeira série de segmentações foi calculada utilizando-se o filtro Gaussiano. O algoritmo de suavização foi executado mais de uma vez para a redução da intensidade de ruídos e a erosão foi aplicada 4 vezes sobre a imagem binarizada. A Figura 4.2 mostra os resultados obtidos.

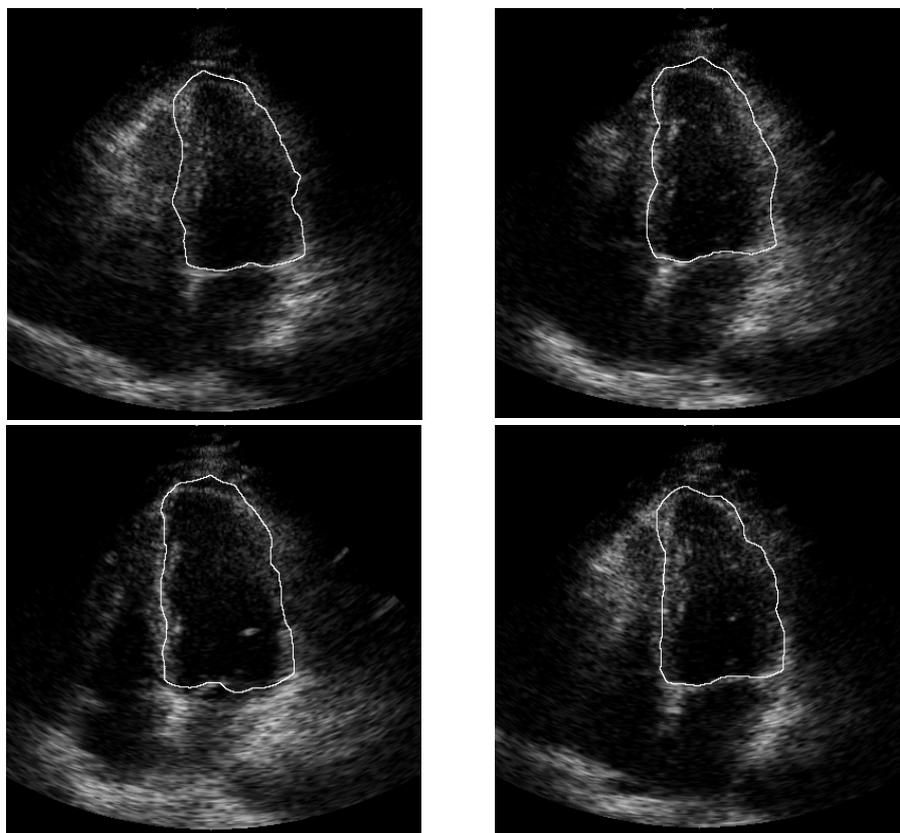


Figura 4.2 – Alguns resultados obtidos com o algoritmo proposto ao aplicar 4 erosões.

As áreas foram semelhantes às obtidas na segmentação manual, mas ainda com considerável taxa de erro. Para esse cálculo, foram analisados a precisão, que compara a proximidade da curva delineadora do VE criada pelo médico e a construída pelo sistema, além da relação de área, comparando a quantidade de *pixels* que compõem as áreas das segmentações automática e manual.

Na segunda série de segmentação, foi inserido o cálculo do Gradiente Multiescalar e o número de erosões foi incrementado para 6. A quantidade de repetições da aplicação da suavização também foi incrementada até chegar ao valor ótimo 6. Na Figura 4.3, podemos ver a melhoria nos resultados com a utilização deste número de erosões. Para um número de erosões superior a 6, o tempo de processamento torna-se excessivamente elevado e oferecendo um ganho mínimo nos resultados podendo, inclusive, danificar a imagem.

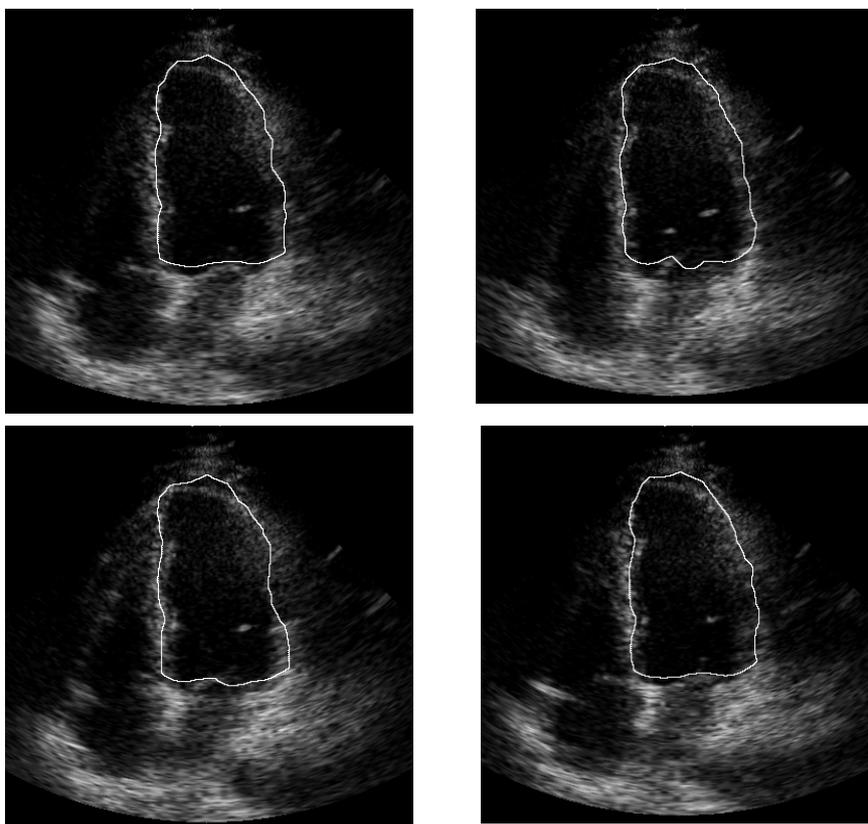


Figura 4.3– Alguns resultados obtidos com o algoritmo proposto usando o Gradiente Multiescalar seguido de 6 erosões.

Para verificarmos a precisão da segmentação, confrontamos os resultados obtidos com as imagens segmentadas manualmente, como é mostrado na Figura 4.4, de modo a identificar visualmente a proximidade significativa entre as linhas de demarcação criadas pelo sistema e colocadas pelo médico. Quanto mais próximas essas linhas, melhor o resultado do algoritmo proposto.

As áreas segmentadas pelo algoritmo da aplicação, na maioria das vezes, foram maiores do que as regiões identificadas pelo médico. Tal fato se deve pela qualidade deficiente da maioria das imagens trabalhadas, existindo um excesso de ruído nas mesmas que não pôde ser completamente eliminado pelas etapas de suavização e filtragem. Essa dificuldade na aquisição das imagens é considerada pelo médico como fato normal. Ela torna este tipo de exame demorado e pode exigir várias tomadas consecutivas de imagens do tórax do paciente.

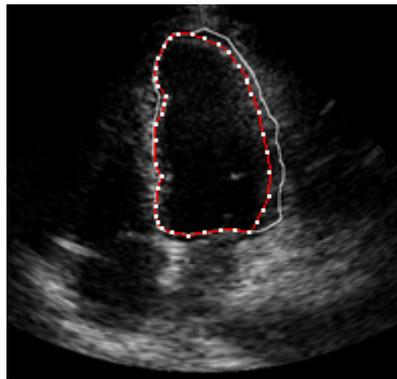


Figura 4.4 – Comparação entre a segmentação automática (linha contínua) e a manual (linha pontilhada) utilizando o Gradiente Multiescalar e 6 erosões.

Podemos considerar tais resultados satisfatórios no contexto de sua realização, principalmente por exigir pouca interação do usuário com a aplicação.

Na Figura 4.5 são mostrados dois resultados comparativos entre a segmentação manual e a executada pelo algoritmo utilizando o filtro de Lee e 6 erosões sucessivas. O detalhe é que a aplicação desse filtro aumenta o tempo de processamento do algoritmo.

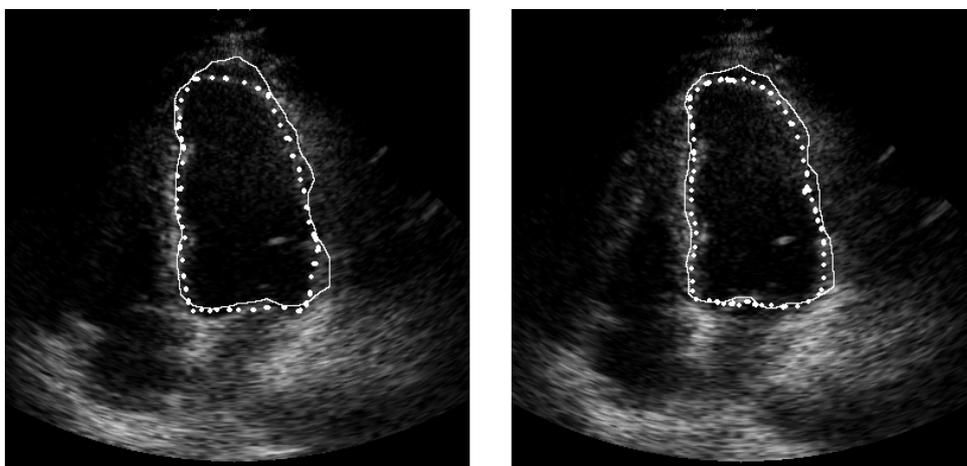


Figura 4.5 – Comparação entre a segmentação semi-automática (linha contínua) e a manual (linha pontilhada) utilizando o filtro de Lee.

Foi observado que a mudança de pré-processamentos através da troca dos filtros de suavização, número de execução de erosões e dilatações, e a identificação das áreas indesejadas para enfatizar a ROI, impactam consideravelmente na etapa de segmentação através do *watershed*.

4.5 – APLICAÇÃO DOS ÍNDICES DE COMPARAÇÃO NOS RESULTADOS DA SEGMENTAÇÃO

Os resultados obtidos na segmentação manual (SM) e semi-automática (SA), utilizando o filtro Gaussiano com os parâmetros definidos na seção anterior, foram comparados de acordo com os índices descritos na seção 4.1. A Tabela 4.1 mostra o índice de correlação (IC) obtido a partir da comparação entre as áreas das imagens obtidas pelas segmentações. O cálculo realizado possui uma precisão de +/- 0,025 devido às variações existentes nas marcações e delineamento das regiões nas imagens, processo este com interferência manual.

Tabela 4.1 – Resultados do Índice de Correlação (IC) para imagens de alta e baixa qualidade

Qualidade	Arquivo	Manual (<i>pixels</i>)	Semi-Automática (<i>pixels</i>)	IC (P ₁ /P ₂)
Alta	img01.bmp	20043	22050	0,909
	img02.bmp	19404	20787	0,933
	img03.bmp	18827	21191	0,888
	img04.bmp	20031	21107	0,949
	img05.bmp	19107	21253	0,899
	img06.bmp	20151	21340	0,944
	img07.bmp	20412	22528	0,906
	img08.bmp	20494	22665	0,904
Baixa	img09.bmp	20120	22765	0,884
	img10.bmp	18087	15001	0,829
	img11.bmp	18409	21635	0,851
	img12.bmp	17202	20134	0,854
	img13.bmp	17978	20513	0,876
	img14.bmp	17867	20514	0,871

Os resultados da segmentação semi-automática foram obtidos, após realizar alguns ajustes dos parâmetros utilizados na segmentação de modo a maximizar o rendimento e a precisão. Após esses ajustes, chegou-se aos índices ideais sendo esses aplicados sobre o conjunto de imagens com a repetição por duas vezes da aplicação do filtro Gaussiano seguida de quatro erosões consecutivas, além dos demais passos de segmentação citados no Capítulo 2.

Ao observar os IC obtidos é identificado uma proximidade com o valor 1, que seria a segmentação perfeita com o completo alinhamento das regiões da segmentação manual e da segmentação automática. O valor médio (μ) desse índice foi 0,9165 com desvio padrão (σ) de 0,0224 que confirma essa proximidade com o valor ideal (IC = 1).

Em 99% dos resultados a área correspondente à segmentação automática é ligeiramente maior do que a área segmentada pelo médico, envolvendo tal região e com pouco deslocamento com relação aos seus centros.

A Tabela 4.2 mostra o Erro Percentual (EP) das segmentações realizadas pelo algoritmo proposto sob as mesmas condições e o mesmo conjunto de imagens utilizados para o cálculo do IC.

Tabela 4.2 – Resultados do índice Erro Percentual (EP) para imagens de alta e baixa qualidade

Qualidade	Arquivo	Manual (pixels)	SM-SA (pixels)	EP (%)
Alta	img01.bmp	20043	667	3,32
	img02.bmp	19404	245	1,26
	img03.bmp	18827	236	1,25
	img04.bmp	20031	307	1,53
	img05.bmp	19107	94	0,50
	img06.bmp	20151	98	0,49
	img07.bmp	20412	211	1,03
	img08.bmp	20494	769	3,75
Baixa	img09.bmp	20120	536	2,66
	img10.bmp	18087	3454	19,10
	img11.bmp	18409	1075	5,84
	img12.bmp	17202	847	4,92
	img13.bmp	17978	1015	5,65
	img14.bmp	17867	972	5,44

Novamente observamos que, para as imagens de alta qualidade, o EP aproxima-se de 0 que seria o valor ideal. Para este índice, a média (μ) foi igual a 1,64 com desvio padrão (σ) de 1,23, muito inferior ao valor médio de 5% que seria a tolerância máxima de erro.

