

Prof. Dr. Franklin César FloresDIN/UEM



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Tecnologia

Eduardo Manarin Daguano

**Algoritmo Genético para Segmentação de Imagens
utilizando Tamanho e Forma dos Objetos**

Limeira
2020

Eduardo Manarin Daguano

**Algoritmo Genético para Segmentação de Imagens utilizando
Tamanho e Forma dos Objetos**

Dissertação apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Tecnologia, na área de Sistemas de Informação e Comunicação.

Orientador: Prof. Dr. Ulisses Martins Dias

Este exemplar corresponde à versão final da Dissertação defendida por Eduardo Manarin Daguano e orientada pelo Prof. Dr. Ulisses Martins Dias.

Limeira
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Faculdade de Tecnologia
Felipe de Souza Bueno - CRB 8/8577

D135a Daguano, Eduardo Manarin, 1990-
Algoritmo genético para segmentação de imagens utilizando descritores de tamanho e forma dos objetos / Eduardo Manarin Daguano. – Limeira, SP : [s.n.], 2020.

Orientador: Ulisses Martins Dias.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Tecnologia.

1. Algoritmos genéticos. 2. Segmentação de imagens. 3. Bacias hidrográficas. I. Dias, Ulisses Martins, 1983-. II. Universidade Estadual de Campinas. Faculdade de Tecnologia. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Genetic algorithm for image segmentation using object size and shape

Palavras-chave em inglês:

Genetic algorithms

Image segmentation

Watersheds

Área de concentração: Tecnologia e Inovação

Titulação: Mestre em Tecnologia

Banca examinadora:

Ulisses Martins Dias [Orientador]

Franklin César Flores

Marco Antônio Garcia de Carvalho

Data de defesa: 02-10-2020

Programa de Pós-Graduação: Tecnologia

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-1963-405X>
- Currículo Lattes do autor: <http://lattes.cnpq.br/6075892200542673>

FOLHA DE APROVAÇÃO

Abaixo se apresentam os membros da comissão julgadora da sessão pública de defesa de disser-
tação para o Título de Mestre em Tecnologia na área de concentração Sistemas de Informação
e Comunicação, a que se submeteu o aluno Eduardo Manarin Daguano, em 02 de outubro de
2020 na Faculdade de Tecnologia – FT/UNICAMP, em Limeira/SP.

Prof. Dr. Ulisses Martins Dias
Presidente da Comissão Julgadora

Prof. Dr. Marco Antônio Garcia de Carvalho
FT/Unicamp

Prof. Dr. Franklin César Flores
DIN/UEM

Ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós Graduação da FT.

Agradecimentos

Agradeço primeiramente minha esposa, Alana Spessoto, por sempre ter apoiado minhas decisões e estar presente em todos os momentos, principalmente nos mais dificeis.

Agradeço a minha família, em especial minha mãe Marta Angélica Manarin, por sempre me incentivar a estudar e pelo suporte incondicional.

Agraço ao meu orientador Prof. Dr. Uliisses Martins Dias que acreditou na construção do trabalho e me auxiliou nos momentos mais dificeis da dissertação.

Agradeço ao Lauro Massaitsu Tachibana por possibilitar o início do meu mestrado.

Agradeço a todos os mestres e doutores que, quando precisei, estavam prontos para esclarecer minhas indagações, em especial, a Marco Antônio G. De Carvalho.

Agradeço aos meus amigos e colegas de trabalho em especial Rodolfo Blazizza Joaquim e Livia Pedon por me concederem férias em momentos críticos do projeto.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001

Resumo

A segmentação de imagens consiste em dividir a imagem em regiões com o intuito de simplificar a visualização e facilitar a identificação dos objetos. O processo de segmentação pode ser efetuado de várias formas distintas como, por exemplo: segmentação utilizando o formato dos objetos, intensidade no nível de cinza, histograma de cores, entre outras formas. Essa vasta gama de técnicas existe porque diferentes situações geram problemas específicos e, por esse motivo, os critérios para segmentar e identificar os objetos variam de acordo com a situação apresentada. Dentro do universo da segmentação, destacam-se as técnicas que utilizam níveis de cinza, que podem ser englobadas em duas categorias principais: (i) técnicas que utilizam a detecção de contornos e formatos e (ii) técnicas que utilizam estruturas hierárquicas. Neste trabalho, utilizamos a transformada *Watershed* hierárquica como técnica de segmentação e a utilização da *Árvore dos Lagos Críticos* como estrutura representativa. Adicionalmente, utilizamos como critérios de seleção o tamanho e a forma dos objetos. De modo mais específico, definimos uma função que é ajustada por meio de um algoritmo genético que otimiza os parâmetros livres da função. Três algoritmos foram desenvolvidos para se obter os resultados. O primeiro algoritmo, denominado de *Algen*, consiste em um algoritmo genético que aprimora os resultados no decorrer da segmentação. O segundo algoritmo, denominado de *Algat*, executa o processo de segmentação repetidas vezes, alterando os valores de entrada (dentro do intervalo permitido) semi-aleatoriamente. O terceiro algoritmo, denominado de *Almod*, é executado em conjunto com os algoritmos *Algen* e *Algat*, sendo que sua função é de classificar os resultados ao final de cada processamento. Os processos e algoritmos desenvolvidos neste trabalho tiveram como foco principal a segmentação de imagens de células, sendo que, todos os testes de otimização e adaptação do algoritmo genético foram realizados no referido tipo de imagem.

Palavras-chave: Algoritmo Genético, Segmentação de Imagens, *Watershed*, Árvore dos Lagos Críticos.

