

UNIVERSIDADE FEEVALE

ANDERSON RODRIGO SPANIOL

PROCESSAMENTO DE IMAGENS DE TOMOGRAFIA  
COMPUTADORIZADA PARA SEGMENTAÇÃO AUTOMÁTICA  
DOS PULMÕES

Novo Hamburgo  
2020

ANDERSON RODRIGO SPANIOL

PROCESSAMENTO DE IMAGENS DE TOMOGRAFIA  
COMPUTADORIZADA PARA SEGMENTAÇÃO AUTOMÁTICA  
DOS PULMÕES

Trabalho de Conclusão de Curso  
apresentado como requisito parcial  
à obtenção do grau de Bacharel em  
Ciência da Computação pela  
Universidade Feevale

Orientador: Marta Rosecler Bez

Novo Hamburgo  
2020

## **AGRADECIMENTOS**

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

Aos meus pais, José e Eliane, pelos ensinamentos e suporte incondicionais.

À Carolina, minha noiva, que me apoiou e incentivou durante todo o desenvolvimento do trabalho.

À professora Marta, que sempre esteve à disposição para orientar e guiar a realização deste trabalho.

À Universidade Feevale, que dispõe de toda a estrutura para formar os alunos com excelência.

Muito obrigado!

## RESUMO

O objetivo deste trabalho é desenvolver um protótipo de software capaz de segmentar completamente e de forma automática os pulmões em exames de Tomografia Computadorizada (TC) do tórax. A metodologia utilizada é caracterizada como pesquisa aplicada, com abordagem quantitativa, de caráter exploratório e com procedimentos técnicos de pesquisa bibliográfica e experimentação. Os resultados foram validados comparando as saídas geradas pelo protótipo com os resultados de um trabalho correlato. Para o desenvolvimento, foi empregada a linguagem de programação Java e utilizada a biblioteca *PixelMed Java DICOM Toolkit* para manipulação dos arquivos no formato DICOM. O desenvolvimento foi realizado em 3 fases: a primeira realizava a processamento das fatias dos exames de forma independente, a segunda agregou a utilização de fatias de referência e a terceira acrescentou varreduras adicionais nas imagens. Os resultados alcançados apresentaram acurácia de 97% para a base de dados utilizada, frente aos 94% atingidos pelo trabalho correlato. O uso desse protótipo pode tornar a tarefa de diagnóstico mais precisa e viabiliza a modelagem 3D dos órgãos, uma vez que apresenta a segmentação de todas as camadas dos exames de TC.

Palavras-chave: Segmentação dos pulmões. Tomografia Computadorizada. Processamento digital de imagens. Informática médica.

## **ABSTRACT**

The goal of this paper is to develop a software prototype capable of completely and automatically segmenting the lungs in Computed Tomography (CT) exams of the chest. The methodology used is characterized as applied research, with a quantitative approach, of an exploratory nature and with technical procedures for bibliographic research and experimentation. The results were validated by comparing the outputs generated by the prototype with the results of a related work. For the development, the Java programming language was used and the PixelMed Java DICOM Toolkit library was used to manipulate files in DICOM format. The development was carried out in 3 phases: the first performed the processing of the slices of the exams independently, the second added the use of reference slices and the third added additional scans on the images. The results achieved showed an accuracy of 97% for the database used, compared to 94% achieved by the related work. The use of this prototype can make the diagnostic task more accurate and enables 3D modeling of the organs, since it presents the segmentation of all layers of CT exams.

**Keywords:** Lungs segmentation. Computed Tomography. Digital image processing. Medical informatics.

## LISTA DE FIGURAS

Figura 1 – Aplicação do brilho em uma imagem	15
Figura 2 – Aplicação do contraste em uma imagem	15
Figura 3 – Aplicação da binarização em uma imagem	17
Figura 4 – Aplicação de filtros passa-baixas em uma imagem	20
Figura 5 – Aplicação de filtros passa-altas em uma imagem	23
Figura 6 – Exemplos de elementos estruturantes	24
Figura 7 – Aplicações de operações de morfologia matemática em uma imagem	26
Figura 8 – Exemplos de vizinhanças de <i>pixels</i>	29
Figura 9 – Resultados obtidos por Rodrigues <i>et al.</i> (2016)	32
Figura 10 – Resultados apresentados por Felix <i>et al.</i> (2007)	34
Figura 11 – Resultados do AUTOIN	35
Figura 12 – Funcionamento de um scanner de TC	40
Figura 13 – Exemplos de uso de WL e WW	43
Figura 14 – Estrutura de um arquivo DICOM	43
Figura 15 – Estrutura de um <i>dataset</i> e de um <i>data element</i>	44
Figura 16 – Resultado do processamento da fatia inicial	51
Figura 17 – Resultado do processamento completo de uma fatia	52
Figura 18 – Resultado do processamento adicional sobre a fatia	53
Figura 19 – Código-fonte da orquestração do processamento	53
Figura 20 – Fluxograma da segmentação de uma fatia	54
Figura 21 – Imagem original e imagem sem valores inválidos de coeficiente de atenuação	55
Figura 22 – Aplicação do Filtro Gaussiano	56
Figura 23 – Identificação do contorno do tórax	57
Figura 24 – Limiarização com definição dinâmica do limiar	58
Figura 25 – Remoção da mesa do tomógrafo	59
Figura 26 – Resultado do processo de rotulação	60
Figura 27 – Identificação dos pulmões por dimensionamento	61
Figura 28 – Código-fonte que identifica a sobreposição de 2 objetos	62
Figura 29 – Pulmão esquerdo e pulmão direito conectados	63
Figura 30 – Pulmões conectados na parte superior, após a separação	64
Figura 31 – Pulmões conectados na parte inferior, após a separação	64
Figura 32 – Código-fonte que separa os pulmões conectados	66

Figura 33 – Código-fonte que busca a menor distância para a borda inferior	6
Figura 34 – Resultado da separação dos pulmões	67
	68

## LISTA DE TABELAS

Tabela 1 – Fragmento dos resultados obtidos pelo <i>Follower of Lung Contour</i>	33
Tabela 2 – Resultados apresentados por Rönnau (2015)	36
Tabela 3 – Comparativo entre os trabalhos de Rodrigues <i>et al.</i> (2016) e Rönnau (2015)	37
Tabela 4 – Valores de exemplo em HU	41
Tabela 5 – Detalhamento dos exames da base de dados	46
Tabela 6 – Atributos do padrão DICOM utilizados no protótipo	49
Tabela 7 – Exames utilizados na validação do protótipo	69
Tabela 8 – Resultados parciais de cada fase	70
Tabela 9 – Resultados comparados com Rönnau (2015)	70
Tabela 10 – Causas dos erros	71



## LISTA DE ABREVIATURAS E SIGLAS

ACC	<i>Accuracy</i>
ACR	<i>American College of Radiology</i>
DICOM	<i>Digital Imaging and Communications in Medicine</i>
DPOC	Doença de Obstrução Pulmonar Crônica
FLC	<i>Follower of Lung Contour</i>
HM	<i>Harmonic Mean</i>
HU	<i>Hounsfield Unit</i>
LOC	Localização do Centro do Objeto
MCA	Método de Contornos Ativos
NEMA	<i>National Electrical Manufacturers Association</i>
PDA	<i>Pulmonary Densities Analysis</i>
PDI	Processamento Digital de Imagens
RNA	Rede Neural Artificial
TC	Tomografia Computadorizada
VR	<i>Value Representation</i>
WL	<i>Window Level</i>
WW	<i>Window Width</i>

## LISTA DE SÍMBOLOS

$\beta$	Letra grega <i>Beta</i>
$\alpha$	Letra grega <i>Alpha</i>
$\ominus$	Erosão
$\oplus$	Dilatação

## SUMÁRIO

<b>1. INTRODUÇÃO</b>	<b>12</b>
<b>2. REVISÃO BIBLIOGRÁFICA</b>	<b>14</b>
2.1. PRÉ-PROCESSAMENTO DAS IMAGENS	14
2.1.1. Brilho e Contraste	14
2.1.2. Limiarização ( <i>threshold</i> )	16
2.1.3. Binarização	16
2.1.4. Remoção de ruídos	17
2.1.4.1. Filtro de Média	18
2.1.4.2. Filtro de Moda	18
2.1.4.3. Filtro de Mediana	18
2.1.4.4. Filtro Gaussiano	19
2.1.5. Detecção de Bordas	21
2.1.5.1. Sobel	21
2.1.5.2. Frei-Chen	22
2.1.5.3. Marr and Hildreth	23
2.1.6. Morfologia Matemática	23
2.1.6.1. Erosão	24
2.1.6.2. Dilatação	25
2.1.6.3. Abertura	25
2.1.6.4. Fechamento	26
2.2. TÉCNICAS DE SEGMENTAÇÃO	27
2.2.1. Crescimento de regiões	27
2.3. TÉCNICAS DE CLASSIFICAÇÃO	28
2.3.1. Rotulação	28
2.4. CONSIDERAÇÕES	29
<b>3. TRABALHOS CORRELATOS</b>	<b>31</b>
3.1. <i>LUNG SEGMENTATION IN CHEST COMPUTERIZED TOMOGRAPHY IMAGES USING THE BORDER FOLLOWING ALGORITHM</i>	31
3.2. <i>AUTOMATIC SEGMENTATION AND MEASUREMENT OF THE LUNGS IN HEALTHY PERSONS AND IN PATIENTS WITH CHRONIC OBSTRUCTIVE PULMONARY DISEASE IN CT IMAGES</i>	33
3.3. <i>AUTOIN: METHOD OF AUTOMATIC INITIALIZATION OF ACTIVE CONTOURS APPLIED TO LUNGS IN CT IMAGES</i>	34
3.4. SEGMENTAÇÃO AUTOMÁTICA DE ÓRGÃOS EM IMAGENS DE TOMOGRAFIA COMPUTADORIZADA DO TÓRAX	36
3.5. COMPARATIVO DAS TÉCNICAS	36
3.6. CONSIDERAÇÕES	37
<b>4. EXAMES MÉDICOS POR IMAGEM</b>	<b>39</b>

	11
4.1. TOMOGRAFIA COMPUTADORIZADA	39
4.2. PADRÃO DICOM	40
4.2.1. Escala <i>Hounsfield</i>	41
4.2.2. Nível de janela e Largura de janela	42
4.2.3. Formato de arquivos	43
4.3. BASES DE EXAMES MÉDICOS POR IMAGEM	45
4.4. CONSIDERAÇÕES	46
<b>5. DESENVOLVIMENTO E VALIDAÇÃO DO PROTÓTIPO</b>	<b>48</b>
5.1. PREPARAÇÃO DO AMBIENTE	48
5.2. IMPLEMENTAÇÃO	49
5.2.1. Manipulação de arquivos DICOM	49
5.2.2. Orquestração do processamento	50
5.2.3. Segmentação dos pulmões	54
5.2.3.1. Conversão de valores	55
5.2.3.2. Aplicação do Filtro Gaussiano	55
5.2.3.3. Identificação do contorno do tórax	56
5.2.3.4. Definição e aplicação do limiar	57
5.2.3.5. Remoção do tomógrafo	58
5.2.3.6. Rotulação	59
5.2.3.7. Identificação dos pulmões por dimensionamento	60
5.2.3.8. Identificação dos pulmões por sobreposição	61
5.2.3.9. Verificação de conectividade dos pulmões	62
5.3. VALIDAÇÃO DOS RESULTADOS	68
5.4. CONSIDERAÇÕES	72
<b>6. CONCLUSÃO</b>	<b>73</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>75</b>

## 1. INTRODUÇÃO

Sabe-se que, quanto mais cedo o diagnóstico de um problema de saúde é realizado, maior a chance de cura e menores os possíveis impactos causados. Para tal, a medicina utiliza diversos artifícios visando aumentar a velocidade e a precisão dessas tarefas. Uma das ferramentas mais utilizadas são os exames médicos, que trazem diversas informações para o especialista para auxiliá-lo no processo de decisão.

A Tomografia Computadorizada é um tipo de exame que consegue trazer uma representação visual dos tecidos internos do paciente para o médico. Exames de TC são largamente utilizados na região do tórax, onde é possível visualizar vários órgãos, como, por exemplo, coração, pulmões e fígado. O processo de análise dessas imagens exige precisão e está sujeito a falhas humanas, uma vez que é o especialista quem analisa visualmente os exames.

Equipamentos de TC normalmente exportam os dados dos exames em um formato de arquivos padronizado chamado DICOM. Como os dados são armazenados digitalmente, é possível desenvolver sistemas computacionais que auxiliam os profissionais na tarefa de análise e diagnóstico.

A área da computação especializada no tratamento de imagens digitais é denominada Processamento Digital de Imagens (PDI). Com ela, é possível realizar transformações nas imagens e operações para identificar regiões de interesse. A segmentação automática de imagens é uma subárea do PDI, e especializa-se em encontrar automaticamente formas e objetos dentro de uma imagem.

Diversos trabalhos têm sido realizados buscando a segmentação automática de órgãos em exames de TC. Em geral, eles visam automatizar a tarefa de identificação dos órgãos, que, muitas vezes, é difícil a olho nu. Para tal, diversos autores utilizam diferentes técnicas e abordagens buscando atingir seus resultados. Rodrigues *et al.* (2016) desenvolveram um algoritmo de segmentação de pulmões em Tomografias Computadorizadas chamado *Follower of Lung Contour* (FLC). Felix *et al.* (2007) utilizam a técnica de crescimento de regiões para identificar os pulmões. Felix *et al.* (2012) também apresentam, em outro trabalho, o AUTOIN, que é uma abordagem para gerar automaticamente o *pixel* inicial (*seed*) a partir de diversas transformações e operações sobre as imagens de TC. Rönnau (2015) propõe uma técnica baseada no pré-processamento das imagens e rotulação das regiões de interesse para identificar os pulmões em TC. Esses trabalhos são detalhados no Capítulo 3.

O objetivo deste trabalho é desenvolver um protótipo de software que segmente automaticamente todas as fatias dos pulmões em imagens de Tomografia Computadorizada do tórax. Para tal, será dada uma atenção especial às fatias próximas às extremidades superior e inferior dos órgãos, que são mais suscetíveis a erros pelo fato de não apresentarem uma forma bem definida. Para alcançar esse objetivo, será composta uma base de dados com exames de TC no padrão DICOM, a qual será utilizada para testes e validações. Será feito um estudo sobre o padrão de arquivos DICOM e uma exploração das técnicas de PDI que servirão de subsídio para o desenvolvimento do software. Então será desenvolvido um protótipo de software que atenda às definições e os resultados serão validados.

Em relação à natureza, esse trabalho é caracterizado como uma pesquisa aplicada, pois envolve o desenvolvimento de um protótipo que visa resolver um problema específico. A forma de abordagem é quantitativa, pois os resultados são validados comparando as saídas geradas pelo protótipo com os resultados de um trabalho correlato. O objetivo do estudo é exploratório, uma vez que apresenta uma proposta para melhorar o processo de análise de exames de Tomografia Computadorizada e segmentação das regiões de interesse, reduzindo a ocorrência de falhas. Os procedimentos técnicos empregados no desenvolvimento do estudo são pesquisa bibliográfica e experimentação. A pesquisa bibliográfica foi utilizada no trabalho através do uso de livros e artigos científicos relacionados ao tema. A pesquisa experimental foi empregada no desenvolvimento de um protótipo de software e de testes práticos, não havendo validação exclusivamente teórica.

Este trabalho está dividido em seis capítulos. O Capítulo 1 é a introdução do trabalho, onde é feita uma contextualização do tema e destacada sua importância. No Capítulo 2 é apresentada uma revisão bibliográfica acerca de técnicas de Processamento Digital de Imagens. No Capítulo 3 são apresentadas algumas pesquisas que já foram feitas na área de segmentação de órgãos a partir de exames de TC. O Capítulo 4 aprofunda o estudo acerca de exames médicos por imagem, dando um foco especial ao formato de arquivos DICOM e detalhando a base de dados utilizada. O Capítulo 5 explica a construção do protótipo e demonstra os resultados obtidos na sua validação. O Capítulo 6 fecha o trabalho trazendo as considerações finais.

## 2. REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentadas as técnicas utilizadas como subsídio para a construção do protótipo de software. Inicialmente, são apresentadas técnicas de pré-processamento de imagens, que incluem as operações elementares realizadas sobre imagens digitais. Em seguida, são apresentadas operações de segmentação e de classificação de regiões de interesse.

### 2.1. PRÉ-PROCESSAMENTO DAS IMAGENS

O pré-processamento é a etapa em que é feito o tratamento da imagem para acentuar suas características, realçando áreas de interesse (NOGUEROL, 2008). No item 2.1.1 são explicados os conceitos de brilho e contraste. O item 2.1.2 trata do processo de limiarização, ou *threshold*. Em seguida, é apresentada uma de suas variações, que é a binarização, no item 2.1.3. O item 2.1.4 explica diferentes técnicas de remoção de ruídos. No item 2.1.5 são abordados métodos de detecção de bordas em imagens. Por fim, são exploradas técnicas envolvendo morfologia matemática no item 2.1.6.

#### 2.1.1. Brilho e Contraste

Em processamento digital de imagens, o conceito de brilho está relacionado com a intensidade luminosa (GONZALEZ; WOODS, 2010). O brilho é a intensidade luminosa dos *pixels* da imagem (PEDRINI; SCHWARTZ, 2008).

O contraste é definido como a variação dos valores dos *pixels*, permitindo que sejam diferenciados elementos a partir da cor ou intensidade dos níveis de cinza (RÖNNAU, 2015). A visualização da imagem pode ser melhorada com a variação do contraste (NOGUEROL, 2008).

O brilho e o contraste podem ser definidos, conforme Becker (2018), através da função matemática  $g = \alpha.f + \beta$ . Onde o  $g$  é o *pixel* resultante,  $\alpha$  é o coeficiente de contraste,  $\beta$  é o coeficiente de brilho e  $f$  é o *pixel* analisado.

Pela fórmula matemática é possível concluir que se o contraste  $\alpha$  for um, varia-se somente o brilho  $\beta$  do *pixel*  $f$ . Por outro lado, se o brilho  $\beta$  tiver valor zero, varia-se somente o contraste  $\alpha$ .

**Figura 1 - Aplicação do brilho em uma imagem**



Fonte: elaborado pelo autor

A Figura 1 apresenta uma imagem cujo brilho foi alterado. À esquerda é exibida a imagem original, ao centro a imagem com o brilho aumentado e à direita a imagem com o brilho diminuído.

**Figura 2 - Aplicação do contraste em uma imagem**



Fonte: elaborado pelo autor

A Figura 2 apresenta uma imagem cujo contraste foi alterado. À esquerda é exibida a imagem original, ao centro a imagem com o contraste aumentado e à direita a imagem com o contraste diminuído.



### 2.1.2. Limiarização (*threshold*)

Segundo Mossmann (2007), a limiarização é uma das mais importantes abordagens na segmentação de imagens. Ela consiste em separar objetos do fundo através da seleção de um limiar.

Essa técnica define limiares para os valores dos *pixels* e os converte para valores padronizados, normalmente zero (preto) ou um (branco). De acordo com Flores (2015), o objetivo é extrair das imagens regiões que atendem a determinados critérios de intensidade de *pixels*. Dado um limiar  $T$ , a imagem resultante será composta de todos os *pixels* abaixo do limiar para zero e todos os *pixels* acima do limiar convertidos para um. Abaixo é apresentada a representação matemática desta transformação:

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

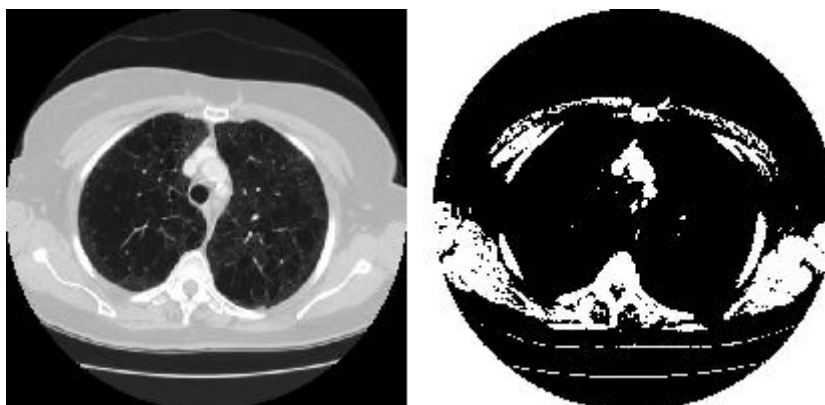
Conforme Flores (2015), o processo de limiarização pode ser tão detalhado e customizado quanto exige o seu contexto de aplicação. É comum que sejam utilizados 2 limiares, um para indicar um limite inferior e outro para indicar um limite superior. Da mesma forma, é possível que os *pixels* que atendam essa regra sejam convertidos para outros valores de marcação, diferentes de zero e um. Portanto, a equação acima pode ser modificada para atender ao cenário ao qual está sendo aplicado, podendo variar o número de faixas, os limiares superior e inferior de cada faixa e o valor de marcação de cada uma dessas faixas.