A Tabela 4.3 mostra os resultados do cálculo do SE, que mostra o percentual das áreas não coincidentes dos dois métodos de segmentação. Tanto para o SE quanto para o EP, há uma taxa de erro de +/- 0,25% devido a imprecisões no delineamento manual das regiões segmentadas.

Tabela 4.3 – Resultados do índice Somatório do Erro (SE) para imagens de alta e baixa qualidade

Qualidade	Arquivo	(SM-SA)/SM (%)	(SA-SM)/SM (%)	SE (%)
Alta	img01.bmp	3,32	1,00	13,34
	img02.bmp	1,26	7,12	8,39
	img03.bmp	1,25	12,55	13,81
	img04.bmp	1,53	5,37	6,90
	img05.bmp	0,49	11,23	11,72
	img06.bmp	0,48	5,90	6,39
	img07.bmp	1,03	10,36	11,40
	img08.bmp	3,75	10,59	14,35
Baixa	img09.bmp	2,66	13,14	15,81
	img10.bmp	19,09	17,06	36,16
	img11.bmp	5,83	17,52	23,36
	img12.bmp	4,92	17,04	21,97
	img13.bmp	5,64	14,10	19,75
	img14.bmp	5,44	14,81	20,26

Os resultados obtidos para o SE serão comparados com o obtido em outros trabalhos de modo a fornecer um parâmetro de comparação para análise de rendimento do algoritmo.

A Tabela 4.4 mostra um resumo de todos os índices obtidos dos resultados alcançados.

Tabela 4.4 – Índices de comparação obtidos para imagens de alta e baixa qualidade

Qualidade/Origem	$IC = \mu \pm \sigma$	$EP = \mu \pm \sigma$	$SE = \mu \pm \sigma$
Alta	$0,917 \pm 0,02$	$1,64 \pm 1,23$	$10,79 \pm 2,95$
Baixa	$0,861 \pm 0,02$	$7,27 \pm 5,91$	$22,89 \pm 6,99$

Realizando testes utilizando o filtro de Lee na suavização, obteve-se, como um dos resultados, o mostrado na Figura 4.6, apresentando um IC igual à 0,961 que é superior ao obtido com o filtro Gaussiano. Porém este processamento demanda um tempo maior de execução, sendo, em média, de 23 minutos e 45 segundos para apresentar o resultado, contra a média de 2 minutos para o processo anterior.

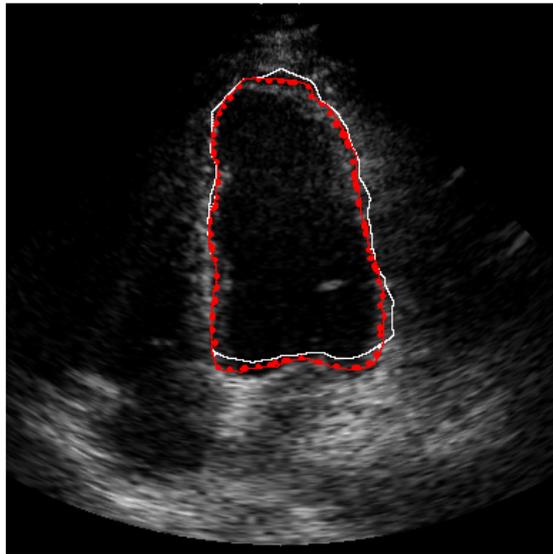


Figura 4.6 – Comparação entre a segmentação automática de alta qualidade e a manual.

Com este tempo de execução, torna-se inviável a aplicação deste algoritmo no ambiente real. Além disso, o filtro ocasionou uma perda superior a 50% das imagens operadas pelo fato de levar a etapa seguinte, erosão, a excluir informações significativas na imagem.

4.6 – COMPARAÇÃO COM RESULTADOS OBTIDOS POR OUTROS AUTORES

Analisando a estrutura de outros algoritmos, observa-se a existência de restrições e peculiaridades no manuseio das imagens exigindo como entrada, inclusive, uma imagem mais bem preparada, recortada e posicionada, como entrada, de modo a realizar a

segmentação na forma correta. O *software* aqui implementado não exige preparação alguma da imagem, realizando de forma automática a leitura do arquivo DICOM, a extração das imagens no formato BITMAP e as operações de segmentação propriamente dita. Essa diferença exigiu um esforço considerável em adaptar as imagens de modo que pudessem ser tratadas por mais de um algoritmo.

Com o objetivo de comparar os resultados obtidos pelo algoritmo apresentado por Reis[39], os diversos valores das Tabelas 4.1, 4.2 e 4.3, das imagens correspondentes, foram agrupados dentro das respectivas qualidades através da média e do seu desvio padrão, da maneira mostrada na Tabela 4.5. O algoritmo de Reis foi otimizado para operações com imagens obtidas no corte paraesternal transversal do VE, mas é capaz de realizar, com menor eficiência, segmentações em imagens de corte apical.

Esse algoritmo foi aplicado sobre cinco imagens de alta qualidade, e o algoritmo proposto neste trabalho foi executado sobre esse mesmo conjunto de imagens.

Tabela 4.5 – Índices de comparação para dois algoritmos: Reis[39] e o deste trabalho

Qualidade/Origem	IC	$EP = \mu \pm \sigma$	$SE = \mu \pm \sigma$
Alta	0,949	$1,64 \pm 1,23$	$10,79 \pm 2,95$
Baixa	0,876	$7,27 \pm 5,91$	$22,89 \pm 6,99$
Alta / Reis[39]	0,95	$2,49 \pm 2,46$	$9,62 \pm 7,90$

Apesar de não ser especializado neste tipo de imagem, o algoritmo de comparação da tabela executou a segmentação com qualidade e apresentou índices semelhantes ao algoritmo proposto neste trabalho. Os resultados inferiores são devido às etapas de pré-processamento das imagens que não exploram as características peculiares existentes das ecocardiografias de corte apical.

A Tabela 4.6 mostra os valores dos índices encontrados com o trabalho proposto por Silva[36], que também faz uso da transformada *watershed* na segmentação de imagens médicas. Estes resultados foram obtidos sobre o mesmo conjunto de 8 imagens apicais de alta qualidade e são inferiores ao obtido pelo algoritmo proposto neste trabalho, como pode ser observado nessa tabela.

Tabela 4.6 – Índices de comparação para dois algoritmos: Silva[36] e o deste trabalho

Qualidade/Origem	IC	$EP = \mu \pm \sigma$	$SE = \mu \pm \sigma$
Alta	0,949	$1,64 \pm 1,23$	$10,79 \pm 2,95$
Alta / Silva[36]	0,897	$2,61 \pm 2,13$	$17,88 \pm 3,40$

5. CONCLUSÃO E TRABALHOS FUTUROS

A importância da análise do ventrículo esquerdo no funcionamento cardíaco como um todo, fornecendo elementos para identificar possíveis anomalias funcionais e anatômicas deste órgão, confere relevância às pesquisas nesta área e foi fator motivador à realização deste trabalho.

O presente trabalho dá uma mostra das potencialidades existentes nesta área de conhecimento, integra a tecnologia da informação com a área médica, e incrementa o trabalho dos profissionais de saúde. De maneira alguma o objetivo deste trabalho foi o de substituir o papel do médico no diagnóstico durante o tratamento cardíaco, no que diz respeito à identificação do ventrículo esquerdo. A meta é unir esforços de modo a criar uma ferramenta de apoio ao cardiologista de maneira a aumentar a precisão do diagnóstico fornecido ao paciente, auxiliando nessa tarefa.

Todos os resultados obtidos neste trabalho foram analisados por médico ecocardiografista sendo considerados satisfatórios dentro do conjunto de imagens consideradas passíveis de avaliação médica, já que um grande número de imagens capturadas pelo equipamento ecocardiográfico são ignoradas por não conterem as informações visuais mínimas.

O conjunto baixa complexidade do código, pouco tempo de execução e portabilidade, alinhado aos consideráveis resultados obtidos, torna o objetivo principal deste trabalho alcançado, tendo enfatizado o importante papel do pré-processamento das imagens no processo de segmentação do ventrículo esquerdo.

Através do acompanhamento, *in loco*, de exames ecocardiográficos pôde-se observar a dificuldade em capturar boas imagens cardíacas e a real necessidade de precisão na segmentação do ventrículo esquerdo, exaltando a região próxima à válvula mitral e com pouca atenção nas paredes laterais. Devido a isso, foi preocupação deste trabalho tratar as imagens afetadas pelas diástoles e sístoles cardíacas.

Vale ressaltar que os equipamentos de ecocardiografia tomados por base neste trabalho foram os existentes no HUB (Hospital Universitário de Brasília) e no HBDF (Hospital de Base do Distrito Federal), que não são os mais modernos e apresentam limitações e deficiências na captura das imagens, mas que são ainda os mais encontrados nas clínicas nacionais com este tipo de consulta.

Além disso, as variações existentes na segmentação manual, devido às diferentes perspectivas e observadores, podem conduzir a diagnósticos imprecisos. Esse problema é

eliminado com a utilização da segmentação semi-automática que aplica, sempre da mesma forma, as mesmas regras matemáticas para identificar o ventrículo esquerdo, servindo de orientador para o trabalho manual e reduzindo as variações no exame.

A utilização da linguagem JAVA na implementação deste trabalho proporcionou uma arquitetura reutilizável e aderente ao uso conectado a equipamentos médicos. Infelizmente, por se tratar de uma linguagem de alto nível, o processamento matemático fica com o seu rendimento comprometido, consumindo longas janelas de tempo em sua execução.

As diferenças de versões da ferramenta MATLAB utilizada nos diversos trabalhos de pesquisadores do grupo de Processamento Digital de Sinais da Universidade de Brasília, e o código complexo e não documentado desses trabalhos gerou dificuldade e limitações no intercâmbio entre partes dessas implementações. Isso dificultou a integração das diversas ferramentas que já foram desenvolvidas. Um trabalho futuro de adaptação dessas diferentes implementações pode ser feito de modo a obter um algoritmo geral otimizado, formado pelas melhores etapas desses diferentes códigos, e que não foram englobados por este trabalho.

Explorar este tipo de segmentação a fim de obter informações de modo a reconstruir, tridimensionalmente, o ventrículo esquerdo é uma aplicação futura valiosa pois viabiliza a obtenção de informações críticas ao diagnóstico médico, como o volume ventricular.

Há a necessidade de aperfeiçoar a identificação da região apical, inserindo novos filtros e tratamentos na imagem de modo a mitigar os ruídos que surgem, de maneira mais intensa, nessa área devido à distância e inclinação do transdutor. Essa área é uma das que fornecem informações valiosas ao médico.

Um trabalho mais específico na parte de implementação do código pode realizar um estudo visando parametrizar as variáveis que otimizam a segmentação de modo que o algoritmo torne-se ótimo para imagens de corte apical, paraesternal e supraesternal. Além disso, esse aperfeiçoamento pode conduzir à melhoria no tempo de processamento.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Sociedade Brasileira de Cardiologia, 2008. Disponível em: < <http://www.cardiol.br/>>
- [2] Guyton, Arthur C.; Fisiologia Humana, 6ª Edição. Guanabara Koogan, 1988.
- [3] Prashant Bansod, U.B. Desai; Nitin Burkule. *Automatic Detection of Left Ventricle in Echocardiographic Sequences using Radial Search and Temporal Smoothing*. 15th Intl. Conf. On Digital Signal Processing, 2007. IEEE.
- [4] Gonzalez, Rafael C.; Woods, Richard E.. *Digital Imaging Processin*., 2a Edição. New Jersey, Prentice Hall. ISBN: 0-201-18075-8, 2001.
- [5] Cheng, Jierong; Foo, Say Wei; Krishnan, Shankar M.. *Automatic Detection of Region of Interest and Center Point of Left Ventricle using Watershed Segmentation*. IEEE, 2005.
- [6] Lee, J.S., 1981. Speckle Analysis and Smoothing of Synthetic Aperture Radar Images. *Computer Graphics and Image Processing*, Vol. 17:24-32. Gonzalez, Rafael C.; Woods, Richard E.. *Digital Imaging Processin*, Vol. 17:24-32.
- [7] Chai, Yu-hua; Gao, Li-qun; Lu, Shun; Tian, Lei. *Wavelet-based Watershed for Image Segmentation Algorithm*. IEEE, 2006. Proc. 6th World Congress on Intelligent Control and Automation, China, 2006.
- [8] MATLAB, v. 7.0. Disponível em:<<http://www.mathworks.com/>>. Acesso em 15 setembro de 2007.
- [9] Paccini, Grasiela; d'Ornellas, Marcos C.. *Segmentação de Imagens por Watersheds: Uma Implementação Utilizando a Linguagem JAVA*. Universidade Federal de Santa Maria, Brasil, 2002.
- [10] Sun Microsystems. Disponível em:<<http://www.sun.com>>. Acessado em 2 Agosto 2007.
- [11] JAVA Advanced Imaging API. Disponível em:<<http://java.sun.com/products/java-media/jai/iio.html>>. Acessado em 5 setembro 2007.
- [12] IEEE Xplore. Disponível em:<http://ieeexplore.ieee.org/Xplore>. Acessado durante toda a realização deste trabalho.
- [13] JmatLink – Componente para conexão de aplicações JAVA com o MATLAB. Disponível em:< <http://www.held-mueller.de/JMatLink/>>. Acessado em 10 maio 2007.
- [14] Pressman, Roger. *Engenharia de Software*. 6ª Edição, McGraw-Hill.
- [15] BIDGOOD WD Jr; HORI SC; PRIOR FW; VAN SYCKLE DE. *Understanding And Using DICOM, The Data Interchange Standard For Biomedical Imaging*. Journal American Medical Informatics Association, 1997, pg 199-212.