Abstract

Image segmentation consists of dividing the image into regions in order to simplify the visualization and the identification of objects. The segmentation process can be build in several different ways, for example: segmentation using the shape of the objects, intensity in the gray level, histogram of colors, etc. This wide range of techniques exists because different situations generate specific problems and, therefore, the criteria for segmenting and identifying objects vary according to the situation presented. Within the segmentation universe, the techniques that use gray levels can be included in two main categories: (i) techniques that use the detection of contours and shapes and (ii) techniques that use hierarchical structures. In this work, we propose the use of the hierarchical Watershed transform as segmentation technique and the use of Tree of Critical Lakes as a representative structure. Additionally, we use the size and shape of objects. In more detail, we work with a size range and adopt the elliptical curvature metric to produce the final result, which is obtained through the weighted average between the size score and the elliptic curvature score. We developed three algorithms to obtain better results. The first algorithm, called Algen, consists of a genetic algorithm that improves the results during the segmentation. The second algorithm, called Algal, performs the segmentation process repeatedly, changing the input values (within the allowed range) semi-randomly. The third algorithm, called Almod, is executed in conjunction with Algen and Algal algorithms. Almod main function is to classifier all results and it is executed at the end of each segmentation. The processes and algorithms developed in this work had as main focus the segmentation of cell images, and all tests of optimization and adaptation of the genetic algorithm were performed on the referred type of image.

Keywords: Genetic Algorithm, Image Segmentation, Watershed, Tree of Critical Lakes.

Lista de Figuras

1.1	Engenharia da imagem	11
1.2	Tipos de métodos para segmentação	12
2.1	Vizinhanças	17
2.2	Componentes Conexas	19
2.3	Representação Component Tree e Max-Tree	19
2.4	Representação da Transformada de Watershed	22
2.5	Cruzamento Cromossômico e Mutação	23
3.1	Diagramação	30
3.2	Processo de inundação	33
3.3	Representação da Árvore dos Lagos Críticos	34
3.4	Formatos e valores da curvatura elíptica	37
3.5	Módulo da diferença	40
4.1	Imagens de células <i>Hoechst 33342</i>	45
4.2	Segmentação manual de células <i>Hoechst 33342</i>	45
4.3	Células adenocarcinoma	46
4.4	Etapas da Segmentação	46
4.5	Resultado da Segmetacão - Lab. Murphy	47
4.6	Resultado da Segmentação - ATCC	47
4.7	Ilustração dos resultados de Medidas <i>F</i>	49
4.8	Métrica Módulo vs F-Score	50
4.9	Melhor vs Pior vs <i>Almod</i>	50
4.10	Desempenho do Algal	51
4.11	Desempenho do Algen	52
4.12	Performance <i>Algen</i> vs <i>Algal</i>	53
4.13	Volatilidade dos Resultados (Lab. Murphy)	55
4.14	Volatilidade da execução Algal com Algen (Lab. Murphy)	56
4.15	Volatilidade dos Resultados (ATCC)	56
4.16	Volatilidade do Algal com Algen (ATCC)	56
4.17	Dados Tabulados da Volatilidade	57
4.18	Comparação Algal, Algen e Algal com Algen (Murphy)	58
4.19	Comparação resultado das Técnicas (Murphy)	58
4.20	Comparação Algal, Algen e Algal com Algen (ATCC)	59
4.21	Comparação resultado das Técnicas (ATCC)	60

Sumário

1	Introdução	10
1.1	Motivação	13
1.2	Objetivos	14
1.3	Organização da Tese	14
2	Fundamentação Teórica	15
2.1	Conceitos Básicos	15
2.1.1	<i>Component Tree e Max-Tree</i>	18
2.1.2	Transformada de <i>Watershed</i>	20
2.1.3	Árvore dos Lagos Críticos	21
2.2	Algoritmo Genético	22
2.3	Revisão da Literatura	24
3	Metodologia Proposta	29
3.1	Diagrama dos processos	30
3.2	Geração da Árvore dos Lagos Críticos	30
3.3	Métricas para identificação dos objetos	35
3.4	Algoritmo Genético	37
3.5	Estabilidade dos Parâmetros de Entrada	43
4	Resultados	44
4.1	Base de Dados	44
4.2	Etapas e Resultados da Segmentação	46
4.3	Formato da Análise Comparativa	47
4.4	Avaliação da Métrica Almod	49
4.5	Resultados do Algen e Algalm	51
4.5.1	Análise por Instâncias do Laboratório Murphy	52
4.5.2	Estabilidade dos Resultados e Tempo de Execução	53
4.6	Comparação entre Métodos de Segmentação	54
4.6.1	Estudo da Volatilidade	54
4.6.2	Estudo dos Resultados da Medida <i>F</i>	57
5	Considerações Finais	61
	Referências bibliográficas	63

Capítulo 1

Introdução

Imagens possuem grande relevância no processo evolutivo humano, sendo utilizadas como apoio para comunicação e tomada de decisões. Com o avanço da tecnologia e a difusão das comunicações, a relevância das imagens aumentou significativamente, assim como a extensão de suas aplicações. Esse novo cenário aumenta a demanda pelo processamento de imagens e pela extração de informações a partir delas.

A segmentação de imagens é frequentemente utilizada em aplicações com o intuito de identificar regiões de interesse e simplificar o reconhecimento de objetos na imagem original. Devido à não existência de uma solução única para o problema da segmentação, as técnicas desenvolvidas são construídas para um cenário previamente conhecido.

Um dos objetivos da segmentação de imagens é simular as habilidades humanas de percepção e interpretação, de modo a solucionar problemas computacionais complexos. Conforme descrito por Pedrini e Schwartz (2008), existem dois níveis de abstração: o baixo nível e o alto nível. O baixo nível, representado pelo processamento digital, tem como finalidade melhorar a qualidade das imagens para facilitar a interpretação humana. O alto nível visa analisar as imagens por meio de suas características, tais como: tamanho, formato, contorno e textura, de modo a executar tarefas complexas, como segmentações de imagens, simplificações de objetos e identificação de áreas de interesse.