Flores (2015) explica que a limiarização pode ocorrer de duas maneiras: globalmente ou localmente. Quando o limiar é constante ao longo de toda a imagem, chama-se o limiar de global, pois resulta em uma transformação uniforme dela. Por outro lado, a limiarização local denota a limiarização variável na qual o limiar  $T$  em um determinado ponto  $(x,y)$  depende das propriedades de sua vizinhança  $(x,y)$ .

### 2.1.3. Binarização

Acima foi apresentada a definição do processo de limiarização. A binarização é uma variação desse processo. Conforme Pohren (2018), quando a limiarização possui apenas um limiar ela é denominada binarização, pois a imagem resultante possui apenas dois valores de intensidade, 0 (preto) ou 1 (branco).

Figura 3- Aplicação da binarização em uma imagem



Fonte: elaborado pelo autor

A Figura 3 apresenta uma imagem binarizada. O limiar utilizado é 40, portanto, *pixels* abaixo desse valor na escala de cinzas são convertidos para preto e acima para branco. À esquerda é exibida a imagem original e à direita a imagem binarizada.

#### 2.1.4. Remoção de ruídos

A presença de ruídos é frequentemente identificada em imagens digitais. Sua origem normalmente se dá no processo de aquisição e/ou transmissão da imagem. Alguns dos fatores que interferem na coleta da imagem são as condições ambientais, a qualidade e as propriedades dos sensores utilizados no processo (GONZALEZ; WOODS, 2010).

O tratamento empregado na remoção de ruídos de imagens é denominado de filtro passa-baixas. O resultado da aplicação de um filtro passa-baixas em uma imagem digital é a suavização e redução das variações nos níveis de cinza dela. O objetivo desse procedimento é reduzir ruídos, mas a tendência é que faça a imagem perder nitidez (CONCI; AZEVEDO; LETA, 2008). A aplicação do filtro é realizada através do processo de convolução (GONZALEZ; WOODS, 2010) entre uma matriz de coeficientes específicos e a imagem (PEDRINI; SCHWARTZ, 2008). O processo de convolução consiste em sobrepor uma matriz, denominada *kernel* sobre uma imagem, multiplicando *pixel* a *pixel* os valores e somando. Ao final, divide-se o valor resultante pela soma dos valores do *kernel*, alterando o *pixel* central na imagem de destino. Abaixo são apresentados 4 filtros passa-baixas utilizados na remoção de ruídos.

#### **2.1.4.1. Filtro de Média**

O filtro de média tem como proposta suavizar as imagens através do borramento e da redução de ruídos. De acordo com Becker (2018), a ideia por trás desta técnica é aplicar uma operação estatística na vizinhança de uma máscara, calculando um novo valor do *pixel* analisado.

O algoritmo consiste em calcular a média aritmética da máscara em processamento e aplicar o valor resultante no *pixel* central. Devido à diminuição de variações na imagem que este filtro produz, os ruídos são eliminados. (GONZALEZ; WOODS, 2010).

#### **2.1.4.2. Filtro de Moda**

O algoritmo de filtro de moda se assemelha ao filtro de média, diferenciando-se pela forma como define o novo valor para o *pixel* central. Conforme Gonzalez e Woods (2010), ele calcula o valor de maior frequência entre os pontos da área em processamento.

A técnica consiste em calcular a frequência de cada um dos valores existentes na máscara. O valor de maior frequência, ou seja, que aparece o maior número de vezes, é usado para definir o *pixel* central (CONCI; AZEVEDO; LETA, 2008).

#### **2.1.4.3. Filtro de Mediana**

A implementação do algoritmo desse tipo de filtro é muito semelhante à implementação dos filtros de média e moda, com a diferença de que é calculada a mediana dos elementos da máscara em processamento. Ele é classificado como um filtro estatístico e provoca um menor borramento do que filtros lineares de suavização, como o filtro de média (BECKER, 2018).

Para determinar o valor da mediana  $m$ , é necessário ordenar o conjunto contendo um número  $n$  de elementos e encontrar o valor central destes elementos. Assim, metade dos elementos situam-se abaixo de  $m$  e a outra metade acima. Se  $n$  for par, determina-se a mediana através da média aritmética dos elementos centrais (CONCI; AZEVEDO; LETA, 2008).

De acordo com Conci *et al.* (2008), o filtro de mediana apresenta resultados melhores que o filtro de média. A justificativa é que, se existe um ruído nos elementos que compõem a máscara, este valor estará presente nas primeiras ou últimas posições. Portanto, pontos destoantes têm grande chance de serem tratados como ruídos e suavizados.

#### 2.1.4.4. Filtro Gaussiano

Conforme Pedrini e Schwartz (2008), os Filtros Gaussianos possuem características úteis para o processamento de imagens. Devido às suas funções serem simétricas, o grau de suavização é aplicado da mesma forma em todas direções (isotrópico). A imagem é suavizada através da substituição de cada *pixel* da imagem pela média ponderada dos vizinhos.

É um dos filtros passa-baixas mais importantes, pois o nível de suavização ocorre de maneira uniforme, o que não ocorre em outros filtros, como, por exemplo, nos filtros de média, moda e mediana. Filtros Gaussianos são adequados para aplicação em conjunto com algoritmos de detecção de bordas (CONCI; AZEVEDO; LETA, 2008).

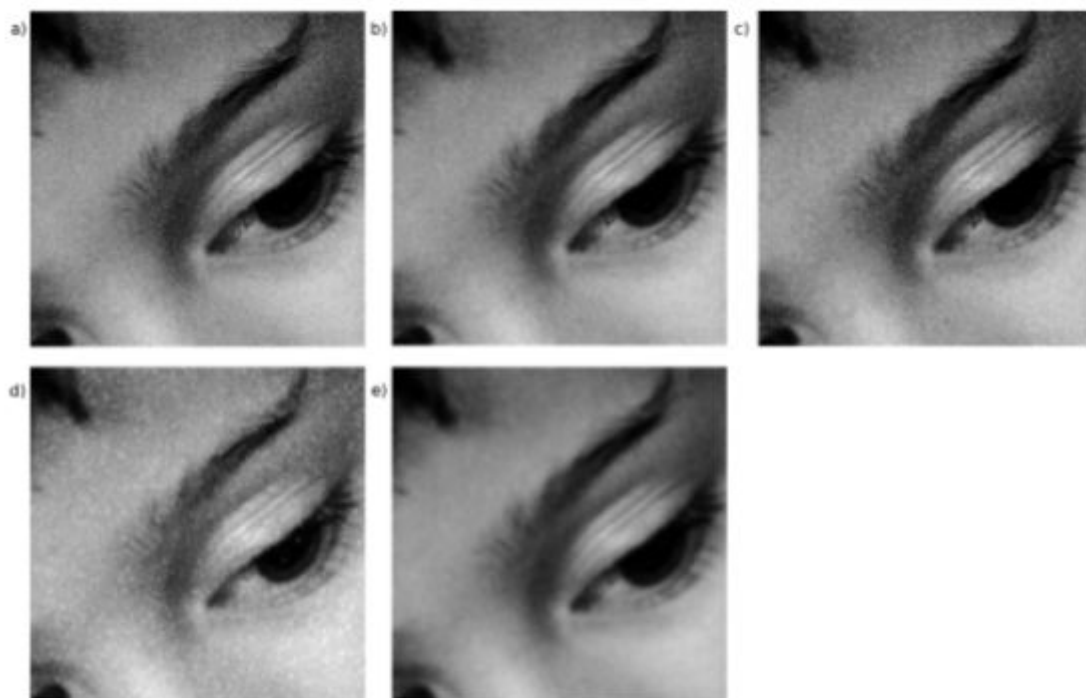
Os coeficientes da máscara de processamento são obtidos através de uma função Gaussiana bidimensional. Becker (2018) apresenta a função Gaussiana com média zero e desvio padrão  $\sigma$  conforme descrito na equação:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right)$$

O Filtro Gaussiano pode ser aplicado através de convoluções por meio de matrizes. Abaixo são apresentadas duas máscaras que podem ser utilizadas na aplicação do filtro.

$$Z = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \quad Z = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Figura 4 - Aplicação de filtros passa-baixas em uma imagem



Fonte: Becker (2018)

A Figura 4 apresenta a suavização da imagem de origem através da utilização de filtros passa-baixas. A imagem (a) representa a imagem original. O primeiro filtro apresentado é o filtro da média (b), seguido do filtro da moda (c), da mediana (d) e do Filtro Gaussiano (e).

### 2.1.5. Detecção de Bordas

O principal objetivo deste tipo de abordagem é identificar descontinuidades significativas em imagens em tons de cinza. Os algoritmos realizam transformações nas imagens que realçam suas bordas. (MOSSMANN, 2007).

Conhecidos como filtros passa-altas, estes algoritmos atenuam ou eliminam os componentes de baixa-frequência. As baixas frequências são responsáveis pelas características que variam lentamente na imagem, como o contraste total e a intensidade média. Portanto, o efeito resultante da aplicação desse tipo de filtro é a redução destas características, gerando uma aparente agudização de bordas, trocas repentinas de intensidade e ruído (POHREN, 2018).

Gonzalez e Woods (2010) definem borda como o limite entre duas regiões com propriedades distintas de níveis de cinza. As técnicas de detecção de bordas realizam operações locais utilizando um operador local diferencial, normalmente o gradiente. Uma abordagem muito comum é a realização de operações a partir de máscaras, sendo estas convolucionadas com a imagem original (MOSSMANN, 2007).

#### 2.1.5.1. Sobel

O operador de Sobel foi proposto por Sobel e Feldman (1968). Ele utiliza duas máscaras que aproximam a magnitude do gradiente com as diferenças dos níveis de cinza da imagem. Os dois gradientes representam, respectivamente, as variações de intensidade dos níveis de cinza no eixo horizontal ( $G_x$ ) e vertical ( $G_y$ ). Ao convolucionar essas máscaras com a imagem, as variações de intensidade dos níveis de cinza em cada eixo são detectadas. O resultado dessa operação é calculado pela fórmula abaixo. Este valor resultante deve ser aplicado a um limiar, identificando se o ponto é ou não a borda. (POHREN, 2018).

$$\sqrt{G_x^2 + G_y^2}$$

O processo é feito através de duas máscaras de tamanho 3 x 3, sendo que uma é aplicada ao eixo  $x$  e outra ao eixo  $y$ . Esta operação realiza a suavização e, ao mesmo tempo, o realce das bordas.

$$G_x \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} G_y \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Considerando  $z_0$  a  $z_8$  os tons de cinza em torno do *pixel* que está sendo analisado, este filtro é definido pela equação de gradiente onde:

$$G_x = (z_6 + 2z_7 + z_8) - (z_0 + 2z_1 + z_2)$$

$$G_y = (z_2 + 2z_5 + z_8) - (z_0 + 2z_3 + z_6)$$

### 2.1.5.2. Frei-Chen

O operador Frei-Chen utiliza um conjunto de 9 máscaras de tamanho 3 x 3. As máscaras  $M1$  a  $M4$  são utilizadas para realizar a detecção de bordas. As máscaras  $M5$  a  $M8$  realizam a detecção de retas. A máscara  $M9$  representa a média dos *pixels* na região da matriz (BECKER, 2018).

$$M1 \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} M2 \begin{bmatrix} 0 & 1 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} M3 \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix}$$

$$M4 \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} M5 \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} M6 \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

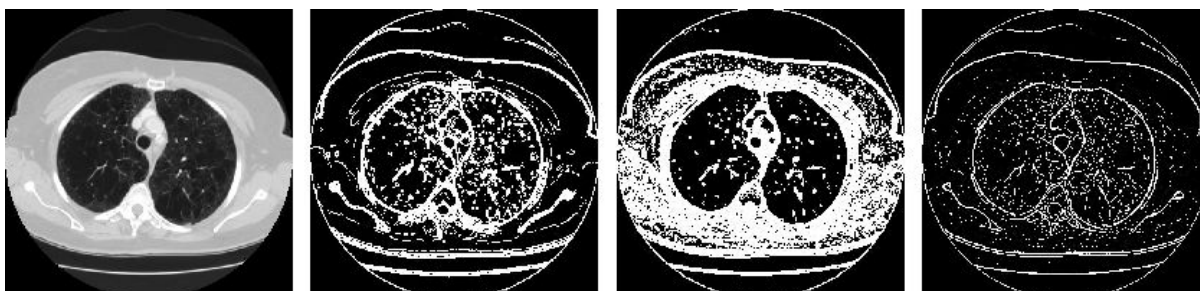
$$M7 \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} M8 \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} M9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & -1 \end{bmatrix}$$

### 2.1.5.3. Marr and Hildreth

A técnica de Marr and Hildreth utiliza somente uma máscara para realizar a convolução. Segundo Becker (2018), o algoritmo usa uma máscara de tamanho 7 x 7 para a operação. A matriz é descrita por:

$$H \begin{bmatrix} 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -2 & -3 & -3 & -3 & -2 & 0 \\ -1 & -3 & 5 & 5 & 5 & -3 & -1 \\ -1 & -3 & 5 & 16 & 5 & -3 & -1 \\ -1 & -3 & 5 & 5 & 5 & -3 & -1 \\ 0 & -2 & -3 & -3 & -3 & -2 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 \end{bmatrix}$$

Figura 5 - Aplicação de filtros passa-altas em uma imagem



Fonte: elaborado pelo autor

A Figura 5 apresenta uma imagem alterada pelos algoritmos de detecção de bordas. Da esquerda para a direita as imagens são as seguintes: imagem original, Sobel, Frei-Chen e Marr and Hildreth. Vale salientar que a imagem original não passou por nenhum filtro de remoção de ruídos e, por isso, os algoritmos de detecção de bordas identificaram diversos elementos indesejados.

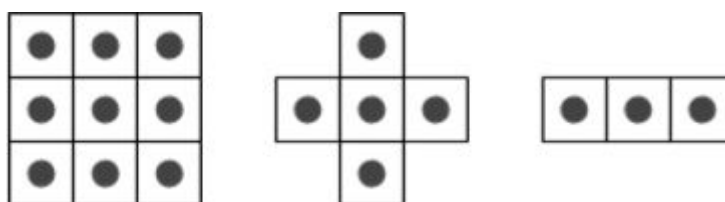
### 2.1.6. Morfologia Matemática

O termo morfologia se refere ao entendimento da forma e da estrutura de um objeto estudado. Gonzalez e Woods (2010) apresentam a definição de morfologia matemática, no contexto de Processamento Digital de Imagens, como “uma ferramenta para extrair



componentes das imagens que são úteis na representação e na descrição da forma de uma região”. É um conceito que utiliza como linguagem a teoria dos conjuntos. Para as operações morfológicas sobre a imagem é realizado um mapeamento entre um conjunto  $a$  que define a imagem e um conjunto  $b$  chamado de objeto estruturante, que são pequenos conjuntos utilizados para buscar propriedades de interesse.

**Figura 6 - Exemplos de elementos estruturantes**



Fonte: Becker (2018)

A morfologia matemática é uma metodologia aplicável em diversas áreas, permitindo a busca de padrões, extração e afinamento de bordas, além do preenchimento de pequenas deformações nas imagens (BECKER, 2018). Pedrini e Schwartz (2008) explicam que as operações de morfologia matemática são realizadas a partir de um elemento estruturante que é utilizado para percorrer a imagem processada. A cada ciclo é realizada uma operação matemática sobre os *pixels* do elemento estruturante, alterando a imagem.

Esta seção apresenta duas operações básicas da morfologia matemática: erosão e dilatação. Também apresenta as operações de abertura e fechamento, que consistem na combinação das duas primeiras.

#### **2.1.6.1 Erosão**

A erosão é uma operação baseada na aplicação de um elemento estruturante sobre a imagem que resulta na redução ou remoção de objetos e detalhes considerados irrelevantes. (RÖNNAU, 2015). Se o objeto for menor que o elemento estruturante, ele é completamente removido, portanto, essa técnica pode ser utilizada para eliminar ruídos de uma imagem (GONZALEZ; WOODS, 2010). Pedrini e Schwartz (2008) definem a fórmula que descreve a operação de erosão em imagens de tons de cinza como:

$$(f \ominus b)(x, y) = \min\{f(x - m, y - n) - b(m, n)\}$$

Em que  $(m, n) \in b$ , que é o elemento estruturante. A cada ciclo, o resultado é subtraído dos *pixels* avaliados.

### 2.1.6.2. Dilatação

A dilatação é uma operação inversa à erosão, que foi apresentada na seção anterior. É uma técnica que se baseia na aplicação de um elemento estruturante sobre os pontos de uma imagem que pode provocar uma adição de informações na vizinhança destes pontos. (NOGUEROL, 2008). A dilatação é comumente utilizada com o propósito de preencher lacunas e intervalos indesejados (RÖNNAU, 2015). De acordo com Pedrini e Schwartz (2008), a fórmula que descreve a operação de dilatação em imagens de tons de cinza é definida por:

$$(f \oplus b)(x, y) = \max\{f(x - m, y - n) + b(m, n)\}$$

Em que  $(m, n) \in b$  tal que  $(m, n) = (0, 0)$  é a origem do elemento estruturante  $b$ . A cada ciclo, o resultado de maior valor será o substituto dos *pixels* avaliados na imagem destino.

### 2.1.6.3. Abertura

A abertura é um processo que tem por objetivo suavizar os contornos dos objetos, podendo eliminar conexões estreitas entre os objetos e remover saliências finas (RÖNNAU, 2015). Após a aplicação desse procedimento, os objetos menores que os elementos estruturantes são removidos da imagem (GONZALEZ; WOODS, 2010).

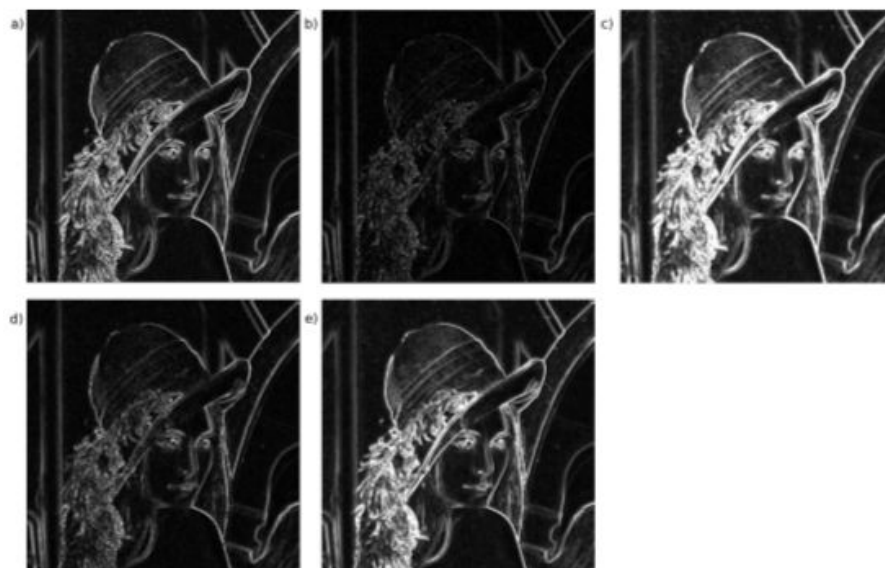
O processo de abertura consiste na aplicação de uma erosão e de uma dilatação na mesma imagem, nesta ordem. Ambas operações devem usar o mesmo elemento estruturante (PEDRINI; SCHWARTZ, 2008). É importante salientar que, apesar da erosão e da dilatação serem operações contrárias, nem sempre a aplicação de uma consegue reverter a aplicação da outra. E é por isso que a combinação delas, na ordem apropriada, causa os efeitos de abertura e fechamento, que é apresentado na seção a seguir (RÖNNNAU, 2015).

#### 2.1.6.4. Fechamento

Segundo Becker (2018), a operação de fechamento também causa uma suavização da imagem. A diferença em relação à abertura é que ele preenche pequenos buracos e preenche as lacunas no contorno de uma imagem.

O processo de fechamento consiste na aplicação de uma dilatação e de uma erosão na mesma imagem, nesta ordem. Ambas operações devem usar o mesmo elemento estruturante (PEDRINI; SCHWARTZ, 2008).

**Figura 7 - Aplicações de operações de morfologia matemática em uma imagem**



Fonte: Becker (2018)

Na Figura 7 é apresentada a aplicação de operações de morfologia matemática, onde (a) representa a imagem original. A primeira operação apresentada é a erosão (b), seguida da operação de dilatação (c), abertura (d) e fechamento (e).