- [16] Eclipse IDE. Disponível em:<<http://www.eclipse.org>>. Acessado em 12 outubro 2007.
- [17] Digital Imaging and Communication in Medicine - DICOM. The NEMA's Official DICOM Page. Disponível em: <<http://medical.nema.org>>. Acesso em: 10 novembro 2007.
- [18] Rorden, C. *The DICOM Standard*, University of South Carolina, Columbia SC, USA, 7 páginas. Disponível em: <<http://www.sph.sc.edu/comd/rorden/dicom.html>>. Acesso em: 15 janeiro 2008.
- [19] Mendes, Raquel Dias. *Inteligência Artificial: Sistemas Especialistas no Gerenciamento da Informação*. Departamento de Ciência da Informação e Documentação (CID) da Universidade de Brasília (UnB), Ci. Inf., Brasília, v. 26, n. 1, p. 39-45, jan./abr. 1997
- [20] Shin, Il-hong; Kwon, Min-Jeong; Chung, Sung-Taek; Park, HyunWook. *Segmentation and Visualization of Left Ventricle in MR Cardiac Images*. IEEE, 2002. IEEE ICIP 2002.
- [21] Mathworks Inc., Toolbox Image. Disponível em:<<http://www.mathworks.com/>>. Acesso em 15 setembro de 2007.
- [22] Camapum, J. F. *O Algoritmo do Gradiente Multiescalar*. Disponível em:<<http://www.ene.unb.br/~juliana/>>. Acesso em 01/09/2008.
- [23] Facon, J. *Processamento e Análise de Imagens*. Embalse: VI EBAI (Escola Brasileira-Argentina de Informática), 1993.
- [24] Fisher, R. Perkins, S. Walker, A. Wolfart, E. (2003). *Hit-and-Miss Transform*. Disponível em: <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/hitmiss.htm>>. Acesso em 15/08/2008.
- [25] Laine, A., Zong X. *Border Identification of Echocardiograms via multiscale edge detection and shape modeling*, IEEE Int.Conf. on Image Processing, vol. 3, pp. 287 - 290 vol.3, Setembro 1996.
- [26] Dangelo, J. G., Fattini, C. A. *Anatomia Humana Básica*, Livraria Atheneu Editora (1995).
- [27] Sobotta, Johannes. *Atlas de Anatomia Humana*. Vol I, Ed Guanabara, 21ª Edição, 2006.
- [28] Sobotta, Johannes. *Atlas de Anatomia Humana*. Vol II, Ed Guanabara, 21ª Edição, 2006.
- [29] Cornell, G., Horstmann, S. *Core JAVA – Vol I – Fundamentals*. 5ª Edição. Prentice Hall. ISBN: 0-13-089468-0, 2000.
- [30] Cornell, G., Horstmann, S. *Core JAVA – Vol II – Advanced Features*. Prentice Hall. ISBN: 0-13-092738-4, 2001.

- [31] Pigoski, Thomas M. *Practical Software Maintenance: Best Practices for Managing Your Software Investment*. Wiley Computer Publishing, 1996.
- [32] Torelli, J. C. *Implementação Paralela da Transformada de Distância Euclidiana Exata*. Dissertação Mestrado, ICMC-USP, 2005.
- [33] Maia, L. P. *Multithread*. Instituto de Matemática da UFRJ/NCE, Departamento de Ciência da Computação, 1998
- [34] Feigenbaum, H. *Ecocardiografia*. Guanabara Koogan, 2007, 6ª Edição. ISBN: 8527712484.
- [35] Carmo, M. do; Camapum, Juliana; Espinoza, Bruno. *Semi-Automatic Detection of the Left Ventricular Border*. UNB, 2008.
- [36] Silva, A. O (2003). *Transformada de watershed para a segmentação de imagens médicas da região pélvica*. Dissertação de Mestrado, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 129p.
- [37] Macchiavello, B. L. E. . *Segmentação Automática do Ventriculo Esquerdo em Imagens de Ecocardiografia Bidimensional*. 2004. Dissertação de Mestrado, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF.
- [38] Moreno, R. A. (2005). *Visualizador Contextual de Imagens Médicas*. Tese de Doutorado em Engenharia Elétrica, Universidade de São Paulo, Escola Politécnica, 152p.
- [39] Reis, M. C. (2007). *Detecção Semi-Automática da Borda do Ventrículo Esquerdo*. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM-310/07, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 114p.
- [40] JUDE. Disponível em:<<http://jude.change-vision.com/jude-web/index.html>>. Acessado em 10 outubro 2008.
- [41] BOUML. Disponível em:< <http://bouml.free.fr/>>. Acessado em 5 julho 2008.
- [42] Microsoft VISIO. Disponível em:< <http://www.microsoft.com/en/us/default.aspx>>. Acessado em 10 outubro 2008.

ANEXO A

PADRÕES UTILIZADOS

A.1 - INTRODUÇÃO

A seguir serão descritos os padrões adotados no trabalho. Formatos de arquivos, estruturas e outras definições consagradas no mercado e no meio acadêmico são listados para explicitar sua adoção e utilização nesta implementação.

A.2 – ARQUIVOS DICOM

A.2.1 - Histórico

O Padrão DICOM (*Digital Imaging and Communications in Medicine*) é um padrão mundialmente reconhecido que define as formas de envio, armazenamento e exibição de imagens médicas capturadas pelos equipamentos utilizados pelos profissionais de saúde para a realização de diagnósticos. Esse padrão foi definido pela NEMA (*National Electrical Manufacturers Association*), e um dos objetivos é definir o formato dos arquivos armazenados.

Uma crise na interoperabilidade entre equipamentos de imagens médicas alastrou-se por volta dos anos de 1980. O ACR (Instituto Americano de Radiologia) e a NEMA, que produziam equipamentos de radiologia digital, trabalharam juntos no desenvolvimento de um padrão a ser adotado por todos os fabricantes. Estava surgindo o DICOM.

O padrão DICOM é definido pela ACR-NEMA (2000), sendo composto de 16 partes e mais de 50 suplementos[38]. A primeira versão, publicada como DICOM 1.0 foi extremamente detalhista, chegando ao ponto de especificar o tipo de conector físico a ser utilizado na conexão dos equipamentos médicos. Com este grau de detalhamento, esta especificação obrigaria os fabricantes a se adequarem para atendê-la. Isso fez com que este padrão não se tornasse sólido no mercado.

A versão DICOM 2.0 também não teve ampla disseminação. Entretanto, serviu para que grandes fabricantes, como Siemens, Picker e General Electric, percebessem que a

interoperabilidade aumentaria suas vendas, à medida que possibilitaria a integração entre departamentos e também a telerradiologia.

Foi em 1991 que surgiu o maior evento na viabilização da interoperabilidade de equipamentos de imagens médicas, o DICOM 3.0. Esta versão deu ênfase à implementação do uso de redes, abriu a possibilidade para a convergência de outros padrões, além de fornecer informações a modalidades radiológicas tais como tomografia, ultra-sonografia e ressonância magnética. Para as modalidades fora do domínio da radiologia foi implementado o módulo chamado captura externa (EC). Ele permitia que imagens genéricas fossem armazenadas como arquivo DICOM. Assim, este módulo assinalava o desejo do padrão vir a suportar qualquer tipo de imagem médica, em suplementos que seriam criados posteriormente.

Desde 1995, as principais modalidades de diagnóstico por imagem começaram a ser padronizadas. Esta lista inclui tomografia computadorizada (CT), ressonância magnética (MRI), radiografia computadorizada, ultra-sonografia, medicina nuclear, radiofluoroscopia, angiografia e captura de vídeo digitalizado, também o suplemento luz visível (VL), útil em endoscopia, microscopia e fotografia.

Atualmente o DICOM é mantido por um grupo de várias especialidades que coordena e controla o uso e aperfeiçoamento do padrão, o Comitê do Padrão DICOM[17].

A.2.2 - Estrutura

O padrão DICOM[15] fornece um protocolo não proprietário para troca de dados, definição de formato de arquivos digitais, além da estrutura do arquivo para imagens biomédicas, tendo inserido em sua estrutura as informações correlacionadas com a imagem. Este padrão é internacionalmente utilizado para interconexão de equipamentos médicos, da área diagnóstica e terapêutica, que capturam, armazenam e transferem imagens.

Interfaces DICOM estão disponíveis para conexão de qualquer combinação das seguintes categorias de equipamentos de imagem digital:

- 1- equipamentos de aquisição de imagens, por exemplo, tomógrafos em geral, ultra-sonografia e outros;
- 2- programas para arquivamento de imagens;
- 3- estações de trabalho para processamento e visualização de imagens e
- 4- dispositivos de impressão, como impressora em papel e filme radiológico.

A sua estrutura é orientada a objetos, onde o arquivo carrega, em sua estrutura, vários objetos com seus valores. Esses valores trazem informações tanto do arquivo quanto do paciente e diagnóstico a que ele se refere.

A.2.3 – Manipulação dos arquivos DICOM

Os arquivos DICOM guardam um conjunto de imagens de maneira seqüencial. Cada imagem compõe um *frame* e cada um destes possui uma matriz que guarda as informações de cor ou nível de cinza de cada *pixel* (ponto) formador do frame.

Os arquivos de imagem no padrão DICOM são identificados através da extensão “.dcm”. A sua estrutura é de uma matriz de 4 dimensões, onde a quarta dimensão indica o número do *frame* (Figura A.1). Como o software implementado neste trabalho utiliza as bibliotecas do MATLAB, o acesso aos arquivos de imagens é feito através do comando *dicomread*, que retorna uma matriz de 4 dimensões com as imagens do arquivo dcm.

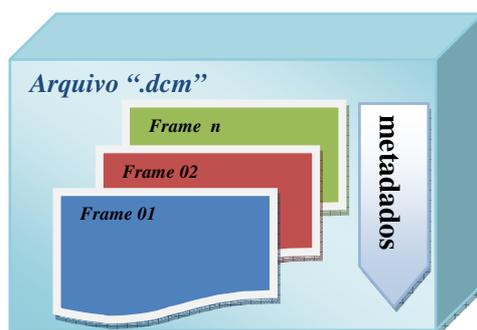


Figura A.1 – O arquivo .dcm com seus quadros e as informações adicionais (metadados)

- A quantidade de *quadros* existentes dentro do arquivo DICOM pode variar de acordo com o equipamento de captura utilizado e de suas configurações. Os arquivos que serão utilizados neste trabalho, obtidos através de um equipamento de ultra-som capaz de realizar a ecocardiografia transtorácica, possuem 90 quadros que, consecutivamente, retratam alguns poucos ciclos cardíacos do paciente (aproximadamente 3 ciclos completos).
- Esse conjunto de *quadros* será reduzido através do agrupamento de imagens baseado na similaridade das mesmas. A matriz utilizada nesta junção será explicada posteriormente.

A.3 – ECOCARDIOGRAFIA

A.3.1 – Introdução

A avaliação do volume, da massa e da função mecânica do ventrículo esquerdo, como bomba ou como músculo, requeria o uso de métodos como o cateterismo cardíaco, com os riscos, desconfortos e despesas relacionados a este procedimento. Além disto, é impraticável o estudo rotineiro ou seriado dos pacientes com a frequência necessária para a avaliação de efeitos de drogas, momento cirúrgico, técnicas cirúrgicas, ou mesmo para o melhor conhecimento da história natural de certas cardiopatias. A ecocardiografia surge como um método não-invasivo, que utiliza ultra-som (2,0-7,0 MHz), capaz de fornecer dados valiosos a respeito do tamanho e da função sistólica e diastólica do ventrículo esquerdo, sem apresentar os mesmos inconvenientes do cateterismo cardíaco.

A ecocardiografia continua a crescer. Em 1954 na Suécia, que Edler e Hertz foram os primeiros a observar os movimentos das estruturas cardíacas por ultra-sonografia[39]. E em 1963 nos Estados Unidos, a técnica de ultra-som foi utilizada pela primeira vez para examinar o coração por Joyner e Reid. Mas um dos maiores avanços ocorreu em meados dos anos 70, com o desenvolvimento da ecocardiografia bidimensional (2D)[37].

A ecocardiografia se divide em transtorácica, transesofágica e fetal e apresenta atualmente diversas modalidades, iniciando-se com a ecocardiografia unidimensional, surgindo a seguir o método bidimensional, posteriormente as modalidades unidimensional e bidimensional com Doppler convencional e finalmente o Doppler colorido e a ecocardiografia 3D.

A.3.2 – Ecocardiografia Transtorácica

O instrumento utilizado para gerar uma imagem utilizando-se um ultra-som é conhecido como um ecógrafo. Os componentes essenciais incluem o transdutor que em contato com o tecido em exame, emite e recebe o ultra-som. O transmissor regula a emissão do ultra-som pelo transdutor através de um regulador (“timer”) que controla a duração e a frequência dos pulsos ultra-sônicos emitidos pelo transdutor. Os ecos refletidos ou impulsos são processados de modo que possam ser visualizados num tubo de raios catódicos ou osciloscópio[39].