Zaitoun e Aqel (2015) determinam que a engenharia de imagens pode ser categorizada em três aspectos distintos, ilustrados na Figura 1.1: (i) processamento da imagem, (ii) análise da imagem e (iii) entendimento da imagem.

O processamento da imagem definido por Zaitoun e Aqel (2015) assemelha-se ao baixo nível definido por Pedrini e Schwartz (2008), pois realiza operações pixel a pixel, produzindo

uma versão modificada da imagem original de forma a realçar as características desejadas. As operações realizadas neste nível são de reconstrução (restauração, reconstrução e mosaico), transformação (alterações no contraste, redução dos ruídos, modificação no histograma, compressão das informações e rotação) e classificação (segmentação e classificação).

A análise da imagem destaca-se por realizar medições possibilitando a criação de novos conjuntos de imagens a partir do conjunto inicial, e, o entendimento da imagem, tem como princípio o estudo da natureza da imagem e dos objetos.

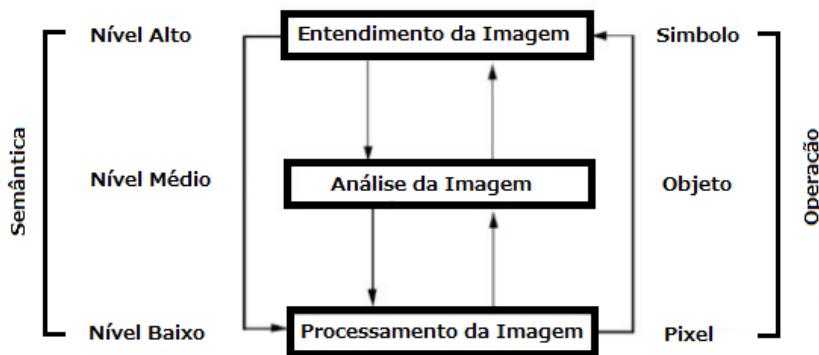


Figura 1.1: Engenharia da imagem: representação dos três níveis propostos por Zaitoun e Aqel (2015).

Zaitoun e Aqel (2015) definem também que os métodos que realizam segmentação de imagens podem ser divididos em duas categorias principais: *Segmentação baseada em Camadas* e *Segmentação baseada em Blocos*, como mostrado na Figura 1.2.

A *Segmentação baseada em camadas* não foi discutida no trabalho de Zaitoun e Aqel (2015), porém, foi classificada como aquela responsável por definir máscaras de formato, realizar ordenação em profundidade e avaliar as instâncias de segmentação.

A *Segmentação baseada em Blocos* foi classificada por Zaitoun e Aqel (2015) como responsável pela detecção dos objetos baseada nas características da imagem. Essas características podem ser em razão do nível de cor para criação de histogramas, informações a reperito dos pixels que indiquem bordas ou níveis de textura. A *Segmentação baseada em Blocos* pode ser sub-dividida em quatro técnicas principais, sendo que apenas as duas primeiras (*baseado em Regiões* e *baseado em Bordas*) foram abordadas em profundidade em seu trabalho. As quatro técnicas pertencentes à *Segmentação baseada em Blocos* são: (i) *Segmentação baseada em Blocos por Regiões*, (ii) *Segmentação baseada em Blocos por Bordas*, (iii) *Segmentação baseada em Blocos por Pixel* (não mencionada na Figura 1.2) e (iv) *Segmentação baseada em Blocos por Modelo* (não mencionada na Figura 1.2).

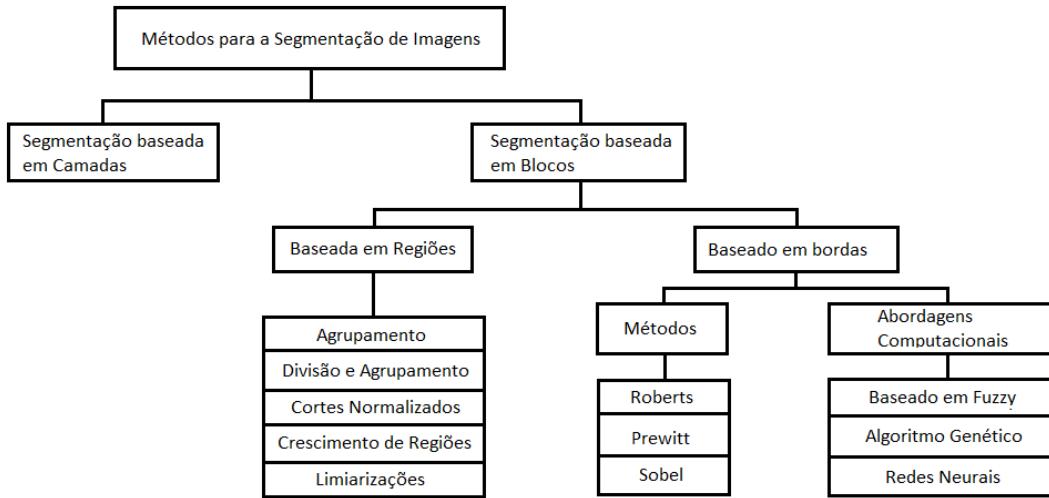


Figura 1.2: Métodos de Segmentação de Imagens definidos por Zaitoun e Aqel (2015).

As técnicas de segmentação sofrem constantemente com ruídos nas imagens originais. Referidos ruídos ocasionam erros de processamento durante a etapa de segmentação e, por esse motivo, aplica-se geralmente uma etapa de pré-processamento, a qual é responsável por amenizar as distorções causadas pelos ruídos e realçar as características consideradas mais relevantes na imagem.

Em nosso trabalho, utilizamos a segmentação por regiões, por bordas e por pixel, sendo todas categorizadas como segmentação por blocos segundo Zaitoun e Aqel (2015). Nossa técnica é fundamentada principalmente na divisão da imagem em regiões, sendo que a quantidade de regiões geradas varia de acordo com a quantidade de objetos que se enquadram nas características de seleção. O processo é finalizado quando os objetos de interesse são identificados e isolados.