## 2.2. TÉCNICAS DE SEGMENTAÇÃO

A segmentação de imagens tem por objetivo dividir a imagem em regiões ou objetos de interesse para que seja possível identificá-los e diferenciá-los do restante da imagem. Essa é uma das tarefas mais difíceis e complexas no processamento de imagens e é decisiva para definir o sucesso ou insucesso do processo final de análise (RÖNNAU, 2015).

Esse processo pode basear-se em diferentes aspectos das imagens, como regiões contínuas, bordas e/ou limiares (FLORES, 2015). Normalmente são usados como base as propriedades dos níveis de cinza, a descontinuidade e a similaridade (BECKER, 2018). Segundo Pedrini e Schwartz (2008), os métodos de descontinuidade, que foram tratados na seção 2.1.5, visam a identificação de objetos a partir de mudanças abruptas nos níveis de cinza da imagem, o que caracteriza sua borda. Já os métodos baseados na similaridade têm como propósito agrupar pontos semelhantes da imagem para um determinado conjunto de características. A seguir é apresentado o método de crescimento de regiões.

### 2.2.1. Crescimento de regiões

Métodos baseados em crescimento utilizam um ponto semente (do inglês, *seed pixel*) para cada objeto de interesse. Essa semente pode ser selecionada de forma automática ou indicada por um operador. O algoritmo, a partir da semente, examina sua vizinhança em busca dos *pixels* que pertençam a determinado critério de uniformidade, também denominado critério de homogeneidade. Esse processo se repete até que não se encontre mais nenhum *pixel* que atende aos critérios de homogeneidade (MOSSMANN, 2007).

Métodos de segmentação baseados no crescimento de regiões separam os objetos de uma imagem a partir de *pixels* que tenham características em comum entre si, com baixa tolerância a variações (RÖNNAU, 2015). Outro requisito é que as regiões tenham bordas bem definidas e fechadas. Desta forma, o conteúdo do interior do objeto fica separado do restante da imagem (CONCI; AZEVEDO; LETA, 2008).

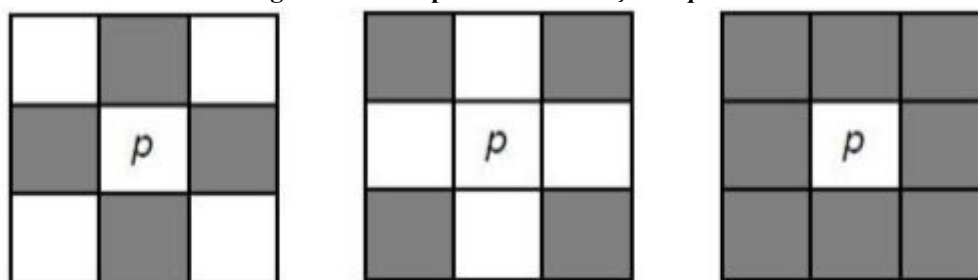
A escolha da semente é fundamental para o sucesso da operação, já que ele é o ponto de partida para a execução do algoritmo (CONCI; AZEVEDO; LETA, 2008). Outro fator importante é a definição dos critérios de parada, pois o crescimento deve ser interrompido quando não houver mais *pixels* vizinhos que atendam aos critérios (BECKER, 2018). Esses critérios podem considerar a comparação com os *pixels* vizinhos, a forma e o tamanho (RÖNNAU, 2015).

## 2.3. TÉCNICAS DE CLASSIFICAÇÃO

### 2.3.1. Rotulação

O processo de rotulação consiste na enumeração ou nomeação das regiões identificadas na segmentação. É utilizado para localizar os objetos em uma imagem e seu funcionamento acontece buscando *pixels* conectados e atribuindo um mesmo valor para cada conjunto (DAMBROS, 2008). É um processo fundamental para a segmentação e contagem de regiões e objetos (CONCI; AZEVEDO; LETA, 2008).

O algoritmo de rotulação percorre a imagem *pixel* a *pixel*. Para cada *pixel*, são avaliados os vizinhos segundo critérios básicos de distância e, em caso afirmativo, o rótulo é atribuído (FLORES, 2015).

Figura 8 - Exemplos de vizinhanças de *pixels*

Fonte: adaptado de Flores (2015)

Na Figura 8 são apresentados alguns exemplos de vizinhança comumente encontrados na literatura. À esquerda é apresentada uma vizinhança  $N_4(p)$  (composta dos 4 *pixels* diretamente conectados a  $p$ ), ao centro uma vizinhança  $N_D(p)$  (composta dos 4 *pixels* indiretamente conectados a  $p$  nas diagonais) e à direita uma vizinhança  $N_8(p)$ , composta de todos os *pixels* que circundam  $p$ .

O processamento ocorre percorrendo os *pixels* da imagem até que seja encontrado um *pixel*  $p$  cuja cor seja diferente do fundo da imagem. Então são verificados os *pixels* que estão conectados a  $p$  pelo critério de vizinhança. Se o *pixel* que está conectado já possui rótulo atribuído, então  $p$  recebe o mesmo rótulo. Se o *pixel* conectado ainda não possui rótulo atribuído, ambos recebem um novo rótulo e único. Esse procedimento é repetido até que todos os *pixels* da imagem sejam analisados. Caso a análise encontre um *pixel* pertencente a duas ou mais regiões, todas devem assumir o mesmo rótulo, pois passam a representar uma única região (FLORES, 2015).

## 2.4. CONSIDERAÇÕES

Este capítulo apresentou algumas técnicas que foram utilizadas no desenvolvimento do protótipo deste trabalho. Cada uma delas compõe ou serviu como base para a elaboração do algoritmo de segmentação de imagens.

A seção 2.1 introduz o conceito de pré-processamento de imagens digitais. Inicialmente, são apresentados os conceitos de Brilho, Contraste, Limiarização (*threshold*) e Binarização. Em seguida, são abordadas técnicas de remoção de ruídos, também conhecidos como filtros passa-baixas. Entre elas, destacam-se o Filtro de média, o Filtro de moda, o Filtro de mediana e o Filtro Gaussiano. Na sequência são apresentados algoritmos de detecção de bordas, também denominados filtros passa-altas. São explicados os filtros de Sobel, de

Frei-Chen e de Marr and Hildreth. Por fim, é feita uma explanação acerca de morfologia matemática em imagens digitais. Os métodos apresentados são Erosão, Dilatação, Abertura e Fechamento.

A seção 2.2 aprofunda o estudo na direção das técnicas de segmentação de imagens. O método apresentado é o de Crescimento de regiões, que utiliza um ponto semente (*seed pixel*, do inglês) para iniciar o processamento rumo à identificação das regiões de interesse.

A seção 2.3 fecha o conjunto de processos aplicáveis a imagens digitais abordando técnicas de classificação. É apresentado o método de Rotulação, que designa uma identificação única para cada região de interesse presente na imagem.

No próximo capítulo são apresentados trabalhos correlatos produzidos na mesma área de estudo deste trabalho. São trabalhos que aplicam alguns dos conceitos apresentados acima e que propõem soluções voltadas para o mesmo propósito do aqui desenvolvido.

### 3. TRABALHOS CORRELATOS

Diversos trabalhos já foram produzidos no sentido de automatizar a segmentação de órgãos em Tomografias Computadorizadas. Na sequência são apresentadas pesquisas científicas desenvolvidas que contribuem propondo algum algoritmo ou técnica nova para a solução do problema.

#### 3.1. LUNG SEGMENTATION IN CHEST COMPUTERIZED TOMOGRAPHY IMAGES USING THE BORDER FOLLOWING ALGORITHM

Rodrigues *et al.* (2016) desenvolveram um algoritmo de segmentação de pulmões em Tomografias Computadorizadas chamado *Follower of Lung Contour* (FLC). A proposta consiste em duas etapas: inicialmente, uma rede neural artificial (RNA) classifica os *pixels* da imagem como pertencentes à região pulmonária ou não. Para treinar a RNA são utilizados 6 resultados obtidos a partir do *Pulmonary Densities Analysis* (PDA) proposto por Rebouças Filho *et al.* (2013).

Em seguida, é acionado um algoritmo de *Border Following* (GONZALEZ; WOODS, 2010) na imagem gerada na etapa anterior para identificar as regiões dos pulmões. O algoritmo percorre a imagem iniciando no canto superior esquerdo e deslocando-se para baixo e para a direita até encontrar o primeiro *pixel* pertencente à região pulmonária. Na sequência, busca, no sentido horário, *pixels* pertencentes à borda até voltar ao primeiro *pixel* encontrado, finalizando a construção do contorno. Conforme os autores, esse tipo de busca é mais eficiente que o PDA pois apresenta resultados semelhantes e sua implementação é mais simples. Porém, a técnica de detecção de bordas utilizada não considera a densidade de *pixels* na região e, portanto, pode classificar erroneamente *pixels* de ruído.

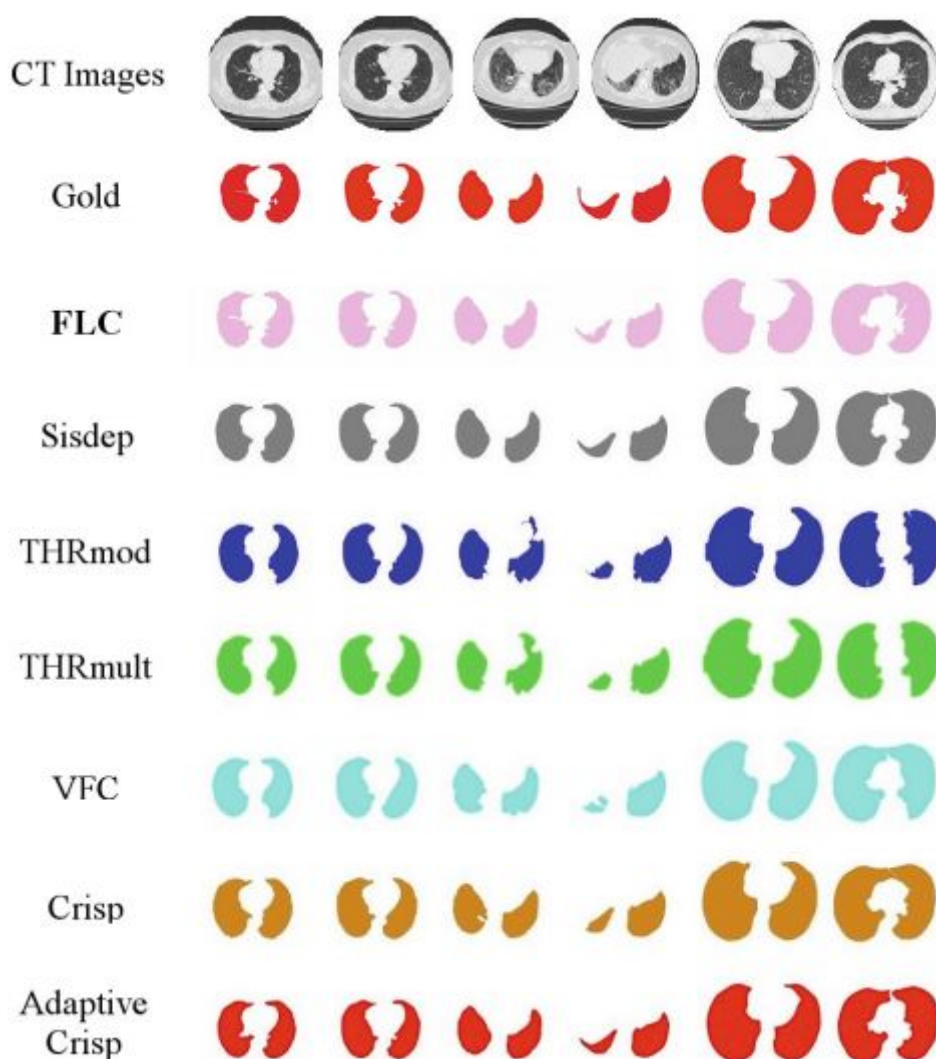
A execução do algoritmo de *Border Following* nas imagens resulta em um conjunto de contornos encontrados, incluindo regiões que não pertencem aos pulmões. Portanto, foram definidos critérios para classificação. As imagens das fatias de Tomografia Computadorizada foram divididas em quatro regiões: superior, inferior, esquerda e direita. A partir disso, foram analisados os centróides dos contornos encontrados para classificá-los como pulmão esquerdo e direito. Para que um pulmão seja classificado como “esquerdo” ou “direito”, sua coordenada central precisa estar localizada no lado esquerdo ou direito da imagem, respectivamente. Se houver mais que um contorno no mesmo lado da imagem, aquele com a maior área é



considerado como pulmão, pois os pulmões são os maiores objetos das regiões esquerda e direita.

Os resultados do método proposto foram comparados com as técnicas SISDEP, VFC, THRmult, THRmod, Crisp e Adaptative Crisp. Foi calculada a taxa de precisão e a média harmônica dessas técnicas em relação ao padrão ouro, que é o resultado obtido de uma segmentação manual feita por um especialista. Com base nesses valores pode-se comparar a eficácia do FLC em relação aos demais algoritmos. A Figura 9 apresenta os resultados obtidos com o uso de todas as técnicas supracitadas.

**Figura 9 - Resultados obtidos por Rodrigues *et al.* (2016)**



Fonte: Rodrigues *et al.* (2016)

A Tabela 1 apresenta um fragmento dos resultados, com a precisão (ACC) e a média harmônica (HM) de cada um dos métodos em relação ao padrão ouro. Conforme pode ser percebido, o método FLC se destaca em termos de precisão, seguido do Adaptive Crisp e SISDEP. O FLC é também o melhor na média harmônica, com 98,69%. Um ponto destacado pelos autores é que a técnica proposta se mantém constante nos diferentes casos examinados e alcança resultados próximos aos obtidos pelos especialistas. Com base nas métricas utilizadas, os autores concluíram que o FLC possui uma boa precisão e uma boa taxa de sucesso.

**Tabela 1 - Fragmento dos resultados obtidos pelo *Follower of Lung Contour***

Image number	FLC		ACACM		Crisp		Sisdep		THRmod		THRmult		VFC	
	ACC (%)	HM (%)	ACC (%)	HM (%)	ACC (%)	HM (%)	ACC (%)	HM (%)	ACC (%)	HM (%)	ACC (%)	HM (%)	ACC (%)	HM (%)
Average	<b>98.8</b>	<b>98.69</b>	98.76	98.46	97.94	97.94	98.64	98.32	97.37	96.17	96.24	92.34	97.71	96.73
SD	<b>0.65</b>	<b>1.42</b>	0.69	1.42	0.90	1.29	0.73	1.32	1.75	2.70	2.79	6.31	1.39	2.64

Fonte: Rodrigues *et al.* (2016)

### 3.2. AUTOMATIC SEGMENTATION AND MEASUREMENT OF THE LUNGS IN HEALTHY PERSONS AND IN PATIENTS WITH CHRONIC OBSTRUCTIVE PULMONARY DISEASE IN CT IMAGES

No artigo de Felix *et al.* (2007) é utilizada a técnica de crescimento de regiões para identificar os pulmões. Inicialmente, é estabelecido um *pixel* inicial (*seed pixel*, no inglês) de forma automática utilizando a técnica proposta por Araújo (2004). A partir disso, a busca faz o crescimento da região no entorno do *pixel* inicial seguindo os critérios apresentados na sequência. O crescimento da região é interrompido quando a primeira ou a segunda regra são descumpridas.

- A diferença absoluta entre a escala de cinza do *pixel* inicial com o candidato a ser incorporado à região é menor que um determinado limiar.
- O *pixel* candidato é conectado por um dos 8 *pixels* vizinhos ao *pixel* inicial.
- Se um *pixel* pertence a duas regiões, ele é arbitrariamente atribuído à região com o maior nível de cinza.

Com base nos dados extraídos, os autores concluem que é possível segmentar precisamente tanto pulmões saudáveis quanto de pacientes com doença de obstrução

pulmonar crônica (DPOC), bem como, calcular a área e o perímetro de cada fatia dos pulmões. Não foram citados dados sobre a validação dos resultados no trabalho. Na Figura 10, é apresentado um exemplo de segmentação de pulmões utilizando o método proposto.

**Figura 10 - Resultados apresentados por Felix *et al.* (2007)**



Fonte: Felix *et al.* (2007)

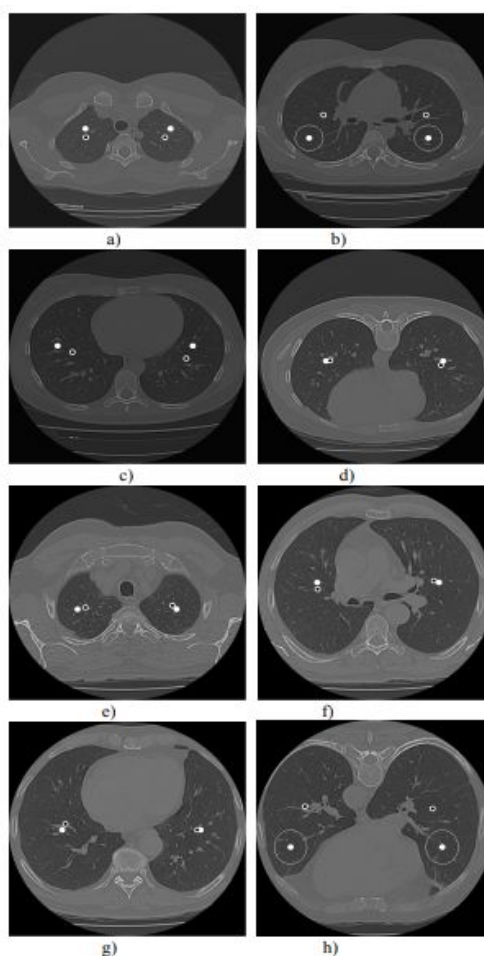
### 3.3. AUTOIN: METHOD OF AUTOMATIC INITIALIZATION OF ACTIVE CONTOURS APPLIED TO LUNGS IN CT IMAGES

Felix *et al.* (2012) também apresentam, em outro trabalho, o AUTOIN, que é uma abordagem para gerar automaticamente o *pixel* inicial (*seed*) a partir de diversas transformações e operações sobre as imagens de TC. Inicialmente, é feita a binarização da imagem inteira e removidos ruídos e estruturas que podem comprometer as etapas seguintes. Em seguida, é aplicada a função de projeção híbrida no sentido horizontal e vertical. Por fim, é selecionada a coordenada de cada pico das funções obtidas anteriormente, gerando os pontos de inicialização dos contornos na imagem final.

Os resultados foram comparados com o algoritmo de localização do centro de objeto (LOC) proposto por Li e Wong (1997), que permite identificar o centro de objetos com diferentes formas. Foram utilizadas 72 imagens no desenvolvimento do trabalho. Na Figura

11 são apresentados alguns resultados obtidos sobre essas imagens tanto com o algoritmo LOC quanto com o AUTOIN. O primeiro é indicado por um círculo de borda branca com centro preto e o segundo por um círculo branco. Para o *dataset* utilizado, as técnicas apresentam resultados semelhantes e ambas geram *seeds* apropriados para as regiões de interesse. Portanto, se ambas fossem utilizadas para inicializar o mesmo método de contornos ativos (MCA), o resultado final seria o mesmo.

**Figura 11 - Resultados do AUTOIN**



Fonte: Felix *et al.* (2012)

Essa abordagem pode ser utilizada como ponto de partida para técnicas que necessitam de um ponto inicial para sua execução. Nos testes realizados pelos autores, ela atende satisfatoriamente o objetivo de identificar o *seed* de regiões pulmonares e pode ser utilizada em outras formas anatômicas.

### 3.4. SEGMENTAÇÃO AUTOMÁTICA DE ÓRGÃOS EM IMAGENS DE TOMOGRAFIA COMPUTADORIZADA DO TÓRAX

Rönnau (2015) propõe uma técnica baseada no pré-processamento das imagens e rotulação das regiões de interesse para identificar os pulmões em TC. Inicialmente, é aplicado um Filtro Gaussiano a fim de eliminar eventuais ruídos e os detalhes que não são relevantes para o processamento. Em seguida, é aplicada uma função de limiarização (também denominado *threshold*, em inglês), que converte os *pixels* da região de interesse em branco e os demais em preto, tendo como resultado uma imagem somente com dois valores de *pixels*. Para identificar as regiões pulmonares, é aplicado um algoritmo de rotulação por crescimento de regiões.

Os resultados foram avaliados visualmente, levando em consideração o acerto ou erro das regiões identificadas pelo algoritmo. Para tal, foram utilizados 5 exames, com um total de 569 fatias que foram processadas. Os resultados são apresentados na Tabela 2.