Há três modos de apresentação dos ecos na tela do osciloscópio. O modo A (A = amplitude), onde o eco é representado como um pico, e o modo B (B = brilho). Em ambos os casos a apresentação se faz tendo em vista apenas a distância do transdutor e a capacidade refletora das estruturas. Devido ao fato de que o coração é um objeto em movimento, pode-se registrar o movimento através da introdução do tempo, como sendo uma segunda dimensão. Neste caso a forma de apresentação osciloscópica é denominada *modo M* (M = movimento) e é a forma utilizada na ecocardiografia unidimensional[39].

A.3.3 – Ecocardiografia Doppler

A ecocardiografia Doppler fundamenta-se no chamado efeito Doppler descrito por Christian Jhoann Doppler em 1842. Este método nada mais é do que a análise da variação da frequência ultra-sônica encontrada nos ecos refletidos pelas hemácias. O efeito Doppler permite não só avaliar se as hemácias, e, portanto o sangue se está aproximando ou se afastando da fonte de origem da energia ultra-sônica, como permite também saber a sua velocidade, já que a variação da frequência é proporcional a esta velocidade. A velocidade do sangue pode ser deduzida pela equação A.1 (Equação Doppler)[39]:

$$V = \frac{C \times \Delta f}{2f \times \cos \theta} \quad (\text{A.1})$$

onde:

- V = velocidade do sangue.
- C = velocidade no ultra-som do meio, (aproximadamente 1.540m/s).
- F = frequência do ultra-som emitido.
- Δf = diferença de frequência entre o som emitido e o refletido.
- θ = ângulo formado entre a direção do fluxo sanguíneo e a direção do feixe ultra-sônico.

Como visto anteriormente o Doppler intracardíaco registra o movimento de sangue dentro do sistema cardiovascular. Esta técnica estuda a função ventricular por verificar alterações no padrão dos fluxos sanguíneos. Assim, a maneira como o sangue é ejetado na aorta pode ser um reflexo da função ventricular esquerda. Vários parâmetros podem

distinguir um ventrículo esquerdo normal de um com uma miocardiopatia, como a velocidade máxima de fluxo aórtico ou a integral do fluxo aórtico ou a velocidade de ascensão do fluxo aórtico[39].

Basicamente há dois tipos de ecocardiografia Doppler, o contínuo e o pulsátil. Um terceiro tipo, o de alta frequência de repetição (“high PRF”) raramente é utilizado. O sistema de onda pulsátil veio solucionar os problemas referentes à onda contínua. Era impossível conhecer onde estavam localizados os alvos individuais em relação ao transdutor e de determinar se havia mais do que um alvo em movimento. A introdução do Doppler pulsado foi um avanço importante que tornou a ecocardiografia Doppler um instrumento clinicamente valioso. Porém a onda pulsátil apresenta desvantagem em relação a medida da velocidade que é limitada.

A.3.4 – O Exame Ecocardiográfico

A capacidade de obter o registro ecocardiográfico de alta qualidade é, provavelmente, o fator mais importante para se determinar a utilidade do exame ecocardiográfico[34]. A realização deste exame deve seguir a técnica apropriada, conforme as normas do Departamento de Ecocardiografia da Sociedade Brasileira de Cardiologia.

Na realização deste exame emprega-se o ecocardiógrafo, que é o aparelho utilizado na obtenção de imagens ultra-sônicas do coração. As unidades de ultra-som devem ser equipadas com os instrumentos básicos que permitam realizar no mínimo exames unidimensionais e bidimensionais.



(a)

(b)

Figura A.2 – Duas das quatro posições padronizadas para colocar o transdutor transtorácico: janela supra-esternal (b) e janela sub-costal (a)[34].

Infelizmente, os exames devem ser realizados de acordo com as condições de cada paciente. De acordo com especialistas é preferível obter o ecocardiograma com o paciente

em semidecúbito lateral esquerdo (cerca de 30°) com a parte superior do tronco ligeiramente elevada, estando o pescoço situado à esquerda do paciente. Nesta posição, o coração se desloca para detrás do esterno ficando mais acessível para o estudo ecocardiográfico[39].

O exame ecocardiográfico necessita da energia ultra-sônica obtida utilizando-se cristais piezelétricos contidos no transdutor. O transdutor a ser empregado é selecionado de acordo com a idade do paciente. Neste caso, em adultos usa-se geralmente transdutores de 3,25 MHz, focalizados a 7cm, já em crianças é utilizado transdutores de 5 a 7MHz. Existem quatro posições padronizadas para o transdutor: janelas paraesternal, apical, subcostal e supra-esternal. A Figura A.2 mostra o registro do ecocardiograma nas posições supra-esternal e subcostal. Outros fatores importantes para um resultado satisfatório do exame são: a otimização dos controles de ganho e do monitor do equipamento, a realização dos estudos em modo M, bidimensional e Doppler (pulsátil, contínuo e colorido), de acordo com protocolos específicos da instituição, a utilização apropriada das técnicas de Doppler (alinhamento do feixe) e uma aquisição correta dos cortes ecocardiográficos.

A.4 – IMAGENS INDIVIDUAIS

A.4.1 - Visão Geral

As imagens operadas pelo *software* estarão sendo lidas, gravadas e exibidas como arquivos BITMAP (.bmp), formadas por um conjunto de pontos individuais chamados *pixels*, codificados sob 24 bits (8 bits para cada componente primária de cor: RGB sem os valores negativos) sem compressão. O objetivo da utilização deste tipo de arquivo é minimizar a deformação e/ou perda de informação pela forma de representação da imagem.

Esse tipo de imagem não possibilita a aplicação de *zoom* sem a sua deformação, pois ao contrário do formato vetorial, a ampliação é aplicada para cada *pixel* sem levar em consideração a sua vizinhança ou pontos de referência.

Apesar das imagens manipuladas pelo *software* serem a representação de ultrasonografias, de conteúdo em nível de cinza (256), optou-se pela representação em imagens de 24 bits para eliminar as perdas durante as operações de converção, tratamento e segmentação. O inconveniente de manipular este tipo de imagem é o tamanho das mesmas,

exigindo espaço em disco e memória maiores para ter-se um bom rendimento do computador executando o algoritmo.

A representação das imagens digitais é feita bidimensionalmente através de uma matriz $n \times m$, onde n e m indicam as dimensões da imagem em largura e comprimento. Cada posição da matriz armazena o nível de cinza do *pixel* em questão.

A.5 – PROGRAMAÇÃO ORIENTADA A OBJETOS

A.5.1 - Visão Geral

A Programação Orientada a Objetos (POO), surgiu como novo paradigma de programação como uma fase posterior à programação estruturada (C, Pascal, FORTRAN, etc.), tendo como algumas linguagens representantes o SmallTalk, o JAVA, o C++ e o C#. Neste novo cenário, a programação busca elevar o nível de compreensão dos códigos, aproximando-se do mundo real, transformando a codificação em uma tarefa mais intuitiva[14].

Na POO, o mundo real será representado em código através de classes e objetos. Tal qual temos uma pessoa no mundo real, passamos a ter uma representação lógica desta mesma pessoa em nosso programa através de um objeto criado a partir de uma classe. Esse objeto terá, também, funcionalidades e atributos que representam as tarefas e características, respectivamente, que essa pessoa realiza ou possui.

Os atributos de um objeto nada mais são do que as características e valores que este objeto carrega. Na Engenharia de Software diz-se que os atributos de um objeto são aquilo que ele sabe e as funcionalidades, ou métodos, são aquilo que ela pode fazer. A interação entre diversos objetos ocorre através da troca de mensagens, representação das chamadas aos métodos de um dado objeto.

A.5.2 - Conceitos

- Uma Classe descreve a estrutura sobre a qual os objetos serão criados, formada pelos atributos, ou seja, os campos capazes de armazenar valores ou o que a classe sabe, e pelos métodos que representam as tarefas, atividades ou funções que a classe é

capaz de realizar ou o que a classe faz. Uma classe pode ser vista, de maneira abstrata, como um conjunto de objetos, já que é a partir dela que os criamos.

- O Objeto, também chamado de instância de uma classe, possui a representação concreta da classe com os valores de seus atributos definidos. Estes objetos podem ter os valores de seus atributos alterados, quando não há uma restrição, e executam os seus métodos através da troca de mensagens. Eles podem possuir valores de tipos simples, como inteiros e fracionários, e até mesmo outros objetos em sua estrutura.

- O Encapsulamento é um recurso da POO essencial para manter a integridade dos valores dos atributos de objetos, pois limita o acesso a estes atributos exclusivamente através de métodos, denominados métodos de acesso, impedindo a alteração direta de um determinado valor carregado por um objeto. O controle deste acesso é definido através dos níveis de visibilidade identificados por palavras-chave.

- A Herança é um recurso largamente empregado pois evita o retrabalho e a repetição de código. Através dela, uma determinada classe pode herdar todos os métodos e atributos de uma classe-mãe, ou seja, é como se toda a estrutura da classe-mãe, ou superclasse[14], estivesse repetida dentro da classe que a herda. Em consequência, os objetos criados a partir desta classe também herdam toda a estrutura.

- O Polimorfismo permite que a identificação do método chamado seja feita em tempo de execução. Isso permite que um mesmo método, em objetos diferentes, possa ter comportamentos ou execuções distintos.

- Objetos Persistentes são objetos que não têm seu estado alterado ou perdido com o encerramento do programa ou limpeza da memória da máquina. O estado do objeto é composto por todos os valores de seus atributos. Para que um objeto seja considerado persistente, há a necessidade de existir um mecanismo que armazene este estado, os valores dos atributos, em alguma estrutura física ou lógica, tal como um CD, um DVD ou um Banco de Dados.

- Em algumas linguagens, como é o caso do JAVA, há um componente que gerencia o uso de memória, o *Garbage Collector*, que exclui os objetos que não estão mais sendo utilizados pelo programa. Com isso, caso haja a necessidade de manter o estado destes objetos preservado, temos que torná-los persistentes.

A.6 – MATLAB

A.6.1 - Visão Geral

O MATLAB é um software especialista criado pela MathWorks[8], altamente interativo, voltado à realização de cálculos numéricos e outras funcionalidades ligadas à engenharia, matemática e estatística. Sua interface gráfica é de uso simplificado e reúne diversas funções que, de outra maneira, seriam de complexa realização e demoradas.

Além de funções prontas, o MATLAB possui uma linguagem própria que permite a criação de algoritmos matemáticos para simulação e realização de cálculos. Estes programas criados seguem uma estrutura própria do software e só podem ser executados sobre o MATLAB.

Ele é um sistema interativo cujo elemento básico de comunicação é uma matriz que não possui dimensões definidas. Este sistema permite a realização de cálculos numéricos de alta complexidade em janelas de tempo que seriam inviáveis de implementação utilizando outra linguagem de programação, como o C ou FORTRAN. Além disso, a notação utilizada pelo MATLAB, ao contrário de outras linguagens, é exatamente igual à utilizada pela Matemática.

A.6.2 – Manipulação dos Arquivos

O ambiente fornecido por este *software* traz consigo uma linguagem específica e recursos que facilitam a manipulação de matrizes e outras funções matemáticas. Os arquivos manipulados por este ambiente estão no formato texto e são gravados com a extensão “.m”. Esta ferramenta possibilita ainda a interação com outros programas que estejam escritos, inclusive, em outras linguagens.

O algoritmo implementado que foi proposto por este trabalho não manipula arquivos do MATLAB, mas como há interações com a sua API, o uso da linguagem adequada foi necessário para que o programa executasse corretamente.

ANEXO B

FUNDAMENTOS TEÓRICOS

B.1 - INTRODUÇÃO

A tarefa realizada sobre as imagens de maneira a identificar, automaticamente, o ventrículo esquerdo é formada por um conjunto de operações morfológicas. Este processamento, que envolve a análise e a manipulação dos *pixels*, visa melhorar a qualidade das imagens através do realce ou da atenuação e remoção de ruídos, de modo a facilitar a segmentação.

Obter informações a partir de imagens exige que estas sejam de boa qualidade para poder ser avaliada, fator incerto por consequência dos diferentes recursos de *hardware* existentes, provocando a inserção de ruídos.

Este anexo apresenta diversas técnicas de processamento de imagens, que foram utilizadas na realização deste trabalho de modo a minimizar estas deficiências das imagens manipuladas. Estas técnicas resume-se em segmentação de imagens e morfologia matemática binária.

B.2 - HISTÓRICO

Na década de 20, onde a transmissão de dados começava a ser feita a longas distâncias através de cabos submarinos que cruzavam milhares de quilômetros submersos, o surgimento de ruídos e até mesmo a perda parcial das informações transmitidas eram freqüentes. Isso exigiu que métodos e ferramentas surgissem de modo a tratar essas informações, maximizando a sua recuperação.

Quando esses dados transmitidos são imagens digitais, este tratamento deve ser criterioso de modo a preservar a informação transmitida. Inicialmente, o uso dessas imagens era feito de maneira precária, tendo sua representação em apenas dezesseis níveis de cinza e não eram tratadas devido a não existência de computadores para tal. Com a inserção dos conceitos de memória para armazenamento de programas e dados, e das instruções computacionais condicionais, houve o surgimento dos primeiros computadores

capazes de realizar cálculos que caracterizavam na melhoria das imagens recebidas, tornando-se o embrião do processamento digital de imagens.