Outras técnicas também realizam segmentações por blocos, contudo, em nosso trabalho propomos a possibilidade de manipular a estrutura hierárquica privilegiando ora o tamanho e ora a forma dos objetos por meio de parâmetros. Os parâmetros são otimizados por um algoritmo genético, responsável por encapsular todo o processo de segmentação.

De forma resumida, um algoritmo genético é uma meta-heurística amplamente empregada na Ciência da Computação para realizar uma busca sobre um conjunto de valores de modo a obter soluções aproximadas. Esses algoritmos usam técnicas inspiradas na biologia, como seleção natural, mutação, herança, reprodução, dentre outras. Uma revisão desses algoritmos será fornecida posteriormente para favorecer o entendimento do que foi produzido neste trabalho.

A utilização do algoritmo genético nos possibilita antever resultados e ajustar os parâmetros de acordo com a necessidade. Nos experimentos, constatamos uma melhora significativa nos resultados, no desvio padrão e na sensibilidade aos parâmetros iniciais. A nossa proposta de segmentação foi dividido em quatro etapas, sendo elas:

1. Na primeira etapa, realizamos o pré-processamento constituído de *gaussian blur*, *median blur*, erosão e dilatação em diferentes níveis de intensidade. A primeira etapa é concluída mediante a obtenção do gradiente morfológico a partir da redução dos ruídos e das possíveis distorções.
2. Na segunda etapa, computamos a transformada *Watershed* hierárquica e obtemos a Árvore dos Lagos Críticos, que passará a representar as informações de interesse em níveis hierárquicos, possibilitando a criação de partições.
3. A terceira etapa é responsável por manipular as informações de acordo com as características de seleção das regiões de interesse. Nesta etapa, utilizamos um algoritmo genético que recebe como entrada seis hiper-parâmetros que ora privilegiam o formato e ora privilegiam o tamanho.
4. A última etapa visa selecionar o melhor resultado na última geração do algoritmo genético. Nesta etapa, selecionamos os cinco melhores resultados para análise conjunta dos resultados.

1.1 Motivação

Existem muitas abordagens para realizar a segmentação de imagens, o que decorre do fato de problemas e cenários diferentes exigirem técnicas e critérios diferentes. Muitas abordagens para segmentação foram desenvolvidas e avaliadas em conjuntos de dados sem nenhuma semelhança com aqueles em que estão sendo aplicadas, fazendo com que os resultados obtidos tenham uma qualidade inferior àquela prometida pelos autores das técnicas. Em geral, as técnicas de segmentação fornecem poucas opções de adaptação, o que limita o uso em novos cenários. Uma técnica de segmentação que fornece parâmetros com semânticas claras e adaptáveis é uma opção importante e viável para reduzir a instabilidade nos resultados.

1.2 Objetivos

O objetivo desta dissertação é propor uma técnica de segmentação de imagem que forneça opções ao usuário para que este encontre segmentações adequadas às suas necessidades. Nossa abordagem permite a configuração de seis parâmetros relacionados ao tamanho e à forma dos objetos. Em caso de incerteza sobre a escolha dos parâmetros, o usuário pode configurar intervalos mais abrangentes para os parâmetros, de modo a permitir que grande parte do ajuste seja feito pelo algoritmo genético. Este algoritmo genético pode ainda explorar um espaço de busca fora dos intervalos configurados pelo usuário, de modo a permitir que o resultado final não seja tão sensível às configurações iniciais, possibilitando a correção de eventuais erros de configuração.

1.3 Organização da Tese

Os próximos capítulos desta dissertação são organizados da seguinte forma:

Capítulo 2: descreve a fundamentação teórica da área de segmentação de imagens. Este capítulo visa (i) trazer para o leitor maior clareza sobre os tópicos que serão apresentados e (ii) apresentar os trabalhos relacionados a esta dissertação.

Capítulo 3: apresenta a metodologia desenvolvida ao longo do trabalho. Este capítulo expõe as técnicas utilizadas e as modificações necessárias para adaptá-las ao nosso contexto. O capítulo é composto por três seções e apresenta respectivamente: (i) a estrutura de dados utilizada, (ii) a métrica para identificação dos objetos e (iii) os algoritmos implementados.

Capítulo 4: descreve os resultados obtidos utilizando o nosso processo de segmentação. Este capítulo apresenta (i) as características das bases utilizadas, (ii) a avaliação das métricas de identificação e classificação de resultado, (iii) a análise da volatilidade das técnicas apresentadas, (iv) a evolução dos resultados por meio do algoritmo genético e (v) a análise dos resultados gerados.

Capítulo 2

Fundamentação Teórica

Neste capítulo, apresentamos a Seção 2.1 com os conceitos básicos de segmentação de imagens e algumas das técnicas mais utilizadas. Na sequência, apresentamos a Seção 2.2 que aborda os principais conceitos sobre algoritmo genético e, por fim, na Seção 2.3 expomos os principais trabalhos relacionados a esta dissertação.

2.1 Conceitos Básicos

Apesar de imagens já serem onipresentes no cotidiano das pessoas, uma definição formal é necessária para facilitar o entendimento dos conceitos apresentados:

Definição 1 Um **pixel** é a menor unidade de representação de uma informação visual. Cada pixel p é associado a um vetor (p_1, p_2, \dots, p_c) , onde p_i representa a intensidade de um comprimento de onda i , $i \in \{1, 2, \dots, c\}$, para o pixel p .

Um modelo de representação muito usado é o RGB, que associa a cada pixel p um vetor (p_1, p_2, p_3) , onde p_1 representa a intensidade da cor vermelha, p_2 representa a intensidade da cor verde e p_3 representa a intensidade da cor azul. Em imagens de escala de cinza (*grayscale*), cada pixel p é associado a um vetor contendo apenas um escalar (p_1).