**Tabela 2 - Resultados apresentados por Rönnau (2015)**

	Exames	Quantidade de imagens	Espessura da fatia	Acertos	Erros	Tempo de processamento
	Exame01	93	2,5mm	79	14	4s 212ms
	Exame02	94	2,5mm	86	8	4s 69ms
	Exame03	77	2,5mm	76	1	3s 653ms
	Exame04	119	2,5mm	113	6	4s 818ms
	Exame05	186	1,3mm	183	3	8s 385ms
<b>Totais</b>	<b>5</b>	<b>569</b>	<b>-</b>	<b>537</b>	<b>32</b>	<b>25s 137ms</b>

Fonte: Rönnau (2015)

### 3.5. COMPARATIVO DAS TÉCNICAS

A seguir, são agrupados os resultados dos trabalhos apresentados acima e feita uma comparação entre eles. O artigo de Felix *et al.* (2007) não apresenta os resultados do estudo e, portanto, não será considerado neste comparativo.

O algoritmo AUTOIN (Felix *et al.*, 2012) tem como propósito a identificação de um *seed* inicial para um algoritmo de segmentação de pulmões e, portanto, não pode ser comparado com as técnicas que realizam o processamento completo. No artigo, os autores apresentam um comparativo com o LOC proposto por Li e Wong (1997). Em média, o

AUTOIN gerou um *seed* a uma distância de 12 mm do gerado pelo LOC em pulmões sadios. Os dados apresentados sempre geraram um *pixel* inicial dentro da região dos pulmões.

Rodrigues *et al.* (2016) desenvolveram o método chamado *Follower of Lung Contour* (FLC). Em sua validação foram utilizadas 30 imagens de Tomografias Computadorizadas do tórax e sua precisão foi de 98,8% nos casos analisados.

Rönnau (2015) apresentou uma proposta que envolve um conjunto de técnicas no processamento dos exames de TC e realizou uma validação envolvendo 569 fatias de 5 exames diferentes. Desse total, o algoritmo rotulou corretamente 537 imagens, ou seja, apresentou taxa de acerto de 94,68%.

**Tabela 3 - Comparativo entre os trabalhos de Rodrigues *et al.* (2016) e Rönnau (2015)**

	FLC (Rodrigues <i>et al.</i> , 2016)	Rönnau (2015)
Imagens usadas na validação	30	569
Taxa de acerto	98,8%	94,68%

Fonte: elaborado pelo autor

A Tabela 3 consolida os dados dos 2 trabalhos que fazem o processamento completo dos exames e que os autores apresentaram os resultados numericamente. Ambas as técnicas apresentam uma boa taxa de acerto nas fatias intermediárias dos exames de TC. Isso ocorre pois, nessas fatias, os pulmões possuem uma grande área e uma forma bem definida. Nas fatias superiores e inferiores essa assertividade diminui, pois a área dos pulmões diminui ao ponto de serem menores que a traquéia, por exemplo. Rönnau (2015) é o único que considera essas fatias em suas análises e, por isso, a taxa de acerto do algoritmo por ele proposto teve uma leve diminuição.

### 3.6. CONSIDERAÇÕES

Este capítulo apresentou técnicas propostas em trabalhos correlatos, detalhando seus métodos e processos e demonstrando seus resultados. Por fim, os dados foram consolidados e feito um comparativo entre eles. O capítulo a seguir aprofunda o estudo sobre os arquivos no

padrão DICOM, bem como especifica a base de dados utilizada na validação do protótipo desenvolvido durante a elaboração deste trabalho.

## 4. EXAMES MÉDICOS POR IMAGEM

Este capítulo apresenta o uso de imagens na área médica, destacando a Tomografia Computadorizada. Em seguida, é apresentado o formato de arquivos DICOM, que é utilizado para aquisição e transmissão de dados médicos.

Na área médica, um dos principais artifícios utilizados no auxílio ao diagnóstico são os exames com imagens, que permitem uma visualização interna dos tecidos do corpo sem a necessidade de operações invasivas. O uso de imagens possibilita ao médico identificar e determinar anomalias, aumentando a precisão e confiabilidade do diagnóstico (DRAKE; VOGL; MITCHELL, 2010).

### 4.1. TOMOGRAFIA COMPUTADORIZADA

Através da Tomografia Computadorizada pode ser feita uma análise de todo o corpo humano. Com ela é possível realizar um exame com grande detalhamento anatômico, pois é baseada na aquisição de imagens que representam cortes transversais do paciente, sem sobreposição de estruturas anatômicas (NOVELLINE, 1999).

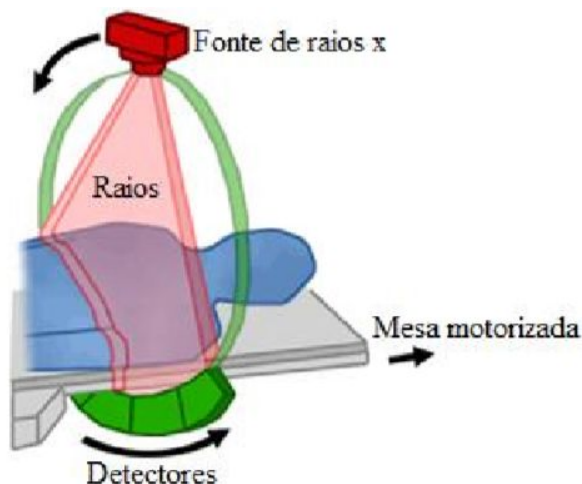
A Figura 12 ilustra o uso de um equipamento de aquisição das imagens, cujo funcionamento é descrito em Stimac (1994, p. 4) conforme segue:

A ampola de raios X produz um feixe em forma de leque que atravessa uma secção (corte) do paciente. Este feixe em forma de leque é recebido por um arranjo circular de detectores do lado oposto ao feixe. Estes detectores recebem raios X ativados durante sua passagem a longo de um trajeto específico através do corpo do paciente. O detector e a fonte de raios X rodam em torno do eixo, produzindo exposição a intervalos de 1 grau de rotação.

O arco detector recebe raios X e os converte em sinais elétricos. A intensidade desses sinais é inversamente proporcional à quantidade de raios X absorvidos pelos tecidos corporais. Quanto mais denso o tecido, maior é a absorção, portanto, os ossos absorvem mais raios que os pulmões, por exemplo (NOVELLINE, 1999).



Figura 12 - Funcionamento de um scanner de TC



Fonte: adaptado de FDA(2015)

A distância entre os cortes, também conhecidos como *slices*, varia de 1 a 10 milímetros e o processo de aquisição é repetido de 300 a 600 vezes por corte. (STIMAC, 1994). O número de *slices* em um exame pode variar de acordo com o equipamento utilizado, distância entre fatias e seção do corpo que está sendo analisado (RÖNNAU, 2015).

Os dados obtidos pelos detectores são enviados para um dispositivo que irá processá-los através de técnicas de reconstrução matemática. Cada ponto da imagem é tratado como um coeficiente de atenuação, que no caso da TC é chamado de *voxel*. Um *voxel* representa uma unidade de volume e tem seu coeficiente representado em Unidades *Hounsfield*, que é tema da seção 4.2.1 (RÖNNAU, 2015).

#### 4.2. PADRÃO DICOM

Na área médica, as imagens são amplamente utilizadas para apoiar o diagnóstico em pacientes. São aplicados diversos métodos para coleta de imagens do corpo humano, que são utilizadas para auxiliar os especialistas no processo de tomada de decisão. Desde os anos 80 são utilizados equipamentos para esse tipo de operação, porém, nos primórdios não existia um padrão definido. Assim, cada fabricante adotava seu próprio formato de representação e armazenamento dos dados. Com o passar dos anos, notou-se a incompatibilidade entre os diferentes equipamentos e observou-se a necessidade de padronizar o modo como as imagens eram geradas, armazenadas e transmitidas (FLORES, 2015).

Em 1983, a *American College of Radiology* (ACR) e a *National Electrical Manufacturers Association* (NEMA) se uniram para definir um padrão de geração,

armazenamento, comunicação e manipulação de imagens médicas. Esse padrão, que é utilizado até os dias atuais, recebeu o nome de DICOM, sigla para *Digital Imaging and Communication in Medicine* (NEMA, 2015).

O formato DICOM não armazena somente as imagens dos exames, mas “abrange e especifica diferentes tecnologias, como a rede de comunicação entre os equipamentos, a sintaxe e a semântica dos comandos utilizados no protocolo, a estrutura de dispositivos e os diretórios de armazenamento, além do formato dos arquivos que contêm imagens e as informações dos exames e pacientes” (RÖNNAU, 2015).

#### 4.2.1. Escala *Hounsfield*

Godfrey Hounsfield foi um dos inventores da Tomografia Computadorizada. Seu sobrenome foi utilizado para denominar a escala que representa o coeficiente de atenuação de um *voxel* nesse tipo de exame (NOVELLINE, 1999).

A escala *Hounsfield* compreende valores entre -1000 HU (*Hounsfield Unit*), que representa o ar, e +3000 HU, que representa ossos com alta densidade. O valor 0 HU é utilizado como referência e representa a água (STIMAC, 1994).

Ao representar visualmente as imagens de uma TC, é possível definir cada ponto do exame como um valor na escala de cinza de 0 a 255, ou seja, há 256 valores possíveis. A escala *Hounsfield* permite a utilização de 4000 valores distintos, o que justifica a utilização do conceito de janela, que é apresentado na seção 4.2.2 (RÖNNAU, 2015). Na Tabela 4 são apresentados os valores HU que representam algumas estruturas anatômicas visíveis nos exames de TC.

**Tabela 4 – Valores de exemplo em HU**

Substância	Densidade (HU)
Ar	-1000
Gordura	-50 a -300
Água	0
Líquido cefalorraquidiano	0 a 10
Sangue	30 a 60
Fígado	60 a 80
Músculo	40 a 80
Osso	200 a 2000

Fonte: Stimac (1994, p. 6)

#### 4.2.2. Nível de janela e Largura de janela

A escala *Hounsfield* compreende uma esfera de 4000 valores para representar todas as densidades possíveis dos tecidos. Porém, em muitos casos, deseja-se realçar o valor correspondente ao tecido de maior interesse. Segundo Flores (2015), os conceitos de nível de janela (WL, do inglês *window level*) e largura de janela (WW, do inglês *window width*) foram criados com o objetivo de não descartar detalhes que podem ser significativos na análise de exames de TC.

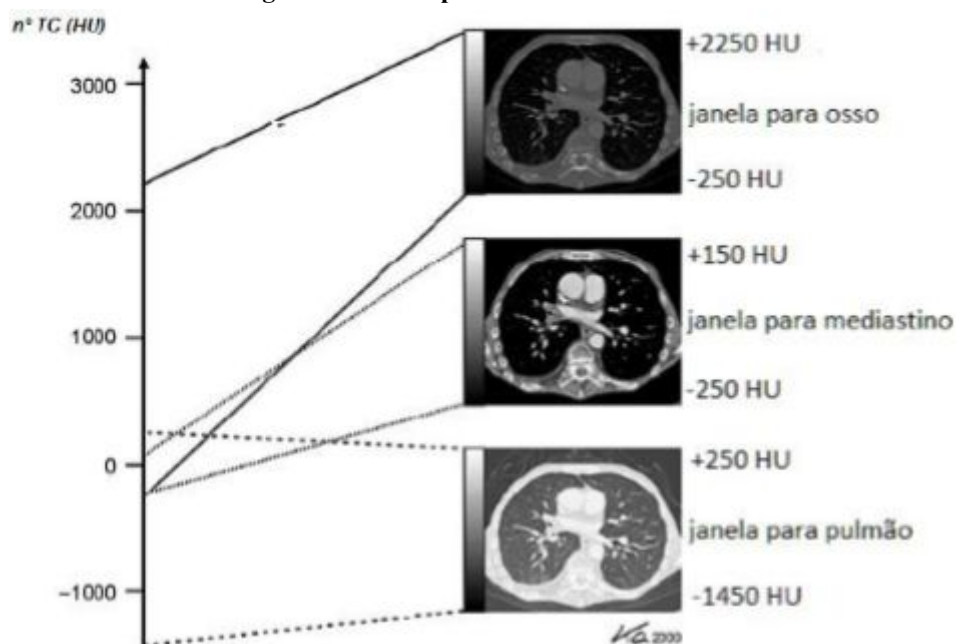
O nível de janela é aplicado para calibrar o nível de densidade média. Normalmente o *window level* é ajustado para que os níveis correspondentes ao tecido de maior interesse sejam representados por um valor intermediário no espectro de cinzas (RÖNNAU, 2015).

A largura de janela representa o intervalo de valores da escala *Hounsfield* que serão exibidos através dos 256 tons de cinza. Esse intervalo é aplicado em relação ao WL e, portanto, metade dele abrange os valores de HU abaixo do WL e a outra metade os valores acima (RÖNNAU, 2015).

Em um cenário hipotético com WL igual a 1000 e WW igual a 500, os valores de +750 a +1000 serão representados por 128 tons de cinza, enquanto os valores de +1000 a +1250 pelos outros 128 tons de cinza. Valores abaixo de +750 assumirão a cor preta e valores acima de +1250 a cor branca. Nesse exemplo, cada tom de cinza agrupa 2 HU.

Com os parâmetros de *window level* e *window width* é possível configurar as informações que serão exibidas conforme a necessidade. O intervalo de HU agrupado por cada tom de cinza é diretamente proporcional ao WW. Isso significa que quanto menor o WW, menor é o intervalo de HU representado por cada um dos tons de cinza e maior é a fidelidade na representação visual da TC. Para que todos os tecidos e estruturas sejam observáveis, um corte deve ser visualizado mais de uma vez, com diferentes valores de WL e WW (STIMAC, 1994).

Figura 13 – Exemplos de uso de WL e WW



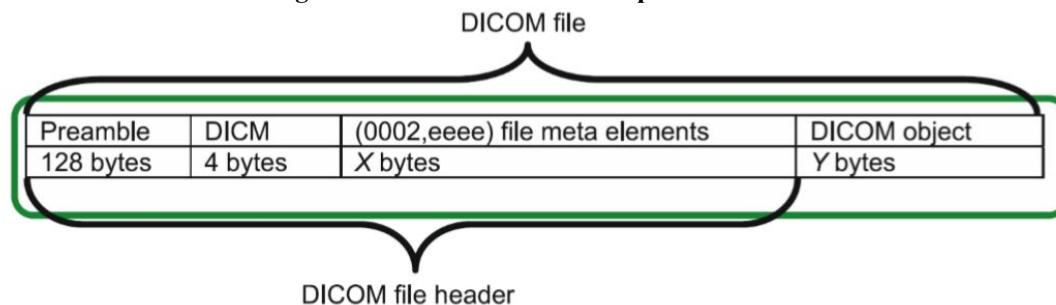
Fonte: Augusto (2009, p. 14)

Na Figura 13 é apresentado um exemplo de visualização com diferentes configurações de WL e WW. Em um mesmo corte, é possível visualizar diferentes estruturas anatômicas através da variação desses níveis.

#### 4.2.3. Formato de arquivos

Arquivos no formato DICOM são divididos em 2 partes: o *header* (cabeçalho) e o *dataset* (FLORES, 2015). A Figura 14 ilustra essa estrutura.

Figura 14 – Estrutura de um arquivo DICOM



Fonte: Pianykh (2008, p. 222)

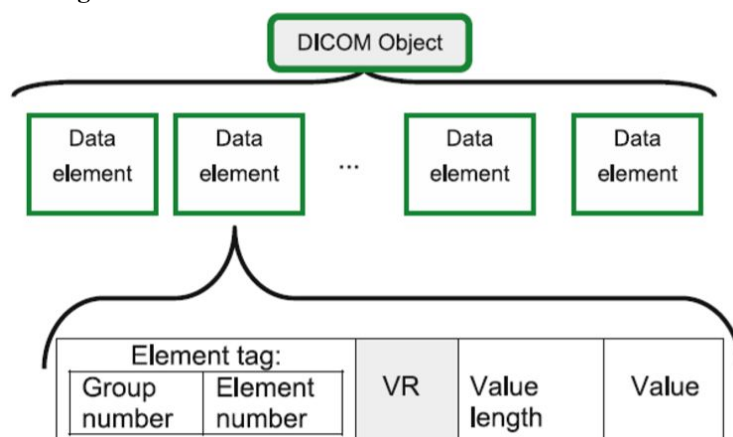
A primeira parte do *header*, denominada *preamble*, é um bloco de 128 bytes que pode ser utilizado de maneira livre, pois o padrão não define seu conteúdo. Como não há uma padronização acerca dessa área, as aplicações normalmente ignoram esse espaço. O elemento

que segue o *preamble* é o prefixo, cujo valor é sempre a literal “DICM”, que indica aos softwares que o arquivo está no formato DICOM. Não existem regras de codificação envolvendo essas duas partes, o que possibilita que qualquer aplicação consiga ler e interpretar (RÖNNAU, 2015).

Finalizando o *header*, a terceira parte é o *File Meta Information*, que tem seu início no byte 133. Ela armazena um conjunto de atributos, denominados *file meta elements*, com informações sobre o arquivo, como o tipo de codificação e a versão (NEMA, 2015).

O *dataset* é composto por uma série de *data elements* que são componentes independentes que agrupam dados de um determinado exame (FLORES, 2015). A Figura 15 apresenta o formato do *dataset* e do *data element*.

Figura 15 – Estrutura de um *dataset* e de um *data element*



Fonte: Pianykh (2008, p. 56)

Cada *meta element* e *data element* começa com uma *tag* que identifica o grupo e o elemento. A *tag* utilizada para descrever um *meta element* utiliza o grupo “0002”, enquanto a *tag* que identifica um *data element* utiliza o grupo “0008”. No exemplo da Figura 14, a *tag* (0002, eeee) utiliza o grupo “0002” e o identificador “eeee”. A partir desses dados, é possível diferenciar as informações do *header* das informações do objeto (PIANYKH, 2008).

NEMA (2015) indica que, além da *tag*, um atributo é composto pelos seguintes campos:

- VR (Value Representation)*: é uma sigla composta por dois caracteres que indica a forma como está representado o valor do atributo. Pode não estar informado, dependendo da codificação do arquivo;
- Value Length*: armazena o tamanho do campo de valor do elemento, em *bytes*. De acordo com o conteúdo do VR, esse campo pode ser formado por 16 ou 32 bits;

- c) *Value*: armazena o valor, ou valores, correspondente ao atributo. Seu tamanho é indicado pelo *value length* e o tipo de dado pelo VR.

Existem mais de 2000 atributos diferentes que são utilizados em arquivos DICOM, para representar as mais variadas informações sobre os exames, como dados do equipamento, dados do paciente e dados do exame. As imagens dos exames também compõem os *data elements* (PIANYKH, 2008). Alguns exemplos de elementos referentes a imagens que serão utilizados neste trabalho são:

- a) (0028,0010) *Image height*: corresponde ao número de linhas da imagem;
- b) (0028,0011) *Image width*: corresponde ao número de colunas da imagem;
- c) (7FE0,0010) *Image pixel data*: corresponde aos valores dos *pixels* da imagem. É responsável, normalmente, por 95% do tamanho físico de um arquivo de imagem no formato DICOM;
- d) (0028,0030) *Pixel spacing*: define o tamanho físico de cada *pixel* da imagem, permitindo que sejam feitas medidas reais a partir da imagem, convertendo *pixels* para milímetros;
- e) (0018,0088) *Spacing between slices*: armazena a distância física entre cada imagem da série.

Tipicamente, uma imagem de TC é representada por uma matriz quadrada de 512 linhas e 512 colunas, totalizando 262.144 *pixels*. O valor de cada um desses *pixels* é armazenado no atributo *image pixel data* (RÖNNAU, 2015).

### 4.3. BASES DE EXAMES MÉDICOS POR IMAGEM

Para o desenvolvimento deste trabalho, foi estruturada uma base de dados com exames no formato DICOM. As pesquisas resultaram em dois portais que disponibilizam exames médicos para download.

O primeiro portal é o Radiopaedia (2020). Ele contém diversos exames de TC do tórax disponíveis para estudo. Porém, não foi encontrada uma forma de obter os exames no formato DICOM, somente JPG, o que inviabiliza seu uso neste projeto.

Foram obtidos diversos exames do portal de imagens médicas *The Cancer Imaging Archive* (2020), os quais são apresentados a seguir. A primeira coleção, denominada QIBA

CT-1C, contém 20 exames de TC coletados por equipamentos da marca Siemens. Foram obtidos da segunda coleção, denominada LIDC-IDRI, dez exames de TC, cada um deles com características particulares. A Tabela 5 apresenta o conjunto de exames que constituem a base de dados para testes e validações deste trabalho.