A invenção do transistor em 1948, a invenção de linguagens de alto nível tais como COBOL e FORTRAN nas décadas de 50 e 60, e a invenção do circuito-integrado(CI) em 1958 são algumas evoluções na área da computação e eletrônica que contribuíram com o surgimento de um cenário favorável para o surgimento de tecnologias mais bem elaboradas que possibilitassem um incremento no processamento digital de imagens.

O governo dos EUA fez uso desse tipo de processamento, mais intensamente, em 1964 com o recebimento das imagens captadas da superfície lunar. O trabalho realizado buscava identificar detalhes do terreno da Lua tais como crateras, marcas de minerais e até mesmo sinais da existência de água. Posteriormente, essa tecnologia foi utilizada na análise de imagens dos terrenos durante a 2ª Guerra Mundial para obtenção de informações militares estratégicas.

Atualmente, a aplicação do processamento digital de imagens reveste-se de significativa importância na área médica, auxiliando na obtenção dos diagnósticos e melhorando os resultados dos tratamentos.

B.3 – IMAGEM DIGITAL

Podemos definir imagem como a função bidimensional de intensidade da luz $f(x, y)$, onde x e y denotam as coordenadas espaciais, e o valor de f em qualquer ponto (x, y) é proporcional à intensidade do brilho (ou níveis de cinza) da imagem naquele ponto[4].

A intensidade da luz $f(x, y)$ pode ser modelada através das componentes iluminação e reflectância, que respectivamente são representadas por $i(x, y)$ e $r(x, y)$, por intermédio do seu produto:

$$f(x, y) = i(x, y)r(x, y) \quad (\text{B.1})$$

onde, $0 < f(x, y) < \infty$, $0 < i(x, y) < \infty$ e $0 < r(x, y) < 1$.

Ao representar com valores discretos a $f(x, y)$ obtemos uma imagem digital. Esta imagem é uma matriz cujos índices de linhas e colunas identificam um ponto na imagem e

o correspondente valor do elemento da matriz identifica o nível de cinza naquele ponto. Cada um destes elementos pontuais que compõem a imagem digital são chamados de *pixels* [4].

As imagens digitais são representadas em quatro modelos distintos[21]:

- imagem indexada;
- imagem por intensidade;
- imagem binária;
- imagem RGB.

Imagens indexadas requerem duas matrizes, uma de valores numéricos e uma segunda matriz formada pelas cores, R(Red – Vermelho), G(Green – Verde) e B(Blue – Azul). A cor de cada *pixel* na imagem é determinada usando o valor correspondente da primeira matriz como um índice na matriz de cores.

Uma imagem de intensidade é uma matriz de dados cujos valores representam intensidades dentro de alguma escala pré-definida. Como exemplo, podemos citar que se os elementos de intensidade forem do tipo uint8 seus valores estarão no intervalo [0, 255]. A Figura B.1 exemplifica estes tipos de imagem.

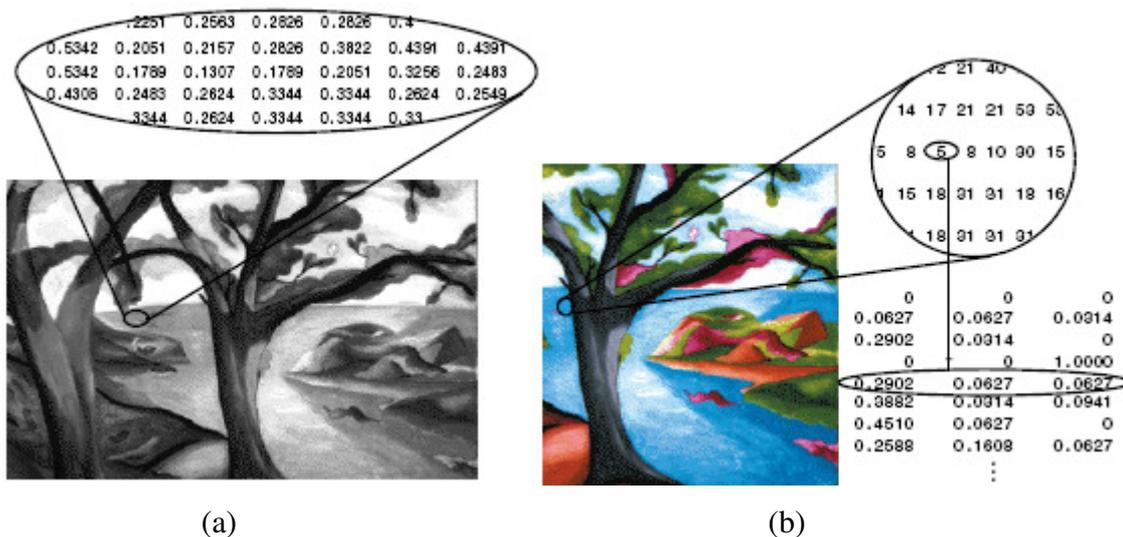


Figura B.1- Representação de imagens digitais (a) modelo intensidade e (b) indexada.

Em uma imagem binária cada *pixel* assume o valor discreto zero ou um. O modelo de representação RGB (Red-Green-Blue) se baseia num sistema de coordenadas cartesianas tridimensional onde cada um dos eixos representa uma das cores primárias aditivas: vermelho, verde e azul [4]. A cor de cada *pixel* é determinada pela combinação

das intensidades de cada componente RGB na posição do *pixel*. Exemplos de imagens RGB e binária são apresentadas na Figura B.2.

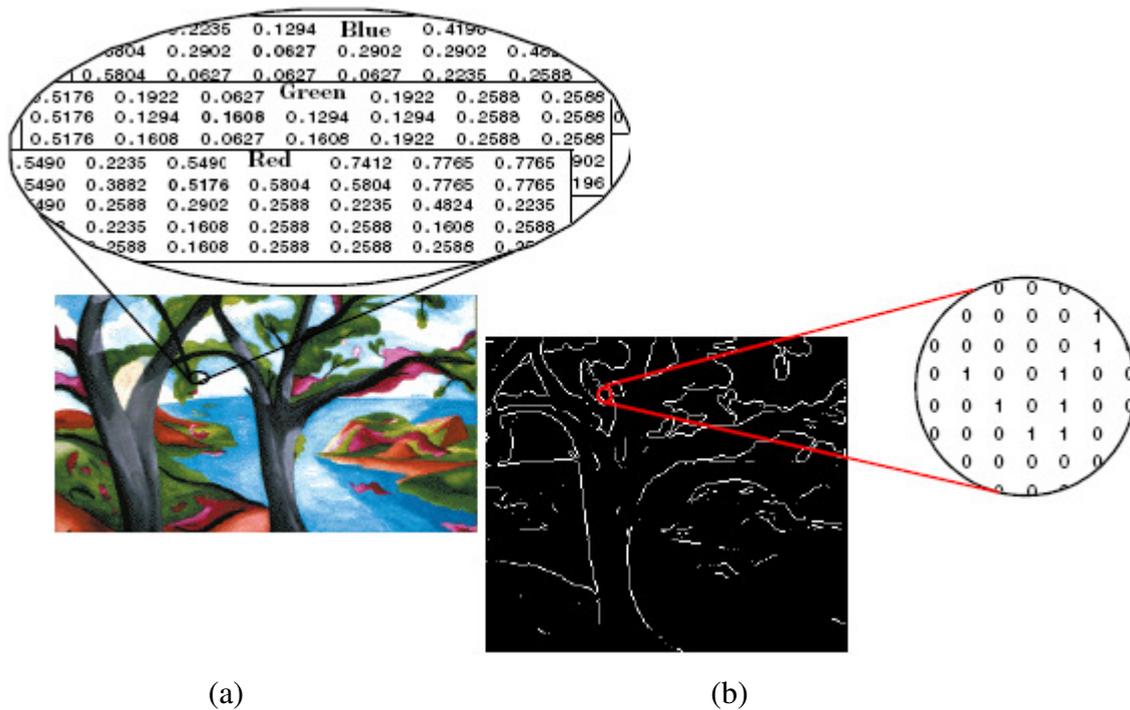


Figura B.2- Representação de imagens digitais (a) modelo RGB e (b) binária.

Além dessas diferentes representações, as imagens digitais também diferem no número de bits por *pixel* (amostragem) e em relação ao tipo e nível de compactação (com ou sem perdas) utilizada.

Nem sempre é possível realizar a conversão perfeita entre os formatos de imagens. Por isso deve-se utilizar o mínimo possível da conversão de formatos para que não sejam introduzidos erros na representação da imagem ou de sua paleta de cores[4].

Os diferentes formatos de imagens se referem à forma como uma imagem é escrita em um arquivo de computador[22], causando variações em suas características, tais como tamanho do arquivo e informações de cores.

Existem vários formatos de arquivos, dentre os quais, os mais utilizados são: RAW, PSD (PhotoShop Image), BMP (Bitmap), GIF (Graphics Interchange Format), PNG (Portable Network Graphics), TIFF (Tagged Image File Format), JPEG (Joint Photographic Expert Group) e JPEG 2000. Cada formato possui uma forma de uso mais adequada.

B.4 – OPERAÇÕES MORFOLÓGICAS

A morfologia matemática tem como princípio a extração de informações relativas à geometria e à topologia de um conjunto desconhecido de uma imagem pela transformação através de um outro conjunto completamente definido, chamado elemento estruturante[23].

Na morfologia binária, uma operação morfológica binária é determinada a partir da vizinhança em torno de um ponto central, analisando a configuração de pontos pretos e brancos através da execução de um determinado algoritmo.

B.4.1 – Elemento Estruturante

O elemento estruturante é uma imagem (ou conjunto), menor que a imagem original, usado para investigar a imagem em estudo. O formato e o tamanho do elemento estruturante possibilitam testar e quantificar de que maneira o elemento estruturante “está ou não contido” na imagem. Marcando os resultados das posições onde o elemento estruturante está contido na imagem, tem-se uma primeira resposta sobre a estrutura geométrica das entidades dessa imagem.

A forma geométrica do elemento estruturante deve ser adaptada às propriedades geométricas dos objetos a serem processados. A escolha do elemento estruturante é fundamental na utilização da morfologia matemática, uma vez que cada elemento estruturante fornece uma aparência nova ao objeto[23]. A Figura B.3 mostra alguns formatos de elementos estruturantes.

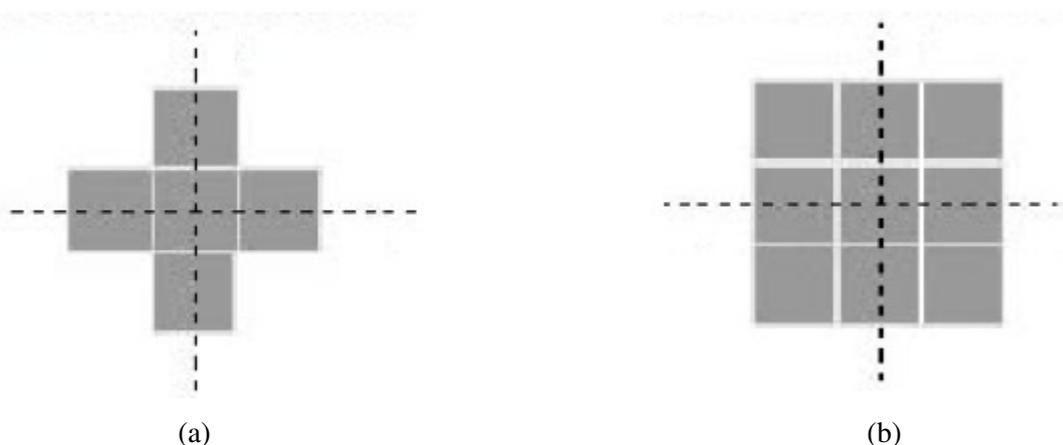


Figura B.3 - Exemplos de elementos estruturantes, (a) com conectividade de 4 e (b) com conectividade de 8.

B.4.2 – Dilatação e Erosão

A dilatação e a erosão morfológicas binárias são definidas, do ponto de vista geométrico, como operações que, respectivamente, reduzem e expandem um conjunto. Os pilares da morfologia matemática são as operações básicas de dilatação e erosão, a partir das quais, por composição, é possível realizar muitos outros operadores, como a abertura e o fechamento[23].

A dilatação de um conjunto A por um elemento estruturante B , denotado por $A \oplus B$, é definida como:

$$A \oplus B = \{x \mid (\hat{B})_x \cap A \neq \emptyset\} \quad (\text{B.2})$$

Dessa forma, a dilatação de uma imagem A , de dimensões $n \times m$, por um elemento estruturante B , é a reflexão de B em torno de sua origem (transposta de B) e seu deslocamento sobre a imagem A , resultando no conjunto de todos os pontos x de B tal que a interseção de B com A possua, ao menos, um elemento não nulo[4].

Essa transformação causa um aumento das partes que compõem a imagem, com a união das que estiverem próximas, arredondamento de bordas retas e preenchimento de buracos. Neste tipo de operação, o número de componentes na imagem pode diminuir pela possível junção. Na Figura B.4 há um exemplo da aplicação de dilatação sobre uma imagem 4×4 .