Definição 2 Uma **Imagem I** é uma matriz que possui dimensões $n \times m$, e cada célula $I_{i,j}$ da matriz corresponde a um pixel p associado a um vetor (p_1, p_2, \dots, p_k) .

As amostras utilizadas neste trabalho correspondem a imagens em nível de cinza, o que implica dizer que cada pixel p guarda a informação de apenas uma intensidade de comprimento

de onda $p = (p_1)$, havendo uma única amostra de um espaço de cores. Além disso, cada pixel é armazenado utilizando um byte, o que permite 256 intensidades possíveis variando entre o preto, como a menor intensidade, e o branco, como a maior intensidade.

Para facilitar a leitura do texto, utilizaremos o termo nível de cinza de um pixel para nos referir à intensidade p_1 e utilizaremos o termo pixel para nos referir ao posicionamento i, j de p na matriz \mathcal{I} . Utilizaremos a notação $\mathcal{I}_{i,j} = p_1$ para representar um pixel de modo a indicar a intensidade do pixel e reforçar o posicionamento deste na imagem.

Definição 3 A *limiarização* de uma imagem em nível de cinza é um procedimento que, dado um inteiro h , transforma a imagem original em uma imagem binária. Existem dois casos: o *limiar superior*, mostrado na Equação 2.1, e o *limiar inferior*, mostrado na Equação 2.2.

$$\mathcal{X}_h^{\geq}(\mathcal{I}_{i,j}) = \begin{cases} 1 & \text{se } \mathcal{I}_{i,j} \geq h \\ 0 & \text{caso contrário} \end{cases} \quad (2.1)$$

$$\mathcal{X}_h^{\leq}(\mathcal{I}_{i,j}) = \begin{cases} 1 & \text{se } \mathcal{I}_{i,j} < h \\ 0 & \text{caso contrário} \end{cases} \quad (2.2)$$

A técnica desenvolvida neste trabalho tem como saída uma imagem limiarizada que diferencia os objetos de interesse em relação ao *background*. A análise dos resultados consiste, então, em comparar o limiar obtido com um *ground-truth* gerado por especialistas.

As definições que seguem foram formuladas para facilitar a compreensão do algoritmo que computa a transformada *Watershed* e gera a árvore dos lagos críticos.

Definição 4 A *vizinhança* de um pixel $\mathcal{I}_{i,j}$ é definida como um conjunto de coordenadas da matriz \mathcal{I} . Duas maneiras muito comuns de definir vizinhanças são: a **Vizinhança-4** ($C4$) e a **Vizinhança-8** ($C8$), onde $C4(\mathcal{I}_{i,j}) = \{\mathcal{I}_{i-1,j}, \mathcal{I}_{i+1,j}, \mathcal{I}_{i,j-1}, \mathcal{I}_{i,j+1}\}$ e $C8(\mathcal{I}_{i,j}) = \{\mathcal{I}_{i-1,j}, \mathcal{I}_{i+1,j}, \mathcal{I}_{i,j-1}, \mathcal{I}_{i,j+1}, \mathcal{I}_{x-1,y-1}, \mathcal{I}_{x-1,y+1}, \mathcal{I}_{x+1,y-1}, \mathcal{I}_{x+1,y+1}\}$.

A Figura 2.1 ilustra as vizinhanças $C4$ e $C8$. As células com textura mostram a vizinhança da célula central em ambas as situações.

Definição 5 Considere dois pixels $\mathcal{I}_{x,y}$ e $\mathcal{I}_{s,t}$ em uma imagem \mathcal{I} . O **caminho** entre $\mathcal{I}_{x,y}$ e $\mathcal{I}_{s,t}$ é qualquer sequência de coordenadas distintas $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, tal que o pixel de

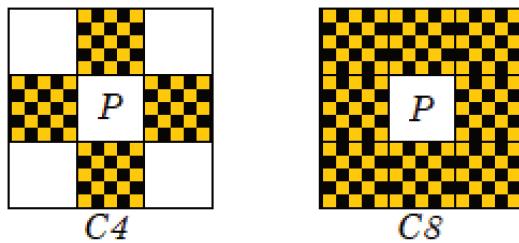


Figura 2.1: Tipos de vizinhanças. Em (a) apresentamos a vizinhança C_4 e em (b) apresentamos a vizinhança C_8 . Figura foi adaptada de Carvalho (2004)

coordenada (x_{i+1}, y_{i+1}) está na vizinhança do pixel de coordenada (x_i, y_i) , para todo $0 \leq i < n$, $(x, y) = (x_0, y_0)$ e $(s, t) = (x_n, y_n)$.

Definição 6 Seja S um subconjunto de pixels em uma imagem \mathcal{I} . Dois pixels $\mathcal{I}_{x,y}$ e $\mathcal{I}_{s,t}$ estão **conectados** em S se existe um caminho entre eles consistindo inteiramente de coordenadas em S .

Definição 7 Uma **componente conexa** C é um conjunto maximal de pixels tal que para todo par de pixel $\mathcal{I}_{x,y} \in C$ e $\mathcal{I}_{s,t} \in C$ existe um caminho em C entre $\mathcal{I}_{x,y}$ e $\mathcal{I}_{s,t}$.

Definição 8 Uma **zonas plana** é uma componente conexa cujos elementos possuem a mesma intensidade de nível de cinza.

Definição 9 Um **mínimo regional** de uma imagem \mathcal{I} é um conjunto maximal de pixels M tal que $\mathcal{I}_{i,j} < \mathcal{I}_{s,t}$, para todo $\mathcal{I}_{i,j} \in M$, $\mathcal{I}_{s,t} \notin M$ e $\mathcal{I}_{s,t}$ na vizinhança de $\mathcal{I}_{i,j}$. Em outras palavras, pixels em um mínimo regional M possuem apenas: (i) vizinhos em M , ou (ii) vizinhos de maior intensidade fora de M .