**Tabela 5 - Detalhamento dos exames da base de dados**

Exame	Imagens
Exame01	93
Exame02	94
Exame03	77
Exame04	119
Exame05	186
Exame06	365
Exame07	365
Exame08	365
Exame09	365
Exame10	365
Exame11	365
Exame12	365
Exame13	365
Exame14	365
Exame15	365
Exame16	365
Exame17	365
Exame18	365
Exame19	365
Exame20	365
Exame01	128
Exame02	133
Exame03	253
Exame04	258
Exame05	122
Exame06	117
Exame07	290
Exame08	264
Exame09	238
Exame10	118

Fonte: elaborado pelo autor

#### 4.4. CONSIDERAÇÕES

Este capítulo apresentou os exames médicos por imagens, que serão a origem de dados para os testes do protótipo de software que será desenvolvido. Inicialmente, são abordados os exames de Tomografia Computadorizada, que formam o artefato utilizado para

coletar dados dos pacientes. Em seguida, é descrito o padrão de arquivos gerado por esses equipamentos, o DICOM. Por fim, é detalhada a base de dados utilizada para testes e validação do protótipo.



## 5. DESENVOLVIMENTO E VALIDAÇÃO DO PROTÓTIPO

O objetivo principal desse trabalho é a segmentação automática dos pulmões, visando a captura do maior número de detalhes possível. A partir do estudo da bibliografia e dos trabalhos correlatos, foi constatada uma lacuna presente em muitos trabalhos nessa área. Em geral, os algoritmos possuem boa precisão em fatias que apresentam os pulmões com forma bem definida e grande área, porém, quando são processadas as fatias próximas às extremidades superior e inferior do exame, a taxa de acerto diminui. Além disso, existem cenários em que, em determinada fatia do exame, o mesmo pulmão é exibido como duas ou mais figuras desconexas. Os trabalhos estudados consideram apenas a maior figura e descartam a outra parte do pulmão. Visando solucionar esse problema e aumentar a precisão do processo de segmentação automática dos órgãos, é proposta a solução de que trata o presente trabalho.

Este capítulo apresenta o desenvolvimento do protótipo e a validação dos resultados. Inicialmente, é detalhado o ambiente utilizado para a programação e testes, bem como os recursos utilizados para prepará-lo para o contexto do projeto. Em seguida, são abordados os algoritmos aplicados na construção da proposta de solução e os resultados obtidos durante a evolução do protótipo. A seção subsequente expõe a validação dos resultados obtidos.

### 5.1. PREPARAÇÃO DO AMBIENTE

O equipamento utilizado no desenvolvimento do protótipo é um computador com as seguintes características:

- Processador: AMD Ryzen 5 3600 6-Core Processor 3.59GHz
- Memória RAM: 8 GB
- SSD: 480 GB
- Sistema operacional: Windows 10 Pro

A linguagem de programação empregada é Java, que é amplamente difundida no meio acadêmico e corporativo, compatível com diversas plataformas e possui excelente documentação. Todo o desenvolvimento dos algoritmos utiliza recursos nativos da linguagem.

Para a manipulação e leitura dos arquivos DICOM, é utilizada a biblioteca de apoio que é citada na seção 5.2.1 e também é escrita em Java. O ambiente de programação escolhido é o Apache NetBeans IDE 12.0, que é totalmente compatível com a linguagem utilizada.

## 5.2. IMPLEMENTAÇÃO

O processo de segmentação é composto por diversas etapas, que são detalhadas no decorrer desta seção. Inicialmente, é apresentada uma visão macro do projeto, explicando a manipulação de arquivos DICOM e a orquestração da segmentação das fatias individuais. Em seguida, é abordado o algoritmo empregado para identificar os pulmões em cada uma das fatias do exame.

### 5.2.1. Manipulação de arquivos DICOM

Para realizar a segmentação automática dos pulmões em exames de Tomografia Computadorizada, é necessária a manipulação de arquivos DICOM para extrair os dados que serão processados. Foi utilizada a biblioteca *PixelMed Java DICOM Toolkit* para a leitura dos exames no formato DICOM e o acesso aos seus atributos.

A seção 4.2 explica o padrão DICOM e suas características. Arquivos nesse padrão armazenam diversas informações acerca do exame através das *element tags*. A Tabela 6 lista os atributos utilizados no protótipo.

**Tabela 6 - Atributos do padrão DICOM utilizados no protótipo**

Atributo	<i>Element tag</i>
<i>Columns</i>	(0028,0011)
<i>Rows</i>	(0028,0010)
<i>Slice Thickness</i>	(0018,0050)
<i>Rescale Intercept</i>	(0028,1052)
<i>Rescale Slope</i>	(0028,1053)
<i>Pixel Padding Value</i>	(0028,0120)
<i>Pixel Padding Range Limit</i>	(0028,0121)
<i>Pixel Data</i>	(7FE0,0010)

Fonte: elaborado pelo autor

Rönnau (2015) explica que os dados correspondentes às fatias são armazenados através do atributo *Pixel Data*, que pode ser representado como uma matriz bidimensional, denominada matriz de coeficientes de atenuação. Os elementos dessa matriz possuem valores no intervalo de 0 a 4096, representando o coeficiente de cada *voxel*. Valores na Escala *Hounsfield* estão compreendidos na faixa entre -1000 e 3000 (vide seção 4.2.1), portanto, percebe-se que os elementos da matriz não estão representando valores em HU. É necessário, portanto, realizar uma conversão entre os coeficientes armazenados no arquivo DICOM e os valores em HU que serão utilizados no processamento. A função que define essa conversão é  $vr = (va * rs) + ri$ , onde  $vr$  representa o valor resultante em HU,  $va$  é o valor armazenado,  $rs$  é o *rescale slope* e  $ri$  é o *rescale intercept*.

Existem também informações relacionadas ao preenchimento das imagens. Os atributos *Pixel Padding Value* e *Pixel Padding Range Limit* indicam se a imagem foi preenchida com um valor específico para torná-la retangular e qual é a faixa de valores válidos para esse propósito. Esses dados são utilizados caso o exame tenha sido gerado por um equipamento que gera imagens circulares. Os valores dos *voxels* preenchidos após a aquisição das imagens devem estar no intervalo compreendido entre o *Pixel Padding Value* e o *Pixel Padding Range Limit*, inclusive. Por simplificação, todos os preenchimentos do protótipo são realizados utilizando o valor de *Pixel Padding Value*.

Após a leitura e conversão dos dados para a matriz com valores em HU, o fluxo de execução do protótipo é direcionado para a orquestração do processamento, que tem como resultado final os pulmões segmentados em todas as fatias do exame.

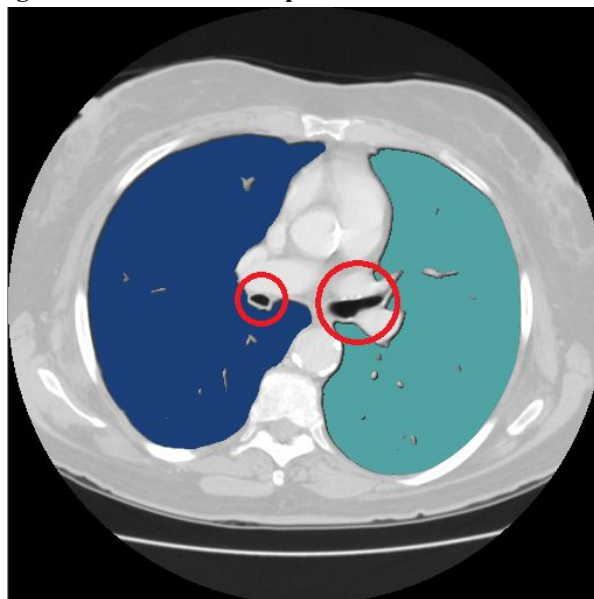
### 5.2.2. Orquestração do processamento

O primeiro passo do algoritmo é identificar a fatia inicial. Foi definido, de forma empírica, que a primeira fatia a ser processada é a que está localizada na posição que representa 60% das fatias do exame. Se, por exemplo, o exame contiver 100 fatias, a fatia inicial é a de número 60. Nos testes realizados, essa é uma posição que apresenta boa definição dos pulmões nos exames das bases de dados utilizadas.

Então é chamada a rotina de segmentação dos pulmões dessa fatia. É importante salientar que esse é o único processamento sobre uma fatia que não leva em consideração suas fatias vizinhas. Pelo fato de não haver referência de outras fatias, nesse momento são

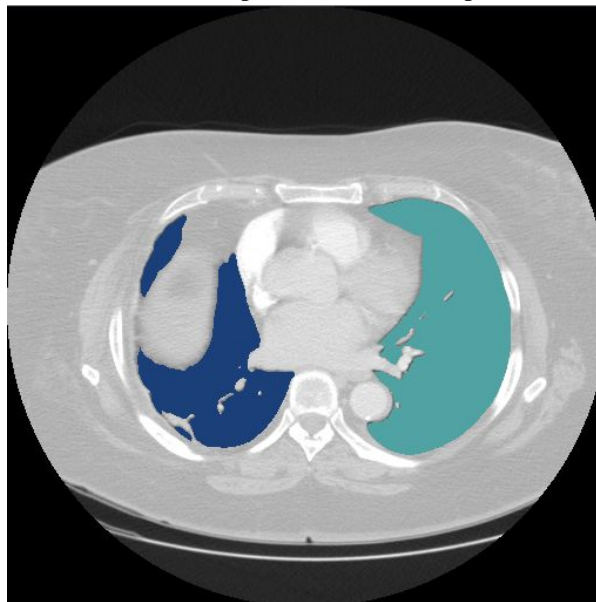
identificadas as duas maiores figuras da imagem como pulmão esquerdo e pulmão direito. Regiões menores, como vasos respiratórios e segmentos dos pulmões, ainda não são encontrados. A Figura 16 apresenta um exemplo desse processo. Em azul é representado o pulmão esquerdo, em verde é representado o pulmão direito e os círculos vermelhos indicam as regiões que compõem os pulmões que ainda não foram identificadas.

**Figura 16 – Resultado do processamento da fatia inicial**



Fonte: elaborado pelo autor

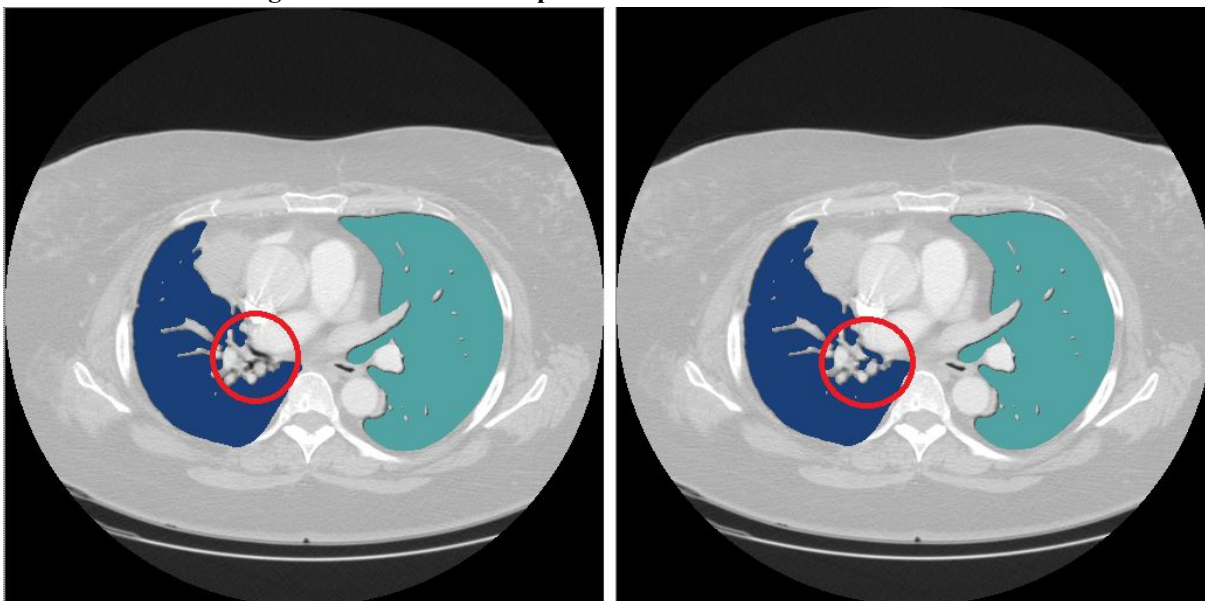
A próxima etapa aplica o procedimento sobre cada uma das fatias restantes do exame, partindo da fatia inicial. É feito um processamento percorrendo as fatias para cima e outro percorrendo para baixo, ambos partindo da fatia inicial. Cada uma das chamadas passa a fatia a ser processada e a fatia recém processada, que serve como referência para identificar a sobreposição dos objetos encontrados. Dessa forma, é possível que o algoritmo encontre mais que um objeto que pertença a determinado pulmão. A Figura 17 apresenta um exemplo em que foram identificados três segmentos do pulmão esquerdo.

**Figura 17 – Resultado do processamento completo de uma fatia**

Fonte: elaborado pelo autor

Após o primeiro processamento completo do exame, já são identificados vários objetos nas imagens pertencentes aos pulmões. Porém, como a sobreposição das figuras é montada durante o processamento das fatias, há casos em que ainda não foram identificadas todas as regiões de interesse. Para resolver essa situação, são feitos dois processamentos adicionais. O primeiro percorre todas as fatias do exame, da primeira até a última, passando como referência a fatia imediatamente superior. O segundo percorre todas as fatias do exame, da última até a primeira, passando como referência a fatia imediatamente inferior. Através de testes foi percebido que essa configuração apresentou os melhores resultados na base de dados utilizada. A Figura 18 demonstra o cenário que essa etapa visa resolver. À esquerda é exibida uma imagem que não passou por esse processamento adicional e à direita a imagem completamente segmentada.

Figura 18 – Resultado do processamento adicional sobre a fatia



Fonte: elaborado pelo autor

A Figura 19 apresenta o código-fonte do algoritmo explicado na seção atual. Os parâmetros passados para o método *segmentSlice* representam o índice da fatia a processar e o índice da fatia de referência. Quando o segundo parâmetro possui valor igual a -1, indica para o método fazer o processamento sem utilizar fatia de referência.

Figura 19 – Código-fonte da orquestração do processamento

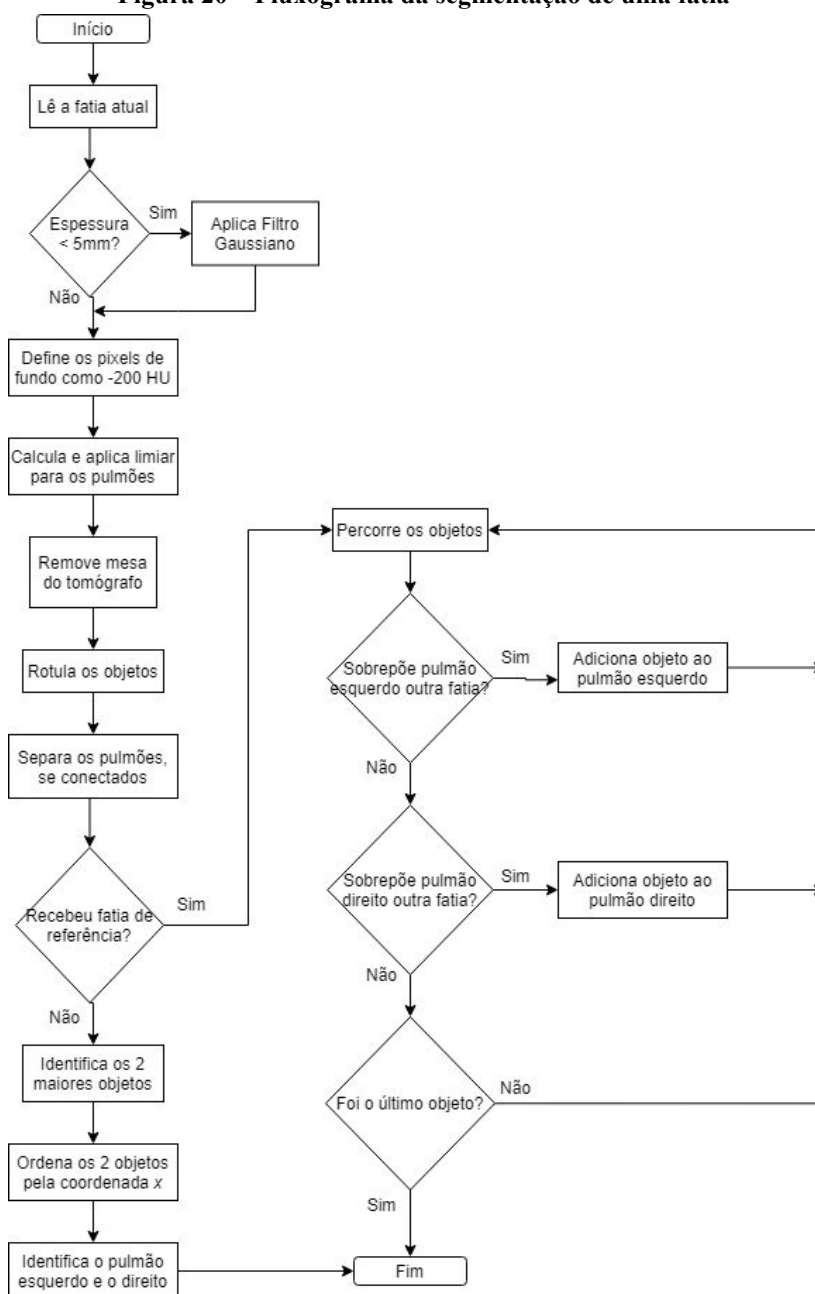
```
/**
 * Segmenta os pulmões do exame
 */
public void segmenta() {
    // Identifica a fatia inicial
    int centralSlice = (int) (exam.getNumberOfSlices() * 0.6);
    // Segmenta a fatia inicial, sem considerar fatias vizinhas
    segmentSlice(centralSlice, -1);
    // Segmenta as demais fatias passando as fatias vizinhas como referência
    for (int indiceFatia = centralSlice + 1; indiceFatia < exam.getNumberOfSlices(); indiceFatia++) {
        segmentSlice(indiceFatia, indiceFatia - 1);
    }
    for (int indiceFatia = centralSlice - 1; indiceFatia >= 0; indiceFatia--) {
        segmentSlice(indiceFatia, indiceFatia + 1);
    }
    // Segunda passada, para capturar objetos que antes não faziam intersecção mas agora fazem
    // Processa de cima para baixo
    for (int indiceFatia = 1; indiceFatia < exam.getNumberOfSlices(); indiceFatia++) {
        segmentSlice(indiceFatia, indiceFatia - 1);
    }
    // Terceira passada, para capturar objetos que antes não faziam intersecção mas agora fazem
    // Processa de baixo para cima
    for (int indiceFatia = exam.getNumberOfSlices() - 2; indiceFatia >= 0; indiceFatia--) {
        segmentSlice(indiceFatia, indiceFatia + 1);
    }
}
```

Fonte: elaborado pelo autor

### 5.2.3. Segmentação dos pulmões

O processo de segmentação dos pulmões, que é aplicado a cada uma das fatias do exame, é composto por duas etapas. A primeira é focada na identificação das regiões de interesse, ou seja, em encontrar objetos da imagem candidatos a pertencerem aos pulmões. A segunda utiliza os objetos extraídos na primeira e verifica a intersecção com as fatias adjacentes para confirmar seu pertencimento às regiões pulmonares. A etapa de verificação da sobreposição não é executada ao processar a fatia inicial, pois ainda não há fatia de referência. O fluxograma do processamento é apresentado na Figura 20.

Figura 20 – Fluxograma da segmentação de uma fatia



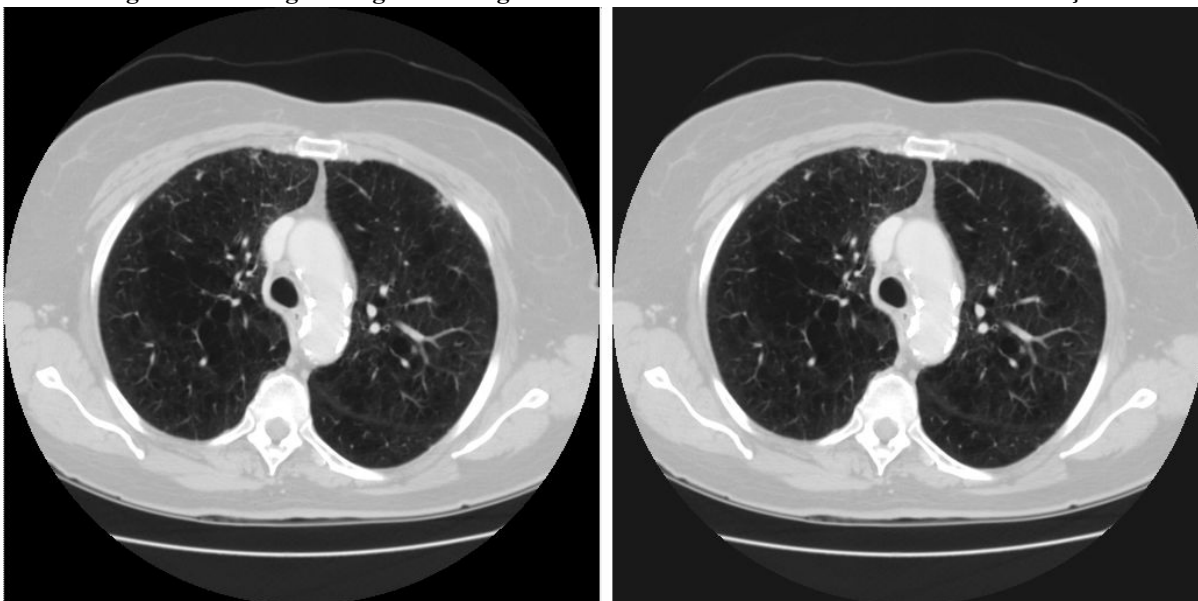
Fonte: elaborado pelo autor

### 5.2.3.1. Conversão de valores

A primeira transformação aplicada a cada uma das fatias é a conversão de valores fora de uma faixa válida para um valor conhecido de *background*. Isso é necessário, pois diversos equipamentos de Tomografia Computadorizada possuem formato circular e podem gerar imagens circulares. Como a matriz de coeficientes de atenuação é retangular, os espaços não preenchidos nas bordas são substituídos por valores determinados pelos atributos *Pixel Padding Value* e *Pixel Padding Value Range Limit*.