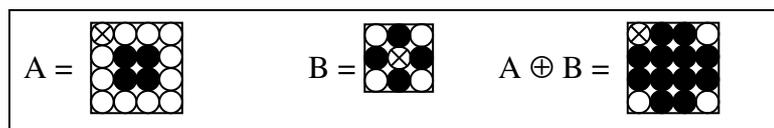


Figura B.4 – Exemplo de dilatação da imagem A com o elemento estruturante B

Podemos definir a erosão de A pelo elemento estruturante B , denotada por $A \ominus B$, como:

$$A \ominus B = \{x \mid (B)_x \subseteq A\} \quad (\text{B.3})$$

O que significa que a erosão de A por B é o conjunto de todos os pontos x , tais que B quando transladado por x , fique contido em A [4]. Essa transformação causa uma redução dos objetos que compõem a imagem, com a exclusão de todas as partes menores

que o elemento estruturante. Pode-se utilizar a erosão para reduzir ruídos e separar objetos ligados. Na Figura B.5 temos um exemplo de aplicação da erosão, com a redução do objeto gráfico:

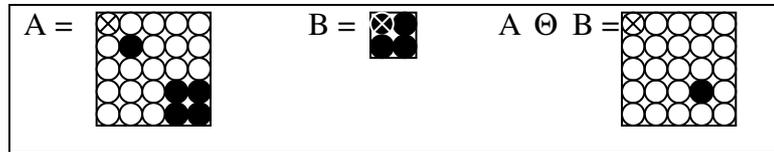


Figura B.5 – Exemplo de erosão da imagem A com o elemento estruturante B

O algoritmo implementado fará muito uso de elementos estruturantes quadrados de tamanho 3×3 e 5×5 , pois usar um elemento de dimensões muito grandes é inadequado para eliminar ruídos na imagem e separar objetos por não poder ser inserido no objeto em suas fronteiras.

B.4.3 – Abertura e Fechamento

A partir da combinação das operações básicas de dilatação e erosão de uma imagem, podemos realizar a abertura e o fechamento da mesma. A abertura geralmente suaviza o contorno de uma imagem, quebra istmos estreitos e elimina profusões finas. O fechamento também tende a suavizar os contornos, mas em oposição à abertura, geralmente funde as quebras em golfos finos, elimina pequenos buracos e preenche fendas em um contorno[4].

Podemos definir a abertura de um conjunto A por um elemento estruturante B , denotada por $A \circ B$, é definida como:

$$A \circ B = (A \ominus B) \oplus B \quad (\text{B.4})$$

Assim, a abertura de A por B é a erosão de A por B seguida da dilatação desse conjunto erodido pelo mesmo elemento estruturante B . A Figura B.6(c) mostra os efeitos da operação de abertura. O conjunto resultante é mais regular e perde os detalhes de contorno. A operação de abertura aplanar os contornos pelo interior, separa elementos e elimina pequenas partículas inferiores ao tamanho do elemento estruturante[23].

Essa transformação, quando utilizados elementos estruturantes horizontais, é capaz de eliminar ruídos horizontais na image original, atuando como um filtro. O similar ocorre

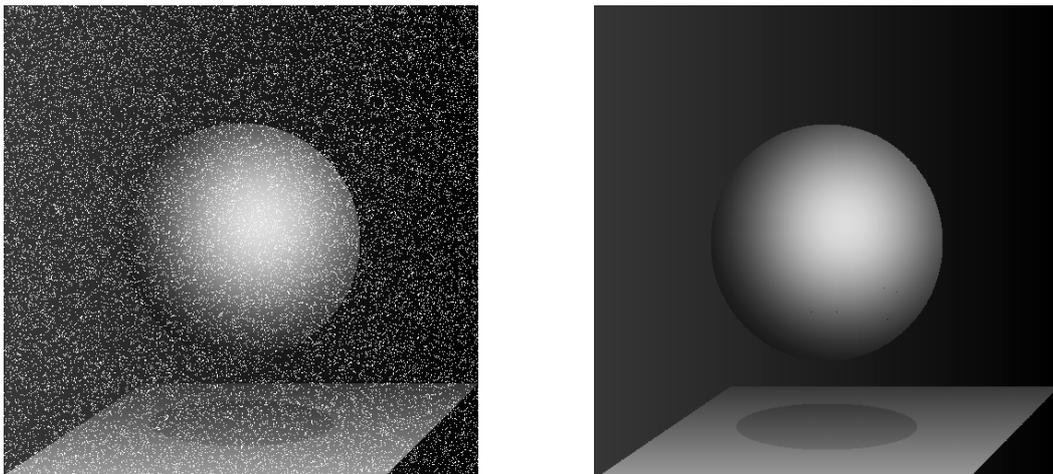
para elementos verticais. A área dos objetos são reduzidas e as bordas são suavizadas de acordo com o elemento estruturante utilizado. As Figuras B.6 e B.7 exemplificam a aplicação deste tipo de filtro.



(a)

(b)

Figura B.6 – Imagem original (a) e após a operação da abertura (b).



(a)

(b)

Figura B.7 – Imagem original com ruído (a) e após a operação da abertura (b).

A operação de fechamento de um conjunto A por um elemento estruturante B , denotada por $A \bullet B$, é definida como:

$$A \bullet B = (A \oplus B) \ominus B \quad (\text{B.5})$$

Significando que o fechamento do conjunto A pelo elemento estruturante B consiste da dilatação de A por B , seguido da erosão do resultado por B . A Figura B.8(d) mostra os efeitos da operação de fechamento. A operação de fechamento tem como efeito

o preenchimento de buracos no interior das partículas, conectando as partículas próximas e suavizando as fronteiras pelo exterior. Um conjunto fechado é também menos rico que o conjunto inicial[23]. A Figura B.8 mostra os efeitos das operações de abertura e fechamento.

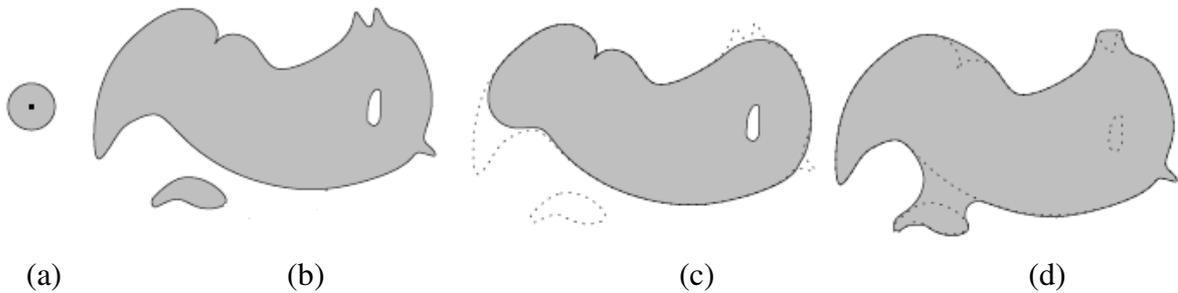


Figura B.8 – Exemplos de Abertura e fechamento. (a) elemento estruturante, (b) imagem original, (c) abertura e (d) fechamento[22].

Quando queremos eliminar pequenos vazios na imagem, aplicamos uma dilatação, mas esta operação engorda a imagem, tornando-a maior. Para evitar isso, aplica-se o fechamento ao invés da simples dilatação, com isso essa distorção é corrigida, a fronteira do objeto é suavizada, buracos são preenchidos e componentes isolados são conectados. Essa transformação é dual em relação à abertura.

Da mesma forma que a abertura, a transformação de fechamento atua como um filtro. Na Figura B.9 temos o resultado da aplicação do fechamento sobre uma imagem. Observa-se que a transformação eliminou algumas falhas e reduziu os buracos internos da imagem, sem, contudo, aumentá-la significativamente.

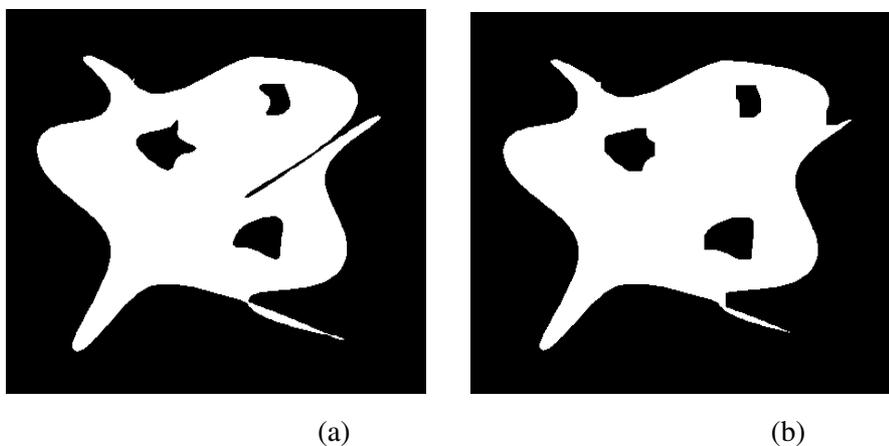


Figura B.9 - Imagem original (a) e após a operação da fechamento (b).

B.4.4 – Afinamento

O afinamento é uma operação na qual imagens binárias são reduzidas a ponto de representar o objeto com a espessura de apenas um *pixel*. O propósito do afinamento é reduzir os componentes da imagem de forma que reste apenas a informação essencial. A maioria dos algoritmos de afinamento baseia-se em repetidas remoções de camadas de *pixels*, até que não existam mais camadas que possam ser removidas[4].

A operação de afinamento consiste em subtrair o resíduo da transformação acerto-ou-erro (Hit-or-Miss) da imagem original [4], sendo definida como:

$$A \otimes (J, K) = (A \ominus J) \cap (A^c \ominus K) \quad (\text{B.6})$$

$$\text{Thin}(J, K) = A - A \otimes (J, K) = A \cap (A \otimes (J, K))^c \quad (\text{B.7})$$

onde, A é a imagem original, J e K são os elementos estruturantes.

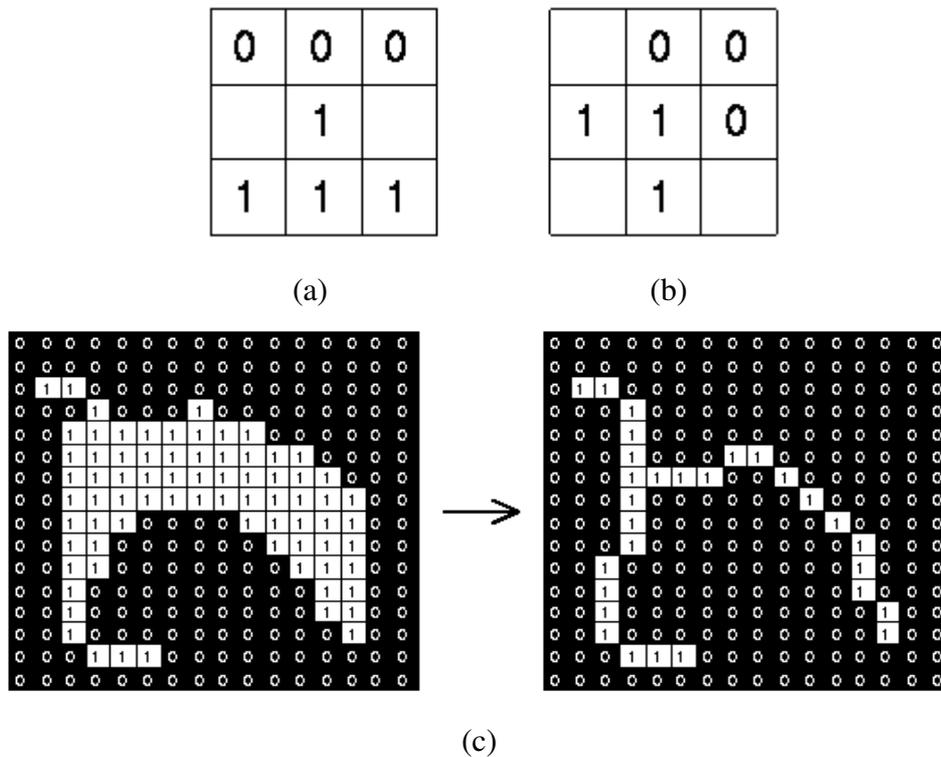


Figura B.10 - Esqueletização através de uma operação de afinamento: (a) elemento estruturante J , (b) elemento estruturante K , (c) esqueletização utilizando os operadores J e K [24].

O afinamento é calculado transladando o elemento estruturante na imagem. Se os *pixels* do elemento estruturante (0 e 1) casarem com os *pixels* da imagem, o *pixel* da imagem, que está sob a origem do elemento estruturante é marcado com zero, ou seja fundo. Caso contrário, o *pixel* na imagem não é alterado. A escolha o elemento estruturante determina as condições para que um determinado *pixel* seja eliminado, ou seja, marcada como sendo fundo. O afinamento é utilizado na operação de esqueletização. A Figura B.10(a) e a Figura B.10(b) mostram dois elementos estruturantes, *J* e *K*, respectivamente, utilizados na operação de esqueletização da imagem da Figura B.10(c).

B.5 – FILTROS

Nesta parte serão tratados os principais filtros existentes e que foram aplicados no algoritmo implementado. Esses filtros são transformações realizadas sobre a imagem de modo a suavizar, realçar, identificar bordas e outras operações que possibilitam o processamento digital de imagens.

Podemos classificar os filtros em três tipos: os de passa-baixa, que permitem a passagem dos sinais de baixa frequência e atenua os de alta; os de passa-alta que permitem a passagem dos sinais de alta frequência; e os de passa-faixa que definem limites inferior e superior de aceitação das frequências dos sinais.