É possível perceber que mínimos regionais são zonas planas rodeadas por outras zonas planas de maior intensidade. Esse conceito é importante porque os algoritmos de agrupamentos normalmente iniciam com a busca por mínimos regionais.

Definição 10 Uma **partição** \mathcal{P} de uma imagem \mathcal{I} é um conjunto de regiões disjuntas e não vazias $\{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_n\}$ da imagem de modo que cada pixel $\mathcal{I}_{i,j}$ esteja em apenas uma dessas regiões: $\emptyset \notin \mathcal{P}$, $\cup R_i = \mathcal{I}$ e $R_i \cap R_j = \emptyset$ para todo $i \neq j$.

Definição 11 Uma sequência de partições $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n$ de uma imagem \mathcal{I} é uma **hierarquia** se, para todo $i < j$, cada região de \mathcal{P}_i é subconjunto de alguma região em \mathcal{P}_j .

A definição de hierarquia permite dizer que uma partição em um dado nível é obtida unindo regiões de uma partição em um nível imediatamente inferior. Os métodos clássicos para agrupamento hierárquico podem ser divididos em *bottom-up* e *top-down*. Os métodos *bottom-up* iniciam colocando cada objeto em uma região e, em seguida, escolhem o melhor par de regiões para unir segundo critérios pré-definidos até que todos os objetos estejam reunidos em uma só região. Os métodos *top-down* iniciam com todos os objetos em uma única região e subdividem essa região em duas novas, até que cada objeto forme uma região por si só.

A Seção 2.1.1 apresenta um método *top-down* capaz de gerar duas estruturas hierárquicas chamadas de *Component Tree* e *Max-Tree*, tais estruturas não foram utilizadas na metodologia proposta nesta dissertação, mas alguns conceitos que utilizamos foram derivados de técnicas inicialmente propostas para essas estruturas. A Seção 2.1.2 introduz o conceito de transformada de *Watershed*, implementada neste trabalho. A Seção 2.1.3 apresenta a estrutura hierárquica computada a partir da transformada de *Watershed*, conhecida como Árvore dos Lagos Críticos.

2.1.1 *Component Tree* e *Max-Tree*

Tanto a *Component Tree* quanto a *Max-tree* são construídas a partir de limiares aplicados a cada pixel $I_{i,j}$ da imagem de entrada. O resultado da aplicação de um limiar gera uma matriz binária que possui componentes conexas representadas por valores iguais a “1”. Quanto maior o valor do limiar aplicado, menor é a região das componentes conexas. Portanto, à medida que o valor do limiar aumenta, são criadas relações de inclusão entre as regiões derivadas em cada limiarização. A *Component Tree* e a *Max-tree* usam essas relações de inclusão para definir graus de parentesco.

A Figura 2.2 mostra diversos processos de limiarização e indica as componentes conexas geradas por meio de cores distintas. A imagem de entrada I é representada por uma matriz M de dimensões 4×4 . As matrizes M_t são formadas utilizando o valor t como limiar. Nesse caso, as células de coordenada (i,j) de cada matriz M_t recebem o valor “0” se $I_{i,j} < t$, caso contrário, recebem o valor “1”: $M_t = \mathcal{X}_t^{\geq}(I_{i,j})$

À medida que o limiar t aumenta, mais componentes conexas são geradas. Por exemplo, na Figura 2.2, M_1 possui apenas uma componente conexa chamada de CC_1 ; M_2 também possui apenas uma componente conexa CC_2 , contida inteiramente em CC_1 ; M_3 possui duas

componentes conexas CC_3 e CC_4 , ambas contidas em CC_2 ; M_4 possui duas componentes conexas CC_5 , contida em CC_3 , e CC_6 , contida em CC_4 ; M_5 possui duas componentes conexas CC_7 , contida em CC_5 , e CC_8 , contida em CC_6 .

$M_t =$	$M_1 =$ (≥ 1)	$M_2 =$ (≥ 2)	CC
$\begin{matrix} 1 & 5 & 3 & 3 \\ 1 & 5 & 5 & 3 \\ 1 & 1 & 2 & 2 \\ 2 & 4 & 4 & 5 \end{matrix}$	$\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{matrix}$	$\begin{matrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{matrix}$	CC_1 CC_2 CC_3 CC_4 CC_5 CC_6 CC_7 CC_8
$M_3 =$ (≥ 3)	$M_4 =$ (≥ 4)	$M_5 =$ (≥ 5)	
$\begin{matrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{matrix}$	$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{matrix}$	$\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$	

Fonte: elaboração

do autor.

Figura 2.2: Limiarização da matriz M e extração das componentes conexas. As formações em M_1 e M_2 resultam da aplicação do primeiro e do segundo limiar respectivamente. A formação em M_3 resulta da aplicação do terceiro limiar e como resultado produziu duas componentes conexas, as quais são: CC_3 e CC_4 . A formação em M_4 foi resultado do quarto limiar e gerou duas componentes conexas no total, sendo elas: CC_5 e CC_6 . A formação em M_5 resulta do último limiar que gerou novamente duas componentes conexas, sendo elas: CC_7 e CC_8 .

A relação de inclusão das componentes conexas permite gerar os nós da *Component Tree* e da *Max-Tree*. Os limiares dos níveis de cinza aplicados na Figura 2.2 estão refletidos no Eixo “y” da Figura 2.3. A componente conexa CC_1 inclui todas as demais e, por isso, é o nó raiz das árvores na Figura 2.3.