Esse processo separa o fundo da imagem e os objetos do exame. Para tal, cada *voxel* é avaliado e alterado caso esteja fora da faixa de valores em HU válidos. Os *voxels* com valores superiores a 4000 HU são substituídos pelo valor -1000 HU, que é o coeficiente de atenuação do ar. A Figura 21 demonstra o resultado desse processamento.

Figura 21 – Imagem original e imagem sem valores inválidos de coeficiente de atenuação



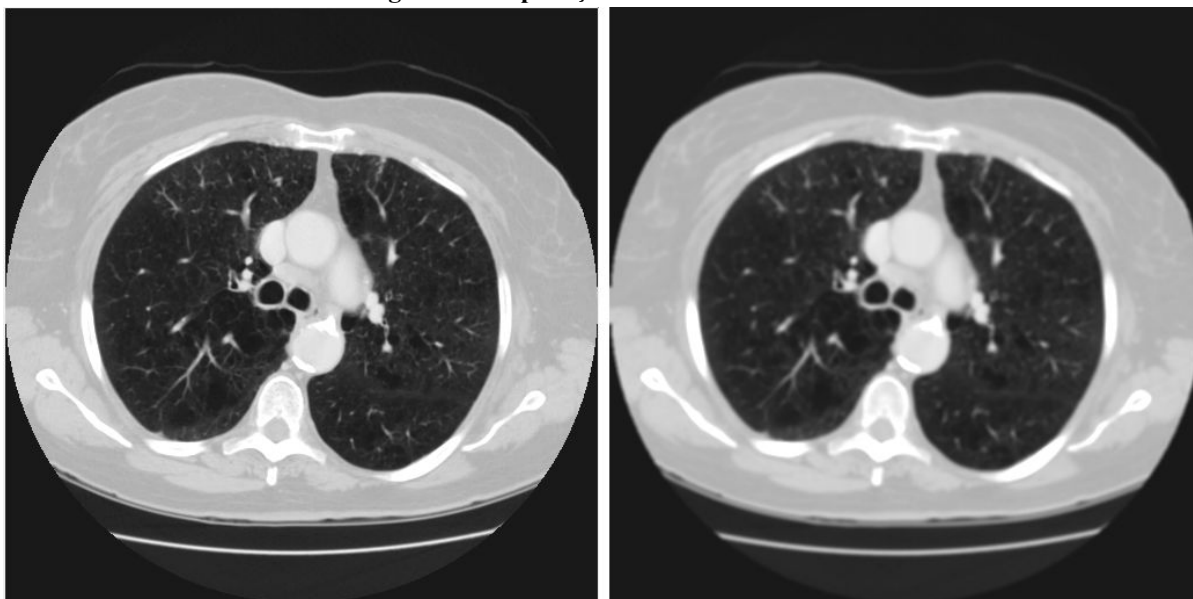
Fonte: elaborado pelo autor

### 5.2.3.2. Aplicação do Filtro Gaussiano

Conforme explica Leader *et al.* (2003), se a espessura da fatia do exame for menor que 5mm, deve ser aplicado um Filtro Gaussiano com *kernel* de uma dimensão, tamanho 5 e desvio padrão igual a 1,76. Esse procedimento tem como objetivo eliminar eventuais ruídos e suavizar a imagem (vide seção 2.1.4.4). A Figura 22 apresenta o resultado da aplicação do Filtro Gaussiano.



Figura 22 – Aplicação do Filtro Gaussiano



Fonte: elaborado pelo autor

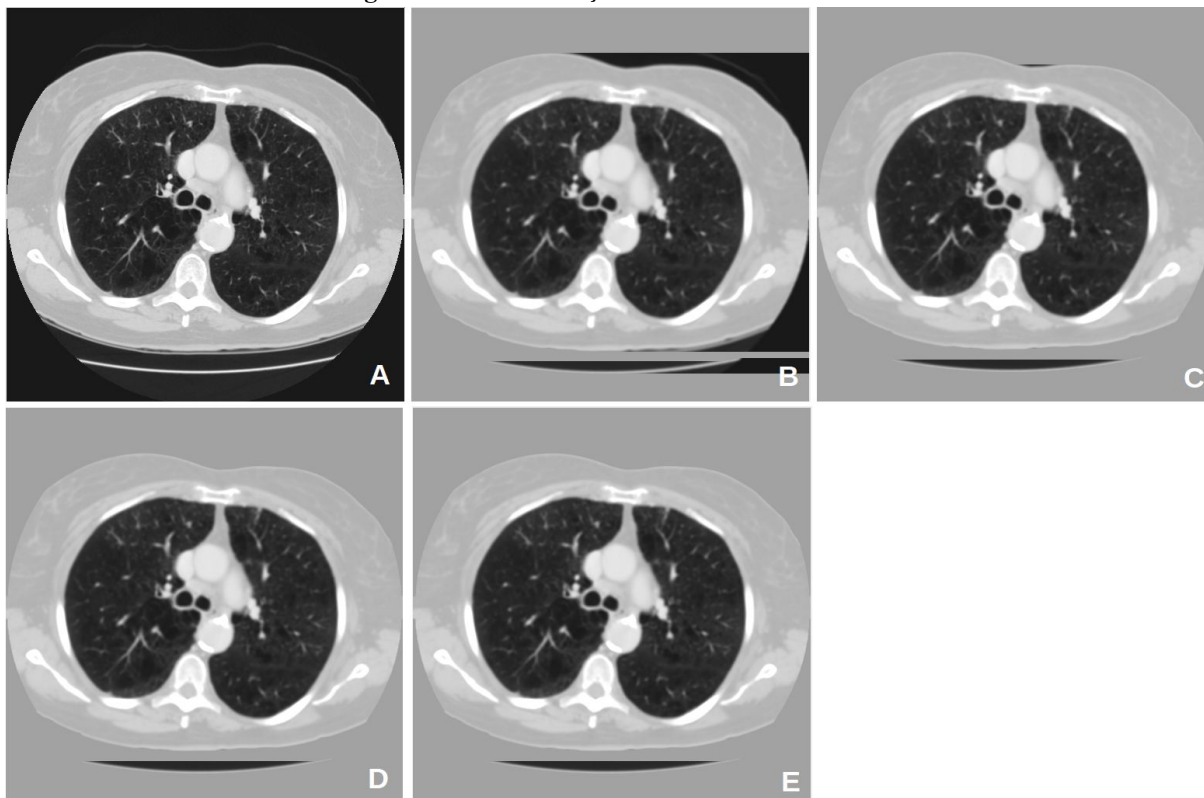
### 5.2.3.3. Identificação do contorno do tórax

Conforme explicado por Rönnau (2015), o coeficiente de atenuação do ar é bastante semelhante nas regiões internas e externas ao corpo humano. Isso faz com que o coeficiente que representa os pulmões seja muito parecido com o coeficiente do fundo da imagem. Parte-se do pressuposto de que não há ar nas regiões do tórax e, portanto, todo *voxel* com valor que representa o ar pertence a uma região candidata a ser pulmão ou ao exterior do corpo.

Para descartar os *voxels* correspondentes ao ar na região externa ao corpo, são feitas quatro varreduras na imagem. Cada varredura é feita em uma direção diferente, até atingir a parede do tórax, que possui um contorno bem definido. Após essa etapa, tudo que resta na fatia é o corpo do paciente e *voxels* de fundo.

A Figura 23 ilustra a execução desse algoritmo. A imagem A representa a imagem antes do processamento. As imagens B, C, D e E demonstram a varredura da esquerda para a direita, da direita para a esquerda, de cima para baixo e de baixo para cima, respectivamente. A cada iteração, se o *voxel* possui valor menor que -200 HU, seu valor é alterado para -200 HU. Se for encontrado um *voxel* com valor maior que -200 HU, o processamento segue para a próxima linha ou coluna.

Figura 23 – Identificação do contorno do tórax

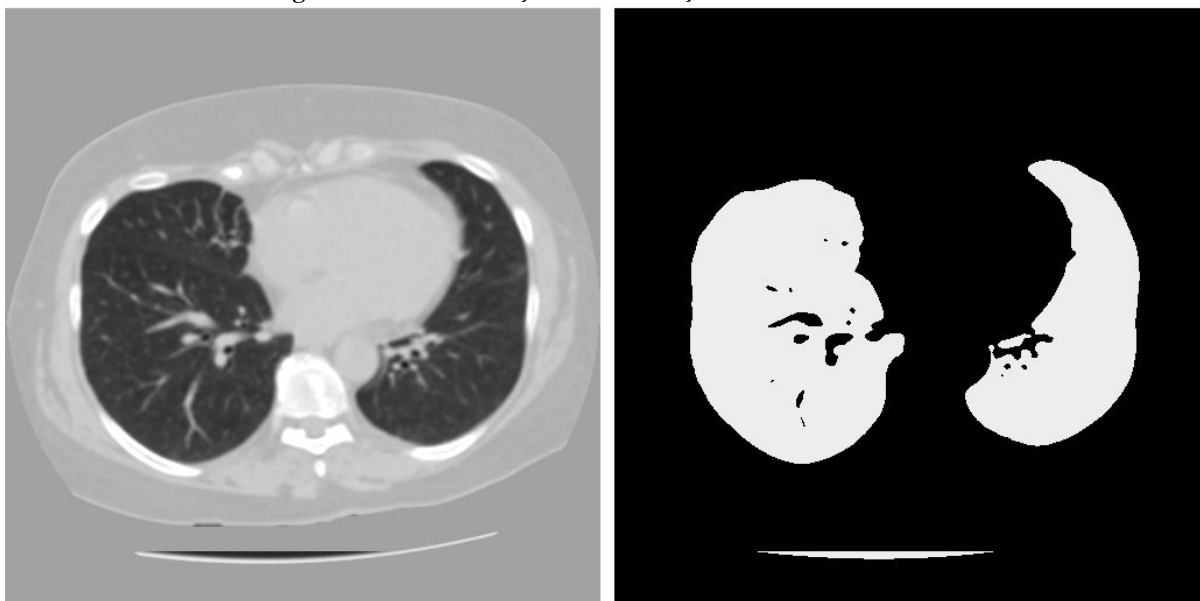


Fonte: elaborado pelo autor

#### 5.2.3.4. Definição e aplicação do limiar

A limiarização (explicada na seção 2.1.2) tem como objetivo destacar as regiões de interesse. O limiar é definido com base no histograma da imagem. As regiões mais escuras são candidatas a pertencerem aos pulmões, enquanto as regiões mais claras pertencem ao tórax do paciente.

O histograma gerado abrange valores entre o coeficiente de atenuação do ar, -1000 HU, e o valor de fundo da imagem, -200 HU. Seguindo a proposta de Leader *et al.* (2003), o limiar escolhido é o valor que tem o menor número de ocorrências no histograma gerado. A Figura 24 apresenta o processo de limiarização aplicado a uma imagem. O limiar, definido dinamicamente, foi de -311 HU. Os *voxels* que se encontram acima desse valor são considerados como o fundo da imagem. Os demais, são considerados como regiões de interesse.

**Figura 24 – Limiarização com definição dinâmica do limiar**

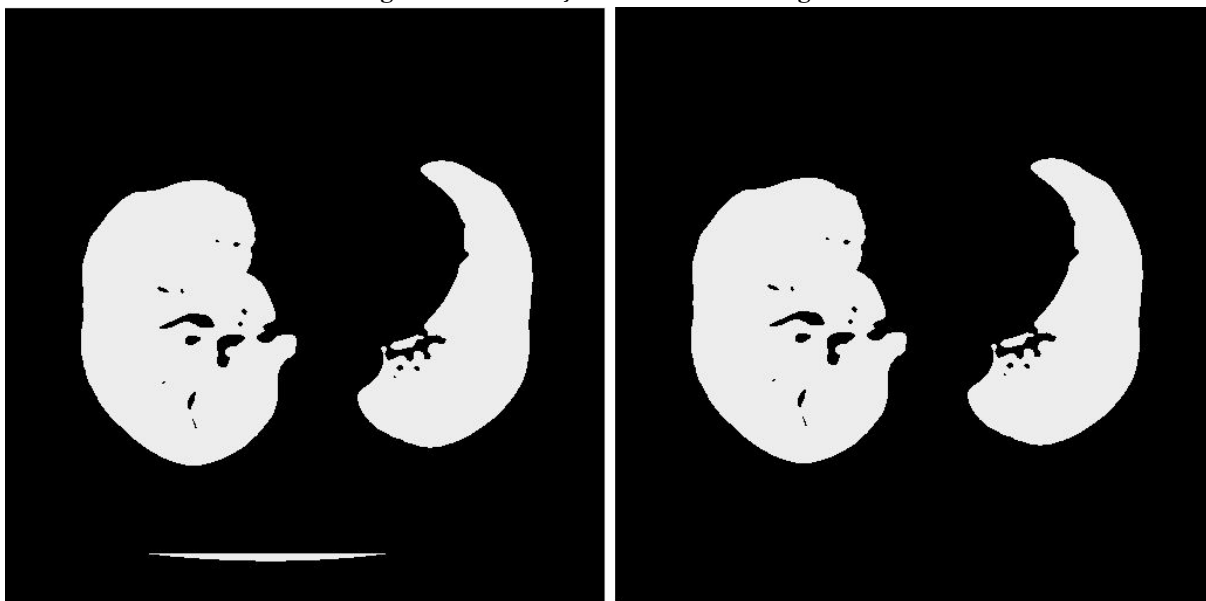
Fonte: elaborado pelo autor

#### **5.2.3.5. Remoção do tomógrafo**

O resultado do processo de limiarização, demonstrado na Figura 24, apresenta diversos objetos, incluindo elementos indesejados. O que mais se destaca é a representação da mesa do tomógrafo, que possui grande quantidade de *voxels* e pode atrapalhar o algoritmo de identificação dos pulmões.

É utilizado um algoritmo proposto por Rønnau (2015) para remover a mesa do tomógrafo da imagem. O processo realiza uma varredura na imagem de cima para baixo, partindo da metade vertical. Quando é identificada uma linha completa composta somente por valores de coeficientes de atenuação correspondentes ao fundo, toda a região abaixo dessa linha é definida como fundo da imagem. O resultado dessa etapa é ilustrado na Figura 25.

Figura 25 – Remoção da mesa do tomógrafo



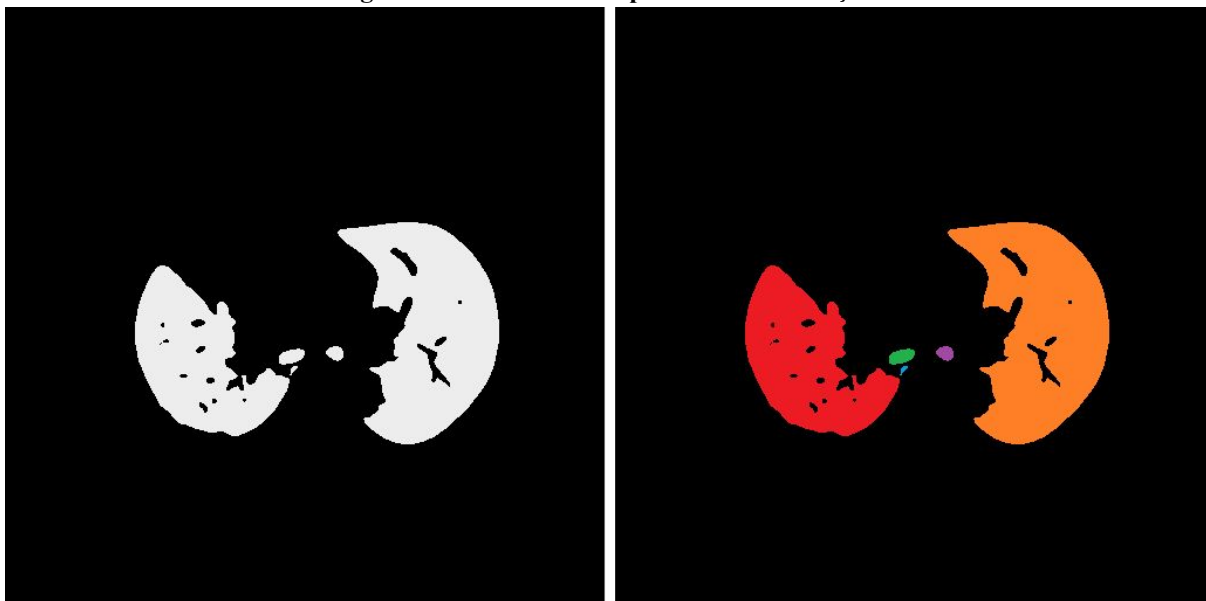
Fonte: elaborado pelo autor

#### 5.2.3.6. Rotulação

Após todos os tratamentos aplicados às imagens, a próxima etapa consiste em identificar os objetos resultantes. Para tal, é aplicado o processo de crescimento de regiões, explicado na seção 2.2.1. O resultado é cada um dos objetos identificados individualmente com rótulos diferentes.

O algoritmo percorre cada *voxel* da fatia e, ao identificar um que pertença às regiões de interesse, dispara um algoritmo de crescimento de regiões utilizando como semente o *voxel* em questão. Após a rotulação de um objeto específico, todos os *voxels* pertencentes a ele possuem o mesmo rótulo. A busca pelos *voxels* que são utilizados como semente considera somente os que ainda não pertencem a nenhum objeto, ou seja, não foram tratados em um processo de crescimento de regiões. Dessa forma, cada *voxel* é processado uma única vez. Na Figura 26 pode ser visto o resultado dessa etapa.

Figura 26 – Resultado do processo de rotulação

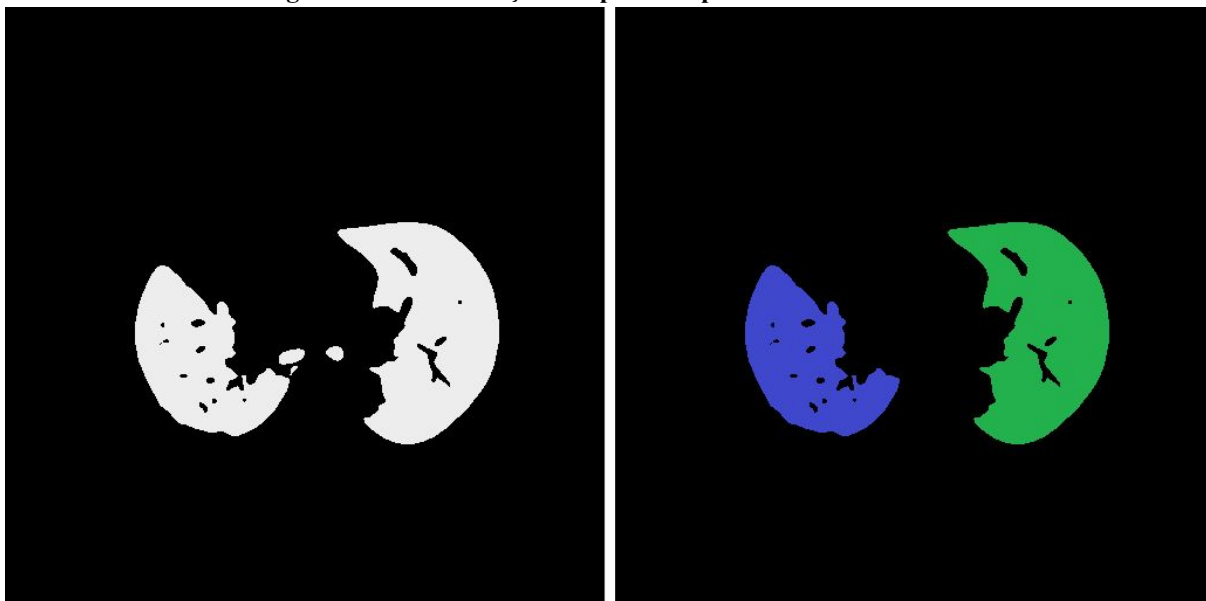


Fonte: elaborado pelo autor

#### 5.2.3.7. Identificação dos pulmões por dimensionamento

Ao processar a primeira fatia do exame, localizada na posição correspondente a 60% do número de fatias (vide seção 5.2.2), não é possível utilizar outras fatias como referência para identificar os pulmões. Portanto, é executado um algoritmo simplificado para identificar as duas maiores regiões da imagem, que normalmente são bem definidos nessa fatia, e classificá-los como pulmões. É importante salientar que, nesse primeiro momento, objetos menores pertencentes aos pulmões podem não ser identificados. Eles são tratados quando a fatia é novamente processada considerando as fatias vizinhas (técnica explicada na seção 5.2.3.8).