As operações de abertura e fechamento são conhecidas como filtros morfológicos básicos, e já foram descritos na seção B.5.

B.5.1 – Filtro de Mediana

O filtro de mediana, assim como os demais filtros, é utilizado para reduzir o nível de ruído de uma imagem. Uma vantagem desse filtro é a capacidade de preservar os detalhes da imagem, ao contrário do filtro de média, mas o processamento necessário para efetuá-lo é relativamente pesado pois é preciso efetuar a ordenação dos elementos que compõem a máscara do filtro.

A aplicação se resume em comparar os valores de um conjunto de *pixels* de cada vez, que serão, previamente, ordenados de maneira crescente, escolhendo a mediana desse conjunto. A consequência dessa operação é a suavização da imagem como um todo, com a redução da diferença entre *pixels* vizinhos.

Na figura B.11 temos a aplicação do filtro de mediana com máscara 3x3 na imagem representada matricialmente pelos valores do nível de cinza dos seus *pixels*.

Essa máscara nos fornece o seguinte conjunto, em ordem crescente: {120, 120, 120, 120, 155, 155, 255, 255, 255}. Nele, é identificado a mediana, e este valor será o novo valor do centro da máscara, marcado em vermelho na figura acima. O resultado final do *pixel* depende apenas dos seus vizinhos definidos pela máscara. Com isso, quanto maior a máscara, maior o resultado da suavização.

120	155	155	100	95	81	81	0
120	120	120	120	100	81	0	0
255	255	255	255	255	255	255	0
255	255	255	0	0	0	0	0
110	110	90	80	70	70	80	90
121	0	0	255	255	90	90	90
145	220	110	190	255	90	90	255
167	188	110	190	120	188	0	0

Figura B.11 – Aplicação de máscara 3x3 sobre a imagem para filtro de mediana.

B.5.2 – Filtro de Média

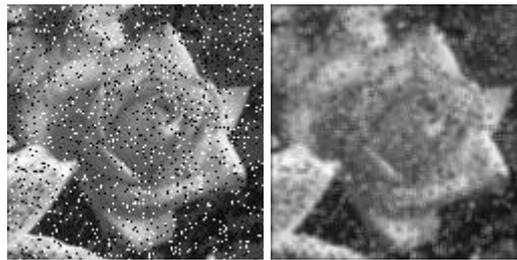
Diferentemente do filtro de mediana, visto na seção anterior, os detalhes da imagem são perdidos pois novos valores são atribuídos aos *pixels* através do cálculo da média dos valores envoltos pela máscara.

120	155	155	100	95	81	81	0
120	120	120	120	100	81	0	0
255	255	255	255	255	255	255	0
255	255	255	0	0	0	0	0
110	110	90	80	70	70	80	90
121	0	0	255	255	90	90	90
145	220	110	190	255	90	90	255
167	188	110	190	120	188	0	0

Figura B.12 – Aplicação de máscara 3x3 sobre a imagem para filtro de média.

Este tipo de filtro pode ser considerado como um passa-baixa pois nivela os níveis de cores dos pixels vizinhos com um valor médio que é sempre menor do que os maiores valores dos *pixels* vizinhos envoltos pela máscara utilizada.

Na figura B.12, a máscara traz os seguintes elementos: 255, 255, 255, 0, 0, 0, 80, 70, 70; que nos fornece o valor de média $(255+255+255+0+0+0+80+70+70)/9= 109,4 \sim 109$. Logo, o *pixel* central da máscara passa a ter o valor 109, e tal valor ocasionará uma perda relativa na imagem, realizando uma suavização mais agressiva.



(a) (b)

Figura B.13 – Filtro de média aplicado sobre imagem com ruído (a), com máscara 3x3, gerando a imagem suavizada (b).

B.5.3 – Filtro Gaussiano

O operador Gaussiano [4] é uma operação de convolução capaz de borrar uma imagem digital com o objetivo de remover detalhes e ruídos. Com esta finalidade, ele se torna muito semelhante ao filtro de média, utilizado da mesma forma, com a diferença da estrutura da máscara. No filtro gaussiano a máscara representa a forma da distribuição Gaussiana, com diversas propriedades especiais.

Em duas dimensões, o filtro gaussiano mais comum é o de distribuição circular simétrica (isotrópica), com média zero e desvio padrão σ . Ele é representado pela seguinte equação:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (\text{B.8})$$

A idéia da suavização por filtragem Gaussiana é a utilização da distribuição 2-D (Figura B.14) como uma função de ponto de espalhamento obtida através de uma convolução. Como a imagem é representada através de uma distribuição discreta de pixels, é necessário produzir uma aproximação discreta da função de distribuição Gaussiana antes de realizar a convolução. Teoricamente a distribuição Gaussiana é assintótica e positiva em

qualquer valor, e com isso, seria necessária uma máscara de convolução infinitamente grande, mas na prática ela se torna zero para qualquer valor mais distante do que três ou quatro desvios-padrões em relação à média, o que permite ignorarmos a máscara a partir deste ponto. Essa máscara de convolução aplicada será um conjunto de valores discretos que se aproxima de uma distribuição Gaussiana.

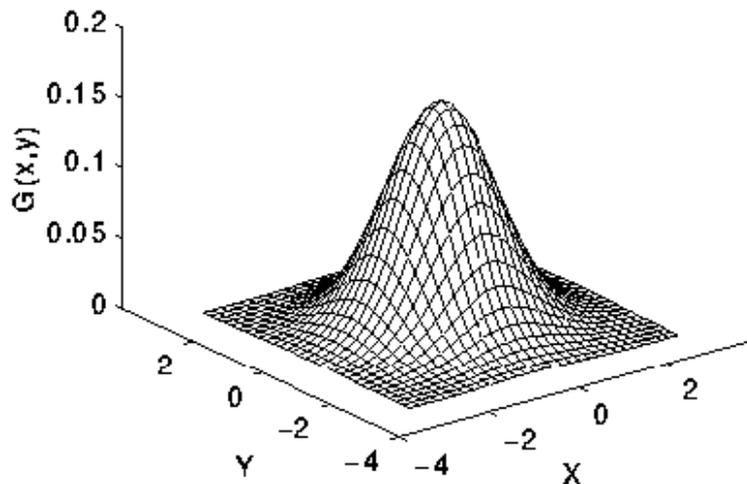


Figura B.14 – Forma da distribuição Gaussiana com média zero e desvio σ em 2-D

Uma vez que a máscara adequada tenha sido definida, a operação de suavização Gaussiana pode ser desenvolvida utilizando métodos de convolução. Assim, a convolução em 2-D pode ser executada primeiramente convoluindo na direção de X com a distribuição Gaussiana em 1-D e depois convoluindo na direção Y , também em 1-D.

O resultado obtido com a utilização dessa distribuição representa desta forma uma média ponderada de cada pixel em relação a sua vizinhança, resultando numa filtragem melhor do que o filtro de média de mesmo tamanho. Enquanto isso, o grau de suavização de uma imagem através da filtragem Gaussiana é determinado pelo seu desvio-padrão. Desse modo, quanto maior o desvio-padrão, maior a suavidade e há a exigência de máscaras de tamanhos maiores capazes de representá-los.

De modo alternativo, podemos calcular o suavizamento de uma imagem por um filtro gaussiano de desvio-padrão elevado, convoluindo-a diversas vezes com uma máscara menor. Essa abordagem impacta na arquitetura, que precisa ser bem especificada de modo a explorar o máximo do algoritmo, e do rendimento do *software*.

B.5.4 – Filtro de Lee

Este tipo de filtro é adequado para a eliminação de ruídos do tipo *speckle*, que é um ruído multiplicativo proporcional ao sinal, muito comum nas imagens de ecocardiografias e que conferem às mesmas um aspecto granuloso que prejudica as operações de segmentação.

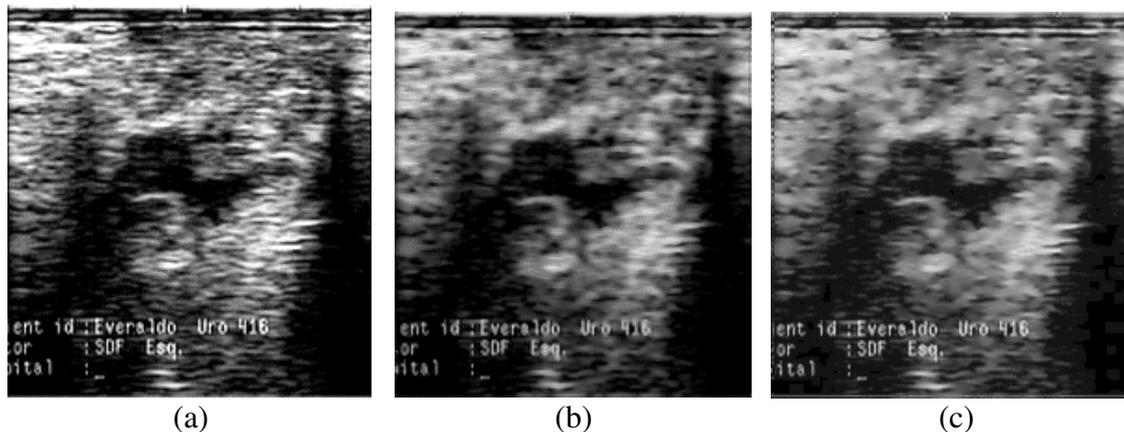


Figura B.15 – Imagem original (a); Resultado da aplicação do Filtro de Lee com máscara 3x3 (b); Resultado da aplicação do Filtro de Lee com máscara 5x5 (c);

O Filtro de Lee, um filtro de estatística local, adota um modelo multiplicativo para o ruído e obedece ao critério de "*local linear minimum mean square error*". Pelo fato de admitir que a média e a variância do sinal não são estacionários, este filtro utiliza as estatísticas locais de cada *pixel* filtrado, tendo este um valor de média e variância compatíveis com os dos demais *pixels* da janela.

Este filtro é linear pois aplica uma expansão, por Série de Taylor, da multiplicação entre o sinal e o ruído em torno da média, através dos termos lineares. O resultado desta linearização torna o ruído e o sinal independentes, pois o modelo deixa de ser multiplicativo e passa a ser aditivo.

Além disso, essa operação minimiza o erro médio quadrático (*mean square error*), e possui a vantagem de não necessitar de um modelo estatístico global do sinal, estimando esses valores através da média e das variâncias locais. A exemplificação dos resultados obtidos com este filtro pode ser vista na Figura B.15.

B.6 – SEGMENTAÇÃO

A aplicação mais direta da segmentação de imagens é a identificação de objetos ou partes específicas em um processamento digital de imagens. Isso é feito através da divisão

da imagem em partes ou regiões formadas por um conjunto de *pixels*, facilitando a análise da mesma.

Essas regiões identificadas, normalmente, compartilham de uma ou mais características comuns, e são delineadas por uma borda bem definida. As diferenças entre essas regiões, quer seja de cor, brilho, textura e outras, são essenciais no processo de sua identificação, através de um processo qualquer, como por exemplo, a limiarização. O nível até o qual essa subdivisão em regiões deve ser realizada depende do predicado da imagem e/ou do problema a ser resolvido, ou seja, a segmentação deve parar quando os objetos de interesse tiverem sido isolados[4].

A segmentação traduz o esforço realizado pela visão computacional quando essa adota um funcionamento análogo ao da visão humana, ou seja, efetuando agrupamentos sobre as imagens percebidas. Esses agrupamentos são baseados na proximidade, na similaridade e a continuidade, decompondo uma imagem em unidades significantes, o que leva a compreensão da imagem[23].

Existem, atualmente, diversos algoritmos de segmentação de imagens, mas todos apresentam pontos fortes e deficiências que são, constantemente, trabalhados e geram uma ampla área de estudos. A aplicação prática dessa operação sobre imagens digitais existe em diversas áreas do conhecimento humano, mas merece especial destaque no que diz respeito ao tratamento de imagens médicas como ferramenta de auxílio ao profissional de saúde na obtenção do diagnóstico final.

O dessa segmentação é reduzir o número de informações presentes na imagem de modo a concentrar o processamento sobre o que se deseja identificar, ignorando todo o restante.

B.6.1 – Limiarização

Este processo, também chamado de *Thresholding*, é um dos principais e mais básicos de segmentação de imagens. Consiste em agrupar os *pixels* de acordo com sua similaridade, no que diz respeito aos níveis de cinza, através da seleção de um limiar T que os separa. O princípio básico é separar a imagem em regiões, quando essa apresenta duas classes, o objeto e o fundo.

Para uma imagem $Im_{n \times m}$ temos que o resultado limiarizado torna-se:

$$Im(x,y) = 1 \text{ se } Im(x,y) \geq T \quad (\text{B-9})$$

$$0 \text{ se } Im(x,y) \leq T \quad (\text{B-10})$$

Onde $Im(x,y)$ é o valor do nível de cinza do *pixel* na posição (x,y) , sendo 1 para o nível correspondente ao objeto de frente e 0 correspondente ao fundo, e T é um nível de cinza qualquer adotado como limiar.

Os métodos de limiarização têm duas abordagens distintas, uma global e outra local. Os métodos de limiarização globais utilizam um único limiar T para toda imagem. Por outro lado, os métodos de limiarização local têm com princípio dividir a imagem em sub-regiões, onde, cada sub-região tem seu limiar específico. Embora a limiarização pareça ser uma operação bastante simples, a qualidade da imagem limiarizada depende da escolha adequada do valor de limiar. Para isso, diversos métodos de limiarização são encontrados na literatura.