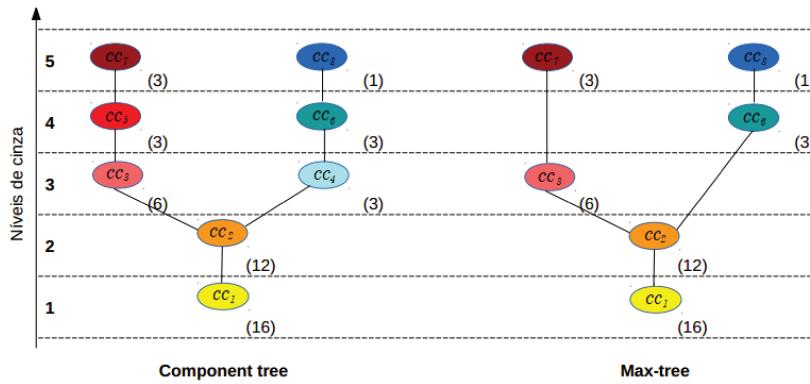


Figura 2.3: *Component tree* e *Max-tree* geradas a partir dos limiares da Figura 2.2. Os números entre parênteses são a quantidade de pixels de cada componente conexa.

Fonte: elaboração do autor.

Cada componente conexa é um nó da *Component Tree*, e é formada uma ramificação para as componentes que ela contém no limiar imediatamente acima. Por exemplo, o limiar 3 na

Figura 2.2 gera duas componentes conexas a partir da CC_2 e, portanto, duas ramificações na Figura 2.3 são geradas.

A *Max-tree* é uma representação compacta da *Component Tree*, que remove componentes conexas idênticas. Por exemplo, as componentes conexas CC_5 e CC_7 possuem a mesma formação e, portanto, permanecem idênticas durante as limiarizações 4 e 5. Devido a essa redundância, a *Max-Tree* representa essas componentes conexas apenas uma vez. O mesmo ocorre com as componentes conexas CC_4 e CC_6 , que possuem o mesmo conjunto que pixels, a representação de ambas é uma redundância removida na *Max-Tree*.

2.1.2 Transformada de *Watershed*

A segmentação de imagens tem por finalidade encontrar objetos de interesse a partir da decomposição da imagem em partições conexas. No desenvolvimento do trabalho, utilizamos uma segmentação multiescala baseada na transformada de *Watershed*, que utiliza elementos hierárquicos de forma a facilitar a segmentação a partir da decomposição das regiões em diferentes níveis.

A transformada de *Watershed* é um método clássico de segmentação no qual a imagem é interpretada como um relevo topográfico, sendo que a variação dos níveis de cinza na imagem representa a variação da altitude do relevo. Considerando que o relevo é inundado gradualmente, iniciando pelos mínimos regionais, a técnica da transformada de *Watershed* consiste na construção de barragens de contenção no momento em que dois ou mais lagos formados pela inundação tentam se unir. Essa abordagem é denominada *Watershed* tradicional e pode gerar super-segmentação da imagem quando há uma grande quantidade de mínimos regionais.

Para lidar com a super-segmentação, outras variações foram criadas na implementação da transformada de *Watershed* como, por exemplo, a *Watershed* por Marcadores e a *Watershed* Hierárquica. A técnica utilizando marcadores consiste na pré-seleção de alguns mínimos regionais da imagem para iniciar o processo de inundação. Entretanto, ao fixar os marcadores (mínimos locais) com finalidade de limitar a quantidade de barragens e reduzir o problema da super-segmentação, cria-se outro problema: como definir quais são os melhores mínimos regionais? Em casos de maior complexidade, a utilização de marcadores é inviável (BEUCHER, 1994).

Outra abordagem para reduzir o problema da super-segmentação é a utilização da hierarquia no processo da transformada de *Watershed* para selecionar regiões a serem unidas. Nesse caso, A finalidade é unir lagos vizinhos gerados pela super-segmentação, em razão de serem regiões homogêneas na prática.

2.1.3 Árvore dos Lagos Críticos

A Árvore dos Lagos Críticos (ALC) é uma estrutura hierárquica que possui como principal característica a aglomeração de regiões vizinhas, caracterizando um método *bottom-up* de criação da hierarquia, ao contrário da *Component tree* e *Max-tree* que seguem a estratégia *top-down*.

O processo de construção da árvore é realizado com base no processo de inundação da Transformada de *Watershed*. Inicialmente, a imagem é convertida em níveis de cinza (caso seja uma imagem colorida) e transformada em uma matriz de números inteiros. Os mínimos regionais dessa matriz representam os pontos mais baixos de suas respectivas regiões e, consequentemente, é por eles que o processo de inundação é iniciado. Os mínimos regionais serão as folhas da ALC e as bacias que são formadas durante o processo de inundação (junção de dois ou mais lagos) serão os novos nós. O processo é finalizado quando a imagem ficar completamente inundada.

Os conceitos se tornarão mais claros a partir do exemplo de construção da ALC mostrado na Figura 2.4. Nesse exemplo, a transformada de *Watershed* ocorre em um vetor (imagem unidimensional). Na Figura 2.4(a), os mínimos regionais são rotulados e representam as folhas da árvore. Em seguida, bacias são formadas a partir da junção dos lagos (inicialmente mínimos regionais representados como folhas). Essa junção cria os nós {9, 10, 11 e 12} da Figura 2.4(b). A inundação continua e novas bacias são representadas como novos nós da ALC, sendo que o nó gerado a partir da junção de dois ou mais lagos é ligado por meio de uma aresta aos nós que lhe deram origem nas figuras 2.4(c) e 2.4(d).

A construção da ALC utilizando o processo de inundação da Transformada de *Watershed* pode ocorrer de dois modos:

- Uniforme: o nível de água começa subir de forma constante a partir do mínimo regional de nível mais baixo. Quando o nível de água atingir a altitude de outros mínimos regionais, estes começarão a ser inundados.