Todos os objetos da imagem são ordenados considerando como critério a área ocupada, do maior para o menor. Os dois primeiros são selecionados para representarem os dois pulmões. Entre esses dois objetos, aquele cuja coordenada  $x$  for a menor é eleito como pulmão esquerdo e o outro como pulmão direito. A coordenada  $x$  de um objeto é definida como a posição no eixo  $x$  do *voxel* mais à esquerda. A Figura 27 exhibe o resultado desse processamento. Pode-se perceber que há regiões que ainda não são identificadas.

**Figura 27 – Identificação dos pulmões por dimensionamento**

Fonte: elaborado pelo autor

#### **5.2.3.8. Identificação dos pulmões por sobreposição**

O maior diferencial desse trabalho é a capacidade de segmentar os pulmões em exames de Tomografia Computadorizada tendo como base, as regiões já identificadas em fatias adjacentes. Isso torna o processo mais preciso e com a capacidade de identificar os pulmões em sua totalidade.

Cada fatia passa pelos processos explicados nas seções 5.2.3.1 a 5.2.3.6. A partir dos objetos rotulados da fatia em processamento e dos objetos identificados na fatia vizinha, o algoritmo verifica a sobreposição das figuras. Dessa forma, se houver mais que um objeto que se sobrepõem com o pulmão esquerdo da outra fatia, todos são classificados como pulmão esquerdo, por exemplo. O mesmo conceito se aplica ao pulmão direito. Assim, é possível capturar fragmentos desses órgãos presentes nas imagens que, devido à sua pequena área, seriam desconsiderados.

Na Figura 28 é apresentado o código-fonte desta etapa. O algoritmo percorre simultaneamente todos os *voxels* de ambas fatias, incrementando os contadores de *voxels* de cada objeto e de *voxels* que se intersectam entre as fatias. Ao final, se o número de *voxels* com intersecção corresponderem a, no mínimo, 30% do número de *voxels* do objeto avaliado ou do objeto utilizado como referência, o objeto avaliado é classificado como pulmão. Esse percentual foi definido experimentalmente. Ao utilizar percentuais menores, ocorreram casos

em que os objetos foram classificados erroneamente, ou seja, houveram falsos-positivos. Por outro lado, ao experimentar percentuais maiores, houveram ocorrências de falsos-negativos, deixando de classificar objetos válidos. Para a base de dados utilizada, os melhores resultados foram obtidos através de uma sobreposição de, ao menos, 30%.

**Figura 28 – Código-fonte que identifica a sobreposição de 2 objetos**

```

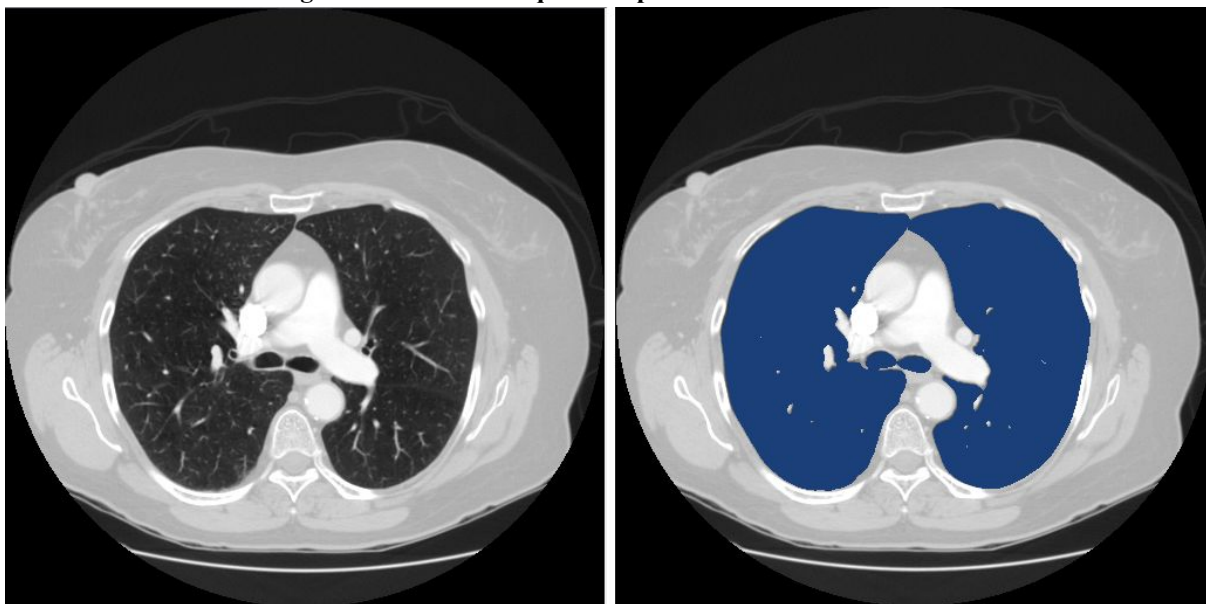
/**
 * Verifica se há intersecção entre os objetos de duas imagens binárias
 */
private boolean hasIntersection(BinaryImage binaryImage1, BinaryImage binaryImage2) {
    if (binaryImage1 == null || binaryImage2 == null) {
        return false;
    }
    // Inicializa contadores de voxels
    double voxelsIntersection = 0;
    double voxelsBinaryImage1 = 0;
    double voxelsBinaryImage2 = 0;
    // Obtém largura da menor imagem
    int width = binaryImage1.getWidth();
    if (binaryImage2.getWidth() < width) {
        width = binaryImage2.getWidth();
    }
    // Obtém altura da menor imagem
    int height = binaryImage1.getHeight();
    if (binaryImage2.getHeight() < height) {
        height = binaryImage2.getHeight();
    }
    // Percorre os voxels das imagens
    for (int x = 0; x < width; x++) {
        for (int y = 0; y < height; y++) {
            // Totaliza os voxels da primeira imagem
            if (binaryImage1.get(x, y)) {
                voxelsBinaryImage1++;
            }
            // Totaliza os voxels da segunda imagem
            if (binaryImage2.get(x, y)) {
                voxelsBinaryImage2++;
            }
            // Totaliza os voxels que há intersecção entre as imagens
            if (binaryImage1.get(x, y) && binaryImage2.get(x, y)) {
                voxelsIntersection++;
            }
        }
    }
    // Verifica se há intersecção de, pelo menos, 30% dos voxels em alguma das imagens
    return voxelsIntersection / voxelsBinaryImage1 > 0.3 || voxelsIntersection / voxelsBinaryImage2 > 0.3;
}

```

Fonte: elaborado pelo autor

### 5.2.3.9. Verificação de conectividade dos pulmões

Em alguns exames, devido ao fato dos pulmões estarem separados por uma distância muito pequena, o algoritmo de rotulação pode identificá-los como uma única região. Isso causa a segmentação equivocada da imagem. A Figura 29 mostra um exemplo desse cenário.

**Figura 29 – Pulmão esquerdo e pulmão direito conectados**

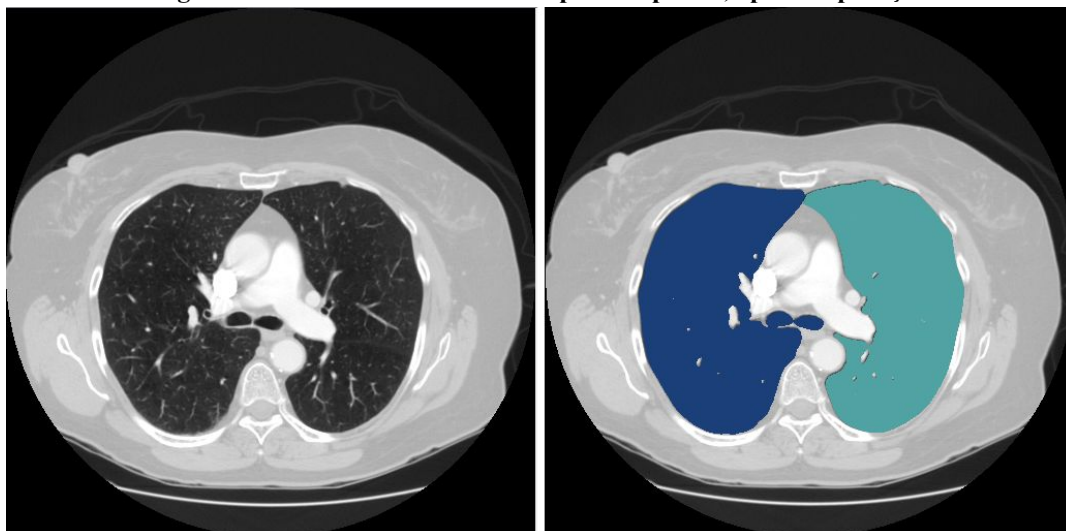
Fonte: elaborado pelo autor

Para identificar esse cenário, é verificada a concentração de *voxels* em cada metade horizontal da imagem. Se pelo menos 30% do objeto estiver na metade esquerda e pelo menos 30% estiver na metade direita, o algoritmo determina que os pulmões estão conectados. Ao identificar um objeto com essas características, é feita a confirmação através da fatia de referência. Para isso, é utilizado o algoritmo que identifica os pulmões por sobreposição, explicado na seção 5.2.3.8. A particularidade desse processo é que ele verifica a sobreposição tanto com o pulmão esquerdo quanto com o pulmão direito da outra fatia. Se isso se confirmar, os pulmões são considerados conectados.

O próximo passo é identificar se a união está na parte superior ou inferior da imagem. A Figura 30 demonstra um caso em que a união ocorre na parte superior. Caso os pulmões estejam unidos na parte inferior, é feito um espelhamento vertical na imagem antes de chamar a função de separação dos pulmões. Após o procedimento, a imagem é novamente espelhada verticalmente. Isso é necessário pois o algoritmo de separação dos pulmões está preparado para realizar a operação somente na região superior.



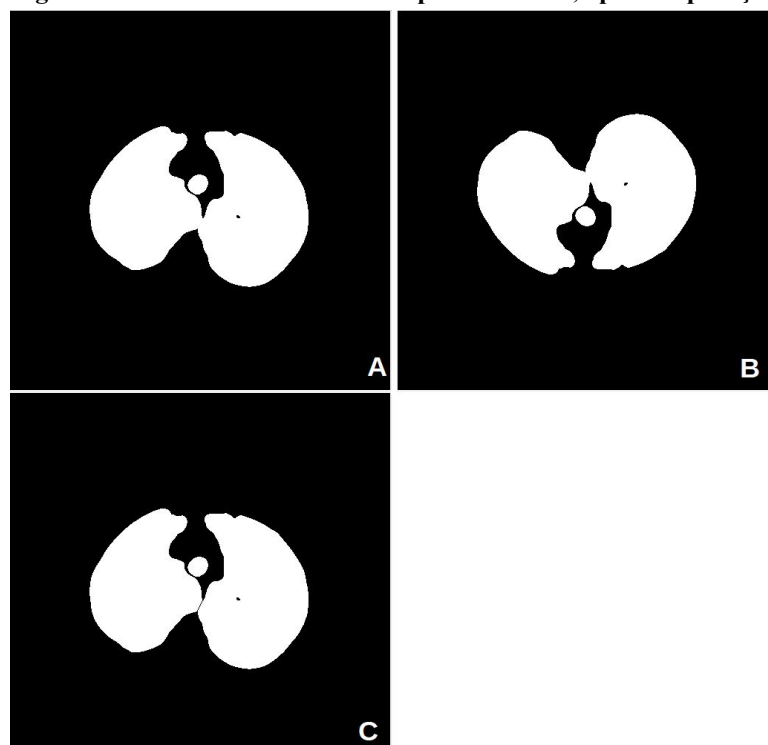
**Figura 30 – Pulmões conectados na parte superior, após a separação**



Fonte: elaborado pelo autor

Na Figura 31, pode-se ver um exemplo com pulmões unidos na região inferior e que foi espelhado verticalmente. A imagem A representa os pulmões antes da separação e a imagem B exibe o resultado do espelhamento vertical na figura. A imagem C apresenta o resultado final do processo, com os pulmões separados e invertidos novamente.

**Figura 31 – Pulmões conectados na parte inferior, após a separação**



Fonte: elaborado pelo autor

A identificação da região de conectividade é feita através de um algoritmo que percorre as colunas da imagem da esquerda para a direita, contemplando um intervalo entre 20 posições à esquerda do centro da imagem e 20 posições à direita, valor obtido empiricamente através de testes. Os testes práticos na base de dados utilizada para estudos demonstraram que a janela de 40 *voxels* no centro da imagem apresenta a maior assertividade nessa busca. Para cada coluna, são percorridos os *voxels*, de cima para baixo, até encontrar a borda dos pulmões. Se o *voxel* encontrado estiver abaixo da metade vertical da imagem, os pulmões estão conectados na parte inferior. Se o processamento chegar ao fim e não encontrar nenhum *voxel* de borda abaixo da metade vertical da imagem, significa que eles estão conectados na região superior e a imagem não precisa ser espelhada verticalmente.

O processo de separação dos pulmões consiste em identificar duas coordenadas  $[x1,y1]$  e  $[x2,y2]$  para traçar uma reta com um *voxel* de espessura e valor correspondente ao fundo da imagem. Essas coordenadas devem, obrigatoriamente, indicar dois pontos em bordas oposta para que a separação tenha sucesso.

São percorridas as colunas da imagem, iniciando 20 posições à esquerda do centro da imagem e encerrando 20 posições à direita. Primeiramente é identificada a posição  $y$  da borda superior do pulmão na coluna inicial. A rotina entra em um laço que percorre os *pixels* da borda superior, no intervalo definido. Para cada *voxel* da borda superior, é chamada uma rotina que identifica a menor distância com a borda inferior. A Figura 32 representa o código-fonte, escrito na linguagem Java, que realiza o procedimento explicado.

Figura 32 – Código-fonte que separa os pulmões conectados

```

// Identifica se a união dos pulmões está na região inferior da imagem
boolean uniaoRegiaoInferior = isUniaoRegiaoInferior(labeledMatrix);
// Inverte a imagem verticalmente, caso necessário
if (uniaoRegiaoInferior) {
    inverteImagem(labeledMatrix);
}
// Identifica a linha inicial com base na coluna inicial
int colunaCentral = image.getWidth() / 2;
int colunaInicial = colunaCentral - JANELA;
int colunaFinal = colunaCentral + JANELA;
int linhaInicial = -1;
for (int y = 0; y < image.getHeight(); y++) {
    if (labeledMatrix.get(colunaInicial, y)) {
        linhaInicial = y;
        break;
    }
}
// Percorre a borda superior do objeto, buscando a menor distância com a borda inferior
int xAnterior = colunaInicial;
int yAnterior = linhaInicial;
for(;;) {
    if (labeledMatrix.get(xAnterior + 1, yAnterior - 1)) {
        xAnterior++;
        yAnterior--;
    } else if (labeledMatrix.get(xAnterior + 1, yAnterior)) {
        xAnterior++;
    } else if (labeledMatrix.get(xAnterior + 1, yAnterior + 1)) {
        xAnterior++;
        yAnterior++;
    } else if (labeledMatrix.get(xAnterior, yAnterior + 1)) {
        yAnterior++;
    } else if (labeledMatrix.get(xAnterior - 1, yAnterior + 1)) {
        xAnterior--;
        yAnterior++;
    } else if (labeledMatrix.get(xAnterior - 1, yAnterior)) {
        xAnterior--;
    } else if (labeledMatrix.get(xAnterior - 1, yAnterior - 1)) {
        xAnterior--;
        yAnterior--;
    } else if (labeledMatrix.get(xAnterior, yAnterior - 1)) {
        yAnterior--;
    } else {
        break;
    }
}
if (xAnterior > colunaFinal) {
    break;
}
buscaMenorDistanciaBordaInferior(xAnterior, yAnterior, labeledMatrix);

```

Fonte: elaborado pelo autor

O método que busca a menor distância com a borda inferior realiza um algoritmo para percorrer a borda inferior muito semelhante ao algoritmo utilizado para percorrer a borda superior. Para cada *voxel* da borda inferior é calculada a distância em relação ao *voxel* avaliado na borda superior. A distância é calculada através da hipotenusa de um triângulo retângulo entre as coordenadas  $[x1,y1]$  e  $[x2,y2]$ . Na Figura 33, é demonstrado o código-fonte dessa função. Toda vez que é identificada uma distância menor, são armazenadas as coordenadas da borda superior e da borda inferior. Ao final do processamento, as posições

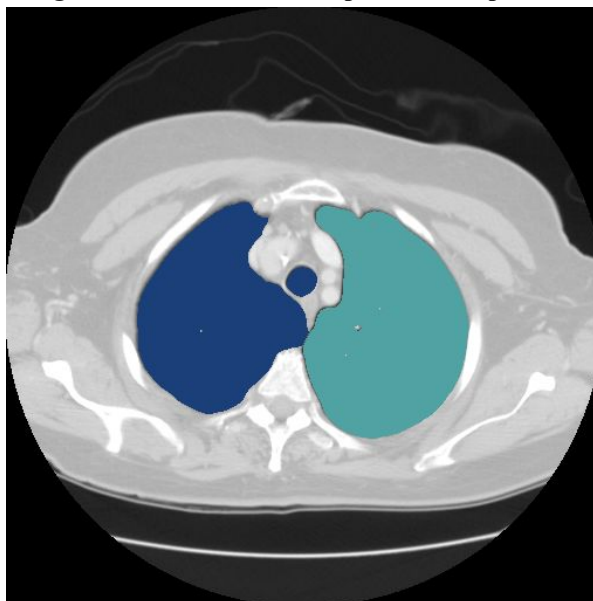
$[x1,y1]$  e  $[x2,y2]$  indicam os dois *voxels* mais próximos entre as bordas. O algoritmo de separação traça uma reta entre esse dois pontos, desunindo os pulmões. A Figura 34 exemplifica o resultado dessa técnica.

**Figura 33 – Código-fonte que busca a menor distância para a borda inferior**

```
// Busca a menor distância com a borda inferior à partir de um ponto na borda superior
private void buscaMenorDistanciaBordaInferior(int xReference, int yReference, BinaryImage labeledMatrixTmp) {
    int colunaCentral = xReference;
    int colunaInicial = colunaCentral - JANELA;
    int colunaFinal = colunaCentral + JANELA;
    int linhaInicial = -1;
    // Busca a linha da borda inferior na coluna inicial
    buscaLinhaInicial:
    for (int y1 = 0; y1 < image.getHeight(); y1++) {
        if (labeledMatrixTmp.get(colunaInicial, y1)) {
            for (int y2 = y1 + 1; y2 < image.getHeight(); y2++) {
                if (!labeledMatrixTmp.get(colunaInicial, y2)) {
                    linhaInicial = y2 - 1;
                    break buscaLinhaInicial;
                }
            }
        }
    }
    // Percorre a borda inferior do objeto, buscando a menor distância com um ponto na borda superior
    int xAnterior = colunaInicial;
    int yAnterior = linhaInicial;
    BinaryImage labeledMatrixCopy = new BinaryImage(labeledMatrixTmp);
    for(;;) {
        if (labeledMatrixCopy.get(xAnterior + 1, yAnterior + 1)) {
            xAnterior++;
            yAnterior++;
        } else if (labeledMatrixCopy.get(xAnterior + 1, yAnterior)) {
            xAnterior++;
        } else if (labeledMatrixCopy.get(xAnterior + 1, yAnterior - 1)) {
            xAnterior++;
            yAnterior--;
        } else if (labeledMatrixCopy.get(xAnterior, yAnterior - 1)) {
            yAnterior--;
        } else if (labeledMatrixCopy.get(xAnterior - 1, yAnterior - 1)) {
            xAnterior--;
            yAnterior--;
        } else if (labeledMatrixCopy.get(xAnterior - 1, yAnterior)) {
            xAnterior--;
        } else if (labeledMatrixCopy.get(xAnterior - 1, yAnterior + 1)) {
            xAnterior--;
            yAnterior++;
        } else if (labeledMatrixCopy.get(xAnterior, yAnterior + 1)) {
            yAnterior++;
        } else {
            break;
        }
    }
    if (xAnterior > colunaFinal) {
        break;
    }
    labeledMatrixCopy.set(xAnterior, yAnterior, false);
    double distanciaEntrePontos = calculaDistanciaEntrePontos(xReference, yReference, xAnterior, yAnterior);
    if (distanciaEntrePontos < menorDistancia) {
        menorDistancia = distanciaEntrePontos;
        xInferiorMenorDistancia = xAnterior;
        yInferiorMenorDistancia = yAnterior;
        xSuperiorMenorDistancia = xReference;
        ySuperiorMenorDistancia = yReference;
    }
}
}
```

Fonte: elaborado pelo autor

**Figura 34 – Resultado da separação dos pulmões**



Fonte: elaborado pelo autor

O resultado desse processo são os dois pulmões, que inicialmente estavam conectados, separados. Após essa etapa, é aplicado o algoritmo descrito na seção 5.2.3.8 para a correta classificação dos objetos.