Os histogramas da Figura B.16 mostram exemplos de como escolher o limiar mais adequado. No diagrama B.16(a), temos nitidamente um nível de cinza (eixo X) em que há nenhum pixel na imagem (eixo Y), podendo a partir dele, agrupar todos os demais em duas regiões bem nítidas. No fluxograma particionado por dois limiares, Figura B.16(b), identificamos três regiões nítidas de agrupamento de pixels.

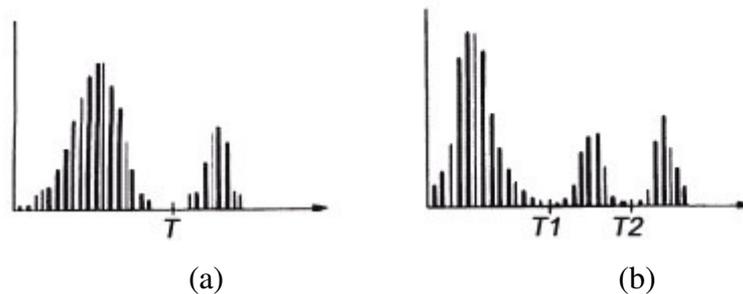


Figura B.16 – Histograma particionado por 1(a) e por 2(b) limiares.

Outra técnica também muito utilizada é o método de limiarização ótima, que parte de uma imagem da qual se conhecem as principais propriedades estatísticas. Assim, conhecendo-se a probabilidade de ocorrência dos *pixels*, bem como a média dos tons de cinza do objeto e do fundo e o desvio padrão dos mesmos, é possível chegar a um valor de limiar T ótimo com erro mínimo[4]. Matematicamente, o valor de limiar ótimo é dado pelas raízes das equações:

$$AT^2 + BT + C = 0 \tag{B.11}$$

$$A = \sigma_1^2 - \sigma_2^2 \tag{B.12}$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2) \quad (\text{B.13})$$

$$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right) \quad (\text{B.14})$$

onde:

- μ_1 e μ_2 são os valores médios dos dois níveis de cinza (objeto e fundo).
- σ_1 e σ_2 são os desvios padrão em torno das médias.
- P_1 e P_2 são as probabilidades de ocorrência dos dois níveis de cinza.

Se forem encontradas duas raízes reais e positivas, quer dizer que a imagem pode requerer dois valores de limiar para obter uma solução ótima.

Segundo Gonzalez[4], há também outras técnicas de limiarização, como seleção de limiar baseada nas características da fronteira e limiares baseados em diversas variáveis. De modo geral, as técnicas de limiarização são extremamente sensíveis ao ruído e não levam em consideração informações espaciais ou conhecimento prévio de especialistas médicos. Elas são mais rápidas e requerem menos recursos computacionais que outras técnicas de segmentação.

B.6.2 – Limiarização de Otsu

O método de binarização de Otsu, também chamado de *Thresholding Dinâmico*[7], é um método para cálculo automático de limiar, que consiste em separar os níveis de cinza em duas classes $C_0 = \{0, 1, \dots, t\}$ e $C_1 = \{t+1, \dots, n\}$, onde n é o número máximo de níveis de cinza da imagem. Uma das classes deve receber o nível de cinza de objeto e a outra o nível de cinza do fundo. Normalmente, adota-se o preto e o branco como estes níveis de cinza, nesta ordem.

Com isso, a variância entre as classes da imagem é maximizada, provocando a separação do objeto e do fundo da imagem. O resultado desta operação é mais significativo quando a variância de intensidade da imagem é maior.

Seja p_i a frequência de pontos na imagem com nível de cinza i . O limiar ótimo é calculado minimizando a variância (σ^2) entre as classes C_0 e C_1 , que é calculado da seguinte maneira:

$$\sigma^2 = w_0 w_1 (\mu_0 \mu_1) \quad (\text{B.15})$$

onde os termos são definidos como:

$$\mu_T = \sum_{i=0}^n ip_i \quad (\text{B.16})$$

$$w_0 = \sum_{i=0}^t p_i \quad (\text{B.17})$$

$$\mu_t = \sum_{i=0}^n ip_i \quad (\text{B.18})$$

$$w_1 = 1 - w_0 \quad (\text{B.19})$$

$$\mu_1 = \frac{\mu_T - \mu_0}{1 - w_0} \quad (\text{B.20})$$

$$\mu_0 = \frac{\mu_t}{w_0} \quad (\text{B.21})$$

Este método elimina as bordas claras, mesmo possuindo bom contraste com o restante da imagem, definindo um limiar (T) global. Uma desvantagem existente é sua sensibilidade à presença de ruídos na imagem, reduzindo consideravelmente seu desempenho, pois exige uma imagem com iluminação mais homogênea.

A binarização da imagem através do limiar de Otsu trata o histograma da imagem como uma Função Densidade de Probabilidade Discreta e nos fornece uma máscara que, se aplicada sobre a imagem original, possibilita separarmos o objeto do fundo.

B.6.3 – Gradiente Morfológico

O gradiente morfológico de uma imagem (Im) é definido a partir da diferença entre a dilatação e a erosão[4]. Podemos representá-la através da Equação B.22:

$$\rho(Im) = (Im \oplus B) - (Im \ominus B) \quad (\text{B.22})$$

onde ρ representa o gradiente por dilatação e erosão. Este tipo de operação salienta as bordas dos objetos e é utilizado para a identificação das mesmas. Na Figura B.17 pode-se observar a aplicação do gradiente morfológico para identificar as bordas dos objetos da imagem.

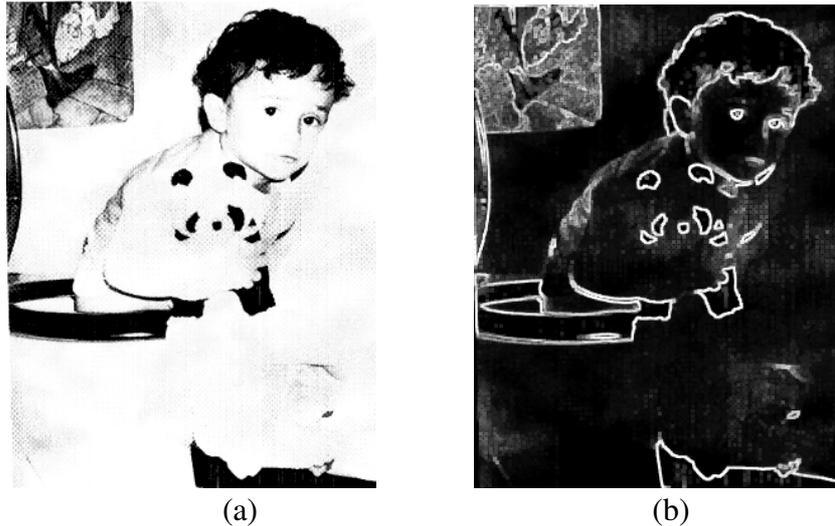


Figura B.17 - Imagem original (a) e após calcular o gradiente(b).

B.6.4 – Watershed

Em Geografia, *watersheds*, ou linhas de partição de águas, são saliências no terreno que dividem os diversos rios de uma bacia hidrográfica. Este conceito é o que norteia a aplicação e o funcionamento deste tipo de técnica de segmentação de imagens. O princípio é trabalhar com a imagem como se fosse um pedaço de terreno (Figura B.18), onde os níveis de altitude são representados pelos níveis (intensidade) de cinza dos *pixels*, inundando-os e identificando áreas mais protuberantes que dividem as diversas bacias formadas, caracterizando assim os objetos segmentados. A segmentação por *watershed* é um método de segmentação por crescimento de regiões.

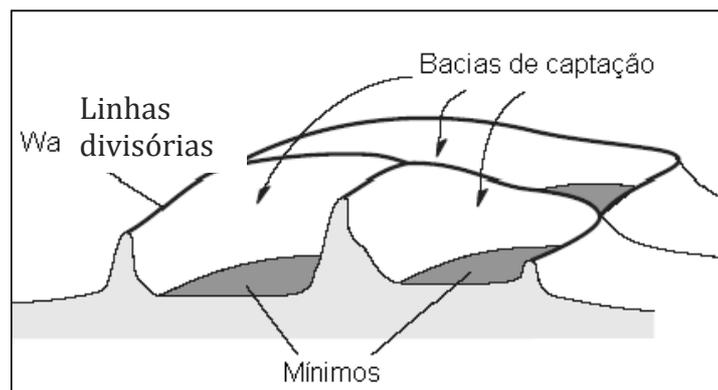


Figura B.18 - Linhas divisórias, mínimos e bacias de captação de um relevo topográfico.

Após identificar os *pixels* de menor nível de cinza (mínimos locais), o algoritmo realiza uma “inundação” a partir destes pontos, formando regiões ou bacias. Antes de uma

bacia se unir a outra, são traçadas linhas divisórias entre elas, chamadas de linhas de partição de águas ou linhas de divisão de *watersheds*. O objetivo principal desta transformada é identificar estas linhas divisoras que, provavelmente, serão as demarcadoras dos objetos a serem segmentados na imagem. A Figura B.19 mostra graficamente os diversos passos da inundação realizada com a definição das linhas divisoras de regiões.

Este método possui limitações. Se a imagem contiver ruídos e interferências nos níveis de cinza, ocorrerá uma supersegmentação, identificando muito mais regiões do que realmente existem. Isso devido às grandes variações de nível de cinza nos *pixels* que representam ruído na imagem.

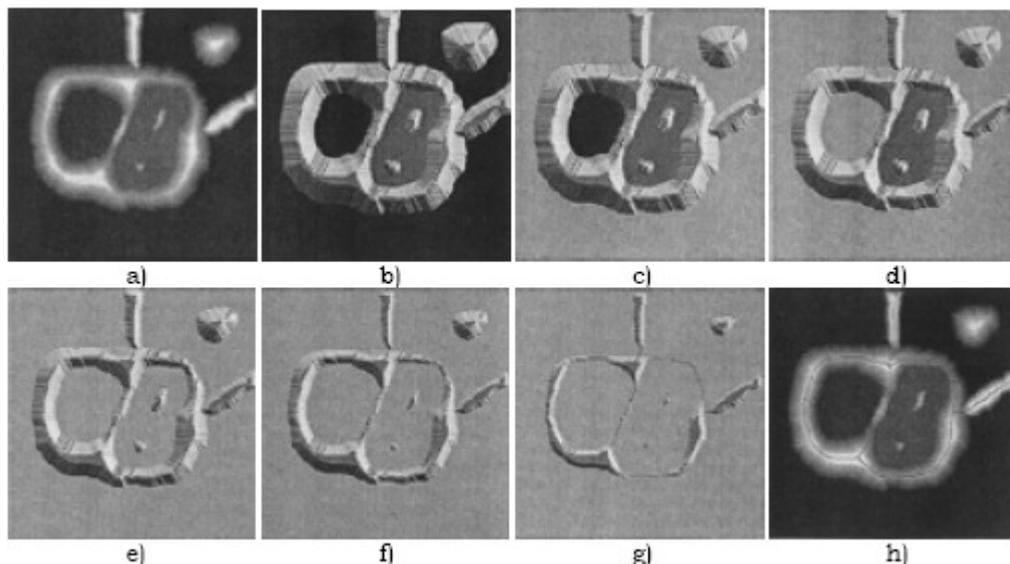


Figura B.19 - Segmentação por *Watershed*: a) imagem original, b) vista topográfica, c) e d) duas etapas do enchimento, e) resultado de enchimento adicional, f) início do preenchimento completo de dois vales (foi construído entre estes vales, uma pequena barragem), g) barragens maiores, h) *Watershed* final[4].

Para evitar este defeito, aplica-se um pré-processamento na imagem, suavizando e/ou eliminando os ruídos através dos filtros vistos anteriormente ou de alguma técnica que elimine ou reduza o nível de ruído da imagem. Somente após esta preparação, o *watershed* é executado para segmentar regiões.

Esse efeito de supersegmentação pode ocorrer também com a aplicação da *watershed* sobre o gradiente morfológico, mas isso só ocorre se houver ruídos excessivos na imagem original antes de calcular o gradiente, e estes devem ser suavizados através de um filtro adequado. Na Figura B.19 podemos observar o resultado de uma supersegmentação.



(a)



(b)

Figura B.20 – Imagem original (a) e supersegmentação com watershed (b).

Para que a formação das regiões na imagem ocorra de maneira adequada, é necessário identificar os pontos pelos quais as inundações iniciarão. Estes pontos geralmente são os mínimos locais, ou seja, os *pixels* com os menores valores de nível de cinza. A suavização da imagem reduz o número de mínimos locais, ocasionando a diminuição da quantidade de regiões identificadas, aproximando-se mais à realidade da imagem original.

A vantagem em utilizar a transformada watershed é que as linhas de separação de águas obtidas ao final do processo já representam as bordas mais significativas entre as regiões. Um aspecto interessante na utilização desta técnica é que sua aplicação, na maioria das vezes, não é feita diretamente sobre a imagem original, mas sim no gradiente morfológico da imagem, pois esse operador realça as descontinuidades da imagem. Disso decorre a necessidade de tratar a imagem antes de obter o gradiente morfológico de modo a eliminar os ruídos existentes e evitar a supersegmentação a que pode ser levada.