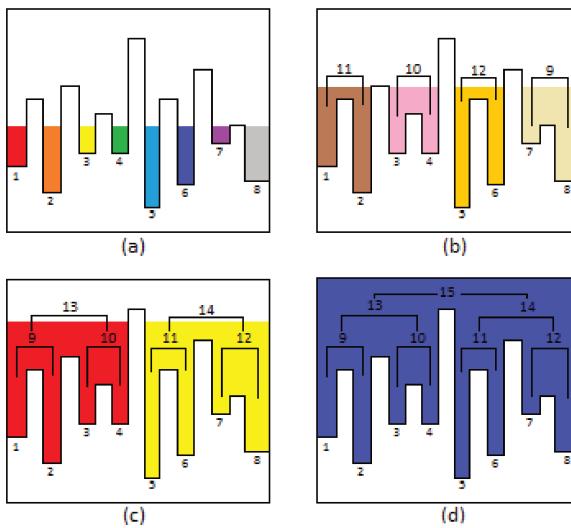


Figura 2.4: Transformada de *Watershed* em uma imagem unidimensional. Em (a), a inundação inicia pelos mínimos regionais, formando 8 lagos; Em (b), a inundação prossegue ocasionando a junção de dois ou mais lagos. Nesse momento, os lagos 1 - 2 formam a bacia 9, 3 - 4 formam a bacia 10, 5 - 6 formam a bacia 11 e 7 - 8 formam a bacia 12; Em (c), a imagem apresenta apenas duas bacias, 13 e 14, sendo elas formadas pela junção das bacias 9 - 10 e 11 - 12, respectivamente; Em (d), a junção das bacias 13 - 14 gera a última bacia possível, 15, e o processo é finalizado com a imagem completamente inundada.

Fonte: elaboração do autor.

- Síncrona: a inundação se inicia por todos os mínimos regionais simultaneamente. O nível de água em todas as bacias é o mesmo até o momento da junção com outra bacia. Desse ponto em diante, os lagos continuam a crescer separados por uma barragem.

2.2 Algoritmo Genético

De acordo com Deng, Liu e Zhou (2015), algoritmos genéticos são métodos adaptativos inspirados no processo evolutivo dos organismos e que podem ser utilizados para resolver problemas de busca e otimização. No processo evolutivo dos organismos, por muitas gerações, as populações evoluíram de acordo com os princípios da “seleção natural” e “sobrevivência do mais forte” (DARWIN, 1859). De modo semelhante, os algoritmos genéticos são capazes de evoluir para melhores soluções de um determinado problema. Segundo Beasley, Bull e Martin (1993), um algoritmo genético é composto de quatro características principais. Sendo elas:

1. Coding: uma potencial solução para o problema é representada por meio de parâmetros nesta etapa. Esses parâmetros (denominados genes) são unificados para formar um conjunto de valores, denominado cromossomos.

2. Fitness Function: deverá existir uma função para guiar a convergência do algoritmo no problema a ser resolvido. Dado um cromossomo, a função deve retornar um valor numérico que representa a utilidade do indivíduo que aquele cromossomo representa.

3. Reproduction: durante a etapa de reprodução do algoritmo genético, os indivíduos são selecionados em pares e recombinados para produzir uma nova geração. A seleção é realizada de forma aleatória. Entretanto, bons indivíduos possuem maiores chances de serem selecionados. Após a seleção dos pais, os cromossomos de ambos são recombinados, tipicamente de duas maneiras: *crossover* e *mutation*.

- **Crossover:** pode ocorrer de várias formas, a mais tradicional realiza um corte em uma posição aleatória do cromossomo para produzir duas partes. A primeira parte, a *cabeça*, é fornecida para um dos filhos, que por sua vez recebe a segunda parte, a *cauda*, de outro pai. Essa técnica é representada na Figura 2.5, adaptada de Beasley, Bull e Martin (1993).
- **Mutation:** ocorre após a criação dos filhos e é executada separadamente para cada um. Altera de modo aleatório cada gene com uma pequena chance (tipicamente 0.001). A Figura 2.5 mostra o quinto gene sofrendo mutação.

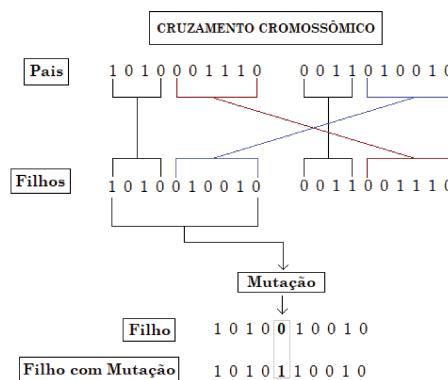


Figura 2.5: Representação dos operadores de **Crossover** e **Mutation** de Beasley, Bull e Martin (1993).

4. Convergence: no cenário em que o algoritmo genético tenha sido implementado corretamente, as populações evoluem ao longo das gerações e os atributos dos

indivíduos das últimas gerações estarão próximos do ótimo. A convergência pode ser entendida como o progresso ao longo das iterações até a uniformidade. Um gene converge quando 95% da população compartilha do mesmo valor. Portanto, a população converge quando todos os seus genes convergem Beasley, Bull e Martin (1993).

Antes da execução do algoritmo genético, deve-se assumir que uma versão adequada do problema já tenha sido codificada em um modelo de cromossomo, e que a *fitness function* já tenha sido definida. Para exemplificar as características do algoritmo genético de Beasley, Bull e Martin (1993), o Algoritmo 1 apresenta as quatro características principais.

Algorithm 1: Na primeira etapa, cria-se a população inicial e calcula-se o *fitness function* para cada indivíduo. Na segunda etapa, é verificado se a população convergiu. No cenário em que a população tenha convergido, o algoritmo é encerrado. No cenário em que a população não tenha convergido, inicia-se o processo para produzir uma nova geração. O processo é executado da seguinte forma: (i) seleção dos indivíduos, (ii) recombinação dos indivíduos da geração anterior para gerar novos indivíduos, (iii) aplicação do operador de mutação e (iv) inclusão dos indivíduos na nova geração.

```

1 Geração da População Inicial;
2 Calcular o fitness