### 5.3. VALIDAÇÃO DOS RESULTADOS

A construção do protótipo foi elaborada com base nas referências e desenvolvida através de experimentação. Essa seção demonstra a evolução do protótipo no decorrer do trabalho, bem como os resultados parciais e finais obtidos.

Para a validação dos resultados, foi utilizada a mesma base de dados de Rönnau (2015). Dessa forma, é possível comparar os resultados e verificar o êxito do projeto. A base de dados é composta por cinco exames completos e é um subconjunto dos dados utilizados para testes (vide seção 4.3). Os exames utilizados são apresentados na Tabela 7.

**Tabela 7 - Exames utilizados na validação do protótipo**

Exames	Quantidade de imagens
Exame01	93
Exame02	94
Exame03	77
Exame04	119
Exame05	186
Total	569

Fonte: elaborado pelo autor

A validação foi realizada de forma visual utilizando o método quantitativo. A segmentação de cada fatia dos exames foi verificada e considerada como correta ou incorreta seguindo alguns critérios, que são listados abaixo.

- Todas as figuras pertencentes ao pulmão esquerdo, incluindo vias aéreas, devem ter sido classificadas corretamente;
- Todas as figuras pertencentes ao pulmão direito, incluindo vias aéreas, devem ter sido classificadas corretamente;
- Caso os pulmões estejam unidos, a rotina de separação deve ter realizado o processo corretamente.

O desenvolvimento do protótipo foi dividido em fases. Na fase 1, as fatias eram processadas de forma totalmente independente. Dessa forma, não havia a busca de objetos por sobreposição. A fase 2 agregou a utilização de fatias de referência. A partir da segmentação da fatia inicial, conforme descrito na seção 5.2.2, o processamento seguia em direção às fatias superiores e às inferiores considerando a fatia anterior como base para a busca de objetos por sobreposição. Constatou-se, então, que em alguns casos ainda faltavam ser classificados alguns objetos, motivando a fase 3 do desenvolvimento, que é a fase final do projeto. Foi adicionada uma etapa no processo que, a partir do resultado da fase 2, realiza mais duas varreduras nas fatias do exame. A primeira percorre todas as fatias do exame, da primeira até a última, passando como referência a fatia imediatamente superior. A segunda percorre todas as fatias do exame, da última até a primeira, passando como referência a fatia imediatamente inferior. A Tabela 8 apresenta os resultados obtidos em cada fase do desenvolvimento do protótipo.

**Tabela 8 - Resultados parciais de cada fase**

Exames	Fase 1		Fase 2		Fase 3	
	Acertos	Erros	Acertos	Erros	Acertos	Erros
Exame01	48 (52%)	45 (48%)	89 (96%)	4 (4%)	89 (96%)	4 (4%)
Exame02	50 (53%)	44 (47%)	88 (94%)	6 (6%)	91 (97%)	3 (3%)
Exame03	35 (45%)	42 (55%)	68 (88%)	9 (12%)	77 (100%)	0 (0%)
Exame04	51 (43%)	68 (57%)	112 (94%)	7 (6%)	113 (95%)	6 (5%)
Exame05	85 (46%)	101 (54%)	166 (89%)	20 (11%)	184 (99%)	2 (1%)
Total	269 (47%)	300 (53%)	523 (92%)	46 (8%)	554 (97%)	15 (3%)

Fonte: elaborado pelo autor

Os resultados finais do protótipo foram consolidados e comparados com os obtidos por Rönnau (2015). É importante salientar que Rönnau (2015) utiliza critérios diferentes para a verificação. Em sua validação, são consideradas corretas as fatias que tiveram a maior região de cada pulmão segmentada corretamente, ignorando as regiões menores que são separadas por descontinuidades. Para fins de comparação, são utilizados os resultados do protótipo deste trabalho considerando os critérios explicados acima, os resultados obtidos por Rönnau (2015) segundo seus critérios e os resultados do protótipo de Rönnau (2015) com base nos mesmos critérios deste trabalho. A Tabela 9 exhibe esses dados.

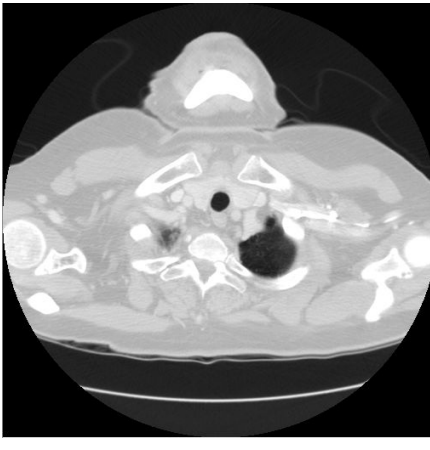
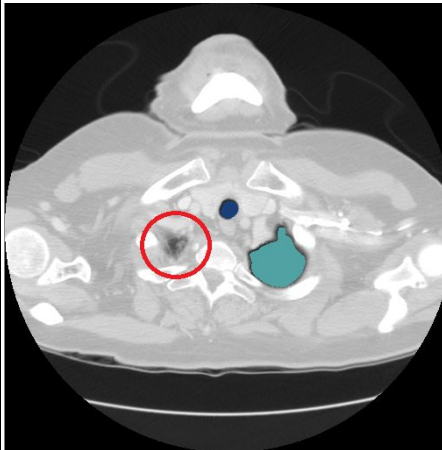
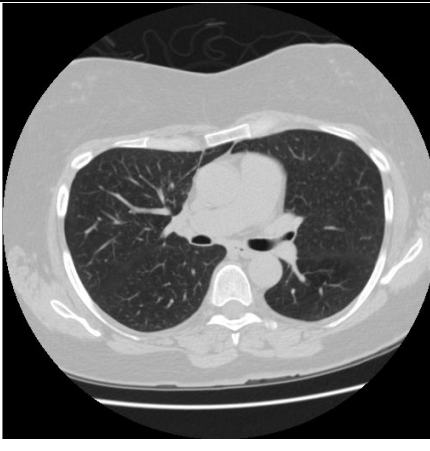
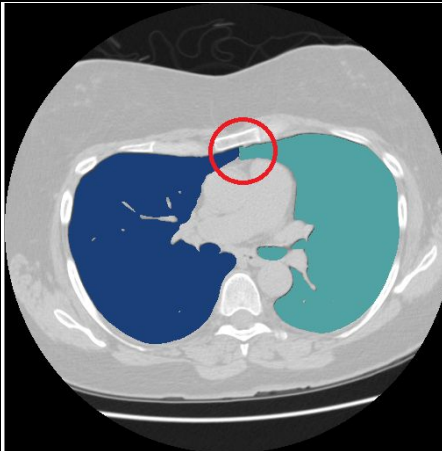
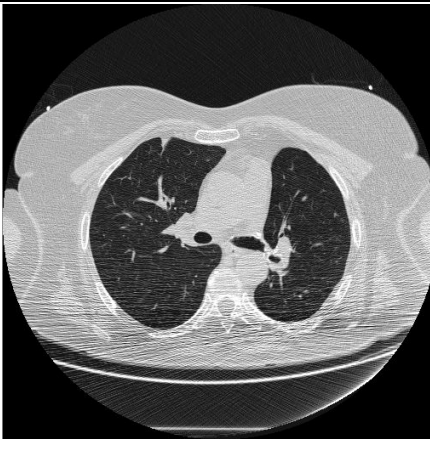
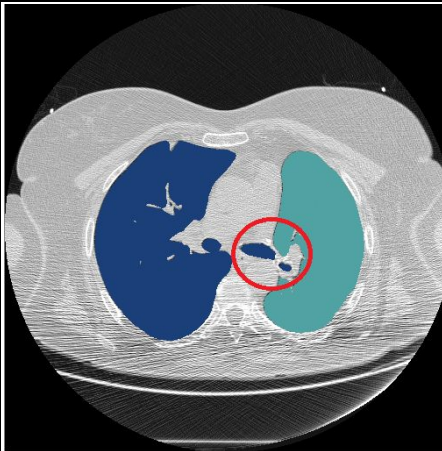
**Tabela 9 - Resultados comparados com Rönnau (2015)**

Exames	Protótipo com critérios do Autor		Rönnau (2015) com critérios de Rönnau (2015)		Rönnau (2015) com critérios do Autor	
	Acertos	Erros	Acertos	Erros	Acertos	Erros
Exame01	89 (96%)	4 (4%)	79 (85%)	14 (15%)	48 (52%)	45 (48%)
Exame02	91 (97%)	3 (3%)	86 (91%)	8 (9%)	50 (53%)	44 (47%)
Exame03	77 (100%)	0 (0%)	76 (99%)	1 (1%)	35 (45%)	42 (55%)
Exame04	113 (95%)	6 (5%)	113 (95%)	6 (5%)	51 (43%)	68 (57%)
Exame05	184 (99%)	2 (1%)	183 (98%)	3 (2%)	85 (46%)	101 (54%)
Total	554 (97%)	15 (3%)	537 (94%)	32 (6%)	269 (47%)	300 (53%)

Fonte: elaborado pelo autor

Em geral, os erros aconteceram porque o Filtro Gaussiano (vide seção 2.1.4.4), que é extremamente importante para o processo, elimina algumas figuras muito pequenas e une os pulmões quando eles estão muito próximos. A Tabela 10 mostra esses casos, exemplificando os problemas encontrados.

Tabela 10 - Causas dos erros

Exame	Fatia	Imagem original	Imagem segmentada
Exame01	1		
Causa		O Filtro Gaussiano eliminou parte do pulmão esquerdo.	
Exame04	55		
Causa		O Filtro Gaussiano juntou os pulmões. Então o algoritmo realizou a separação considerando a menor distância entre as bordas. A separação entre os pulmões deveria ser em diagonal.	
Exame05	82		
Causa		Alguns objetos do pulmão direito foram classificados como pulmão esquerdo. Esses objetos se sobrepõem com o pulmão esquerdo na fatia inferior ou superior.	

Fonte: elaborado pelo autor



#### 5.4. CONSIDERAÇÕES

Neste capítulo foram apresentados os resultados obtidos a partir do protótipo construído. Inicialmente, foram explicados os critérios utilizados para definir se uma fatia foi segmentada de forma correta ou incorreta. Foi apresentada a evolução do desenvolvimento através de suas três fases, bem como os resultados parciais de cada etapa. Por fim, com o objetivo de validar o trabalho, foi apresentada uma comparação entre os resultados do protótipo e o resultados alcançados por Rönnau (2015).

## 6. CONCLUSÃO

Este trabalho inicia contextualizando, no Capítulo 1, o tema de pesquisa e sua relevância nos dias atuais. É destacada a importância de um diagnóstico preciso em exames de Tomografia Computadorizada e como a computação, através do Processamento Digital de Imagens, pode contribuir para esse processo.

No Capítulo 2 é apresentada uma revisão bibliográfica acerca de técnicas de Processamento Digital de Imagens. Inicialmente, é tratado do pré-processamento de imagens, abordando transformações que podem ser feitas nas imagens para acentuar determinadas características e realçar regiões de interesse. Em seguida, são apresentadas técnicas de segmentação de imagens, que são empregadas na identificação de objetos para diferenciá-los do restante da imagem. Por fim, são tratadas das técnicas de classificação, que visam localizar os objetos em uma imagem, diferenciando-os entre si.

O Capítulo 3 expõe outros trabalhos produzidos na área de segmentação automática de órgãos em exames de Tomografia Computadorizada. O objetivo deste capítulo é explorar as técnicas utilizadas pelos autores, como elas foram aplicadas e quais os resultados obtidos.

Em seguida, o Capítulo 4 aprofunda o estudo acerca de exames médicos por imagem. Inicialmente, apresenta os exames de Tomografia Computadorizada. Então, detalha o padrão DICOM, que é um formato específico de arquivos gerados com base em exames de TC. Por fim, é descrita a base de dados que foi utilizada para testes e validações no desenvolvimento do protótipo deste trabalho.

O desenvolvimento e a validação do protótipo são abordados no Capítulo 5. Inicialmente, foram detalhados os passos realizados para preparar o ambiente de desenvolvimento e testes. Então são aprofundados os algoritmos e técnicas utilizados na implementação, detalhando cada uma das etapas do processo de segmentação dos exames de TC. Os resultados obtidos a partir da execução do protótipo são demonstrados e validados através da comparação com os resultados apresentados por Rönnau (2015).

O protótipo apresentado atende o propósito de segmentar, com alta acurácia, os pulmões em exames de TC. Destaca-se o fato de, a partir do protótipo, ser possível segmentar imagens em que os pulmões não sejam facilmente identificados, como nas extremidades, sendo esta, a principal contribuição científica.

Como trabalho futuro, pode-se buscar diminuir o percentual de falhas no processo de segmentação dos pulmões. Também pode ser proposta a segmentação de outros órgãos presentes nesse tipo de exame, como o coração ou traqueia. Além disso, viabiliza o

desenvolvimento de uma ferramenta de visualização em 3D das figuras anatômicas do tórax, que gere um modelo a partir dos dados obtidos com a segmentação dos exames.

## REFERÊNCIAS BIBLIOGRÁFICAS

- AUGUSTO, Fernando M. **Estudo da Qualidade da Imagem e Índices de Dose em Tomografia Computadorizada**. 2009. 61f. Dissertação (Mestrado em Engenharia Nuclear) - COOPE/UFRJ, Rio de Janeiro, 2009.
- ARAÚJO, R. T. S. **Detecção de manchas de óleo na superfície do mar em imagens de radar de abertura sintética**. 2004. Dissertação (Mestrado) – Universidade Federal do Ceará, Fortaleza, CE, 2004.
- BECKER, Jonata Daniel. **Desenvolvimento de uma Ferramenta Gamificada para o Ensino de Processamento Digital de Imagens**. 2018. 125 f. Trabalho de Conclusão de Curso (Monografia) – Curso de Ciência da Computação, Universidade Feevale, Novo Hamburgo, RS, 2018.
- CONCI, Aura; AZEVEDO, Eduardo; LETA, Fabiana R. **Computação Gráfica: Teoria e Prática**. Rio de Janeiro, RJ: Elsevier, 2008. 407 p.
- DAMBROS, Alexandre Luís. **Sistema de Reconhecimento de Placas de Veículos Automotores**. 2008. 76 f. Trabalho de Conclusão de Curso (Monografia) – Curso de Ciência da Computação, Universidade Feevale, Novo Hamburgo, RS, 2008.
- DRAKE, Richard L.; VOGL, A. Wayne; MITCHELL, Adam W. M. Gray. **Anatomia para estudantes**. 2. ed. Rio de Janeiro, RJ: Elsevier, 2010. 1103 p.
- FDA. **What is Computed Tomography?** Disponível em: <<http://www.fda.gov/Radiation-EmittingProducts/RadiationEmittingProductsandProcedures/MedicalImaging/MedicalX-Rays/ucm115318.htm>>. Acesso em: 29 mar. 2015.
- FELIX, J. H. S.; CORTEZ, P. C.; CAVALCANTE, T. S.; ALEXANDRIA, A. R.; HOLANDA, M. A. *Autoin: Method of automatic initialization of active contours applied to lungs in ct images*. **IEEE Latin America Transactions**, v. 10, n. 4, p. 1954 - 1960, 2012.
- FELIX, J. H. S.; CORTEZ, P. C.; HOLANDA, M. A.; COSTA, R. C. S.. *Automatic Segmentation and Measurement of the Lungs in healthy persons and in patients with Chronic Obstructive Pulmonary Disease in CT Images*. **IV Latin American Congress on Biomedical Engineering 2007, Bioengineering Solutions for Latin America Health**, p. 370 - 373, 2007.
- FILHO, Pedro P. Rebouças; CORTEZ, P. C.; FELIX, J. H. S.; CAVALCANTE, T. D. S.; HOLANDA, M. A. *Adaptive 2D crisp active contour model applied to lung segmentation in CT images of the thorax of healthy volunteers and patients with pulmonary emphysema*. **Revista Brasileira de Engenharia Biomédica**, v. 29, n. 4, p. 363-376, 2013.
- FLORES, Jéferson C. **Detecção Automática de Calcificação Arterial Coronariana em Tomografia Computadorizada**. 2015. 96 f. Trabalho de Conclusão de Curso (Monografia) – Curso de Ciência da Computação, Universidade Feevale, Novo Hamburgo, RS, 2015.
- GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento Digital de Imagens**. 3. ed. São Paulo, SP: Pearson, 2010. 624 p.

LEADER, Joseph K.; ZHENG, Bin; ROGERS, Robert M.; SCIURBA, Frank C.; PEREZ, Andrew; CHAPMAN, Brian E.; PATEL, Sanjay; FUHRMAN, Carl R.; GUR, David. "Automated Lung Segmentation in X-Ray Computed Tomography: Development and Evaluation of a Heuristic Threshold-Based Scheme". *Academic radiology*, v. 10, n. 11, p. 1224-1236, nov. 2003.

LI, Y., WONG; W. C. K. *An efficient and robust method to detect object centre*. *Pattern Recognition*, v. 30, n. 5, p. 659 - 671, 1997.

MOSSMANN, João Batista. **Estudo de Técnicas de Processamento de Imagens Aplicadas ao Apoio do Diagnóstico de Derrames Serosos de Etiologia Benigna e Maligna**. 2007. 118 f. Trabalho de Conclusão de Curso (Monografia) – Curso de Ciência da Computação, Universidade Feevale, Novo Hamburgo, RS, 2007.

NEMA. *The DICOM Standard 2015a*. Disponível em: < <http://medical.nema.org/standard.html>>. Acesso em: 02 jun. 2020.

NOGUEROL, Luana. **Sistema de Informação como Apoio ao Diagnóstico em Parasitologia**. 2008. 75 f. Trabalho de Conclusão de Curso (Monografia) – Curso de Sistemas de Informação, Universidade Feevale, Novo Hamburgo, RS, 2008.

NOVELLINE, Robert A. **Fundamentos de Radiologia de Squire**. 5. ed. Porto Alegre: Artmed, 1999. 623 p.

PEDRINI, Hélio; SCHWARTZ, William R. **Análise de Imagens Digitais: Princípio, Algoritmos e Aplicações**. 1. ed. São Paulo, SP: Thomson Learning, 2008. 508 p.

PIANYKH, Oleg S. *Digital Imaging and Communications in Medicine (DICOM)*. Berlim: Springer-Verlag, 2008. 383 p.

POHREN, Nicolás. **O Uso de Algoritmos Genéticos para Segmentação Automática de Imagens**. 2018. 84 f. Trabalho de Conclusão de Curso (Monografia) – Curso de Ciência da Computação, Universidade Feevale, Novo Hamburgo, RS, 2018.

RADIOPAEDIA PTY LTD. *Radiopaedia.org, the wiki-based collaborative Radiology resource*. Disponível em: < <http://https://radiopaedia.org/>>. Acesso em: 02 jun. 2020.

RODRIGUES, Murillo B.; MARINHO, Leandro B.; NÓBREGA, Raul Victor M.; SOUZA, João Wellington M.; FILHO, Pedro Pedrosa Rebouças. *Lung segmentation in chest computerized tomography images using the border following algorithm*. *International Conference on Intelligent Systems Design and Applications*, p. 539 - 548, 2016.

RÖNNAU, Rodrigo Freiberg. **Segmentação Automática de Órgãos em Imagens de Tomografia Computadorizada do Tórax**. 2015. 81 f. Trabalho de Conclusão de Curso (Monografia) – Curso de Ciência da Computação, Universidade Feevale, Novo Hamburgo, RS, 2015.

STIMAC, Gary K. **Introdução ao Diagnóstico por Imagens**. Rio de Janeiro, RJ: Guanabara Koogan, 1994. 467 p.

TCIA. *Welcome to The Cancer Imaging Archive - The Cancer Imaging Archive (TCIA)*. Disponível em: < <http://https://www.cancerimagingarchive.net/>>. Acesso em: 02 jun. 2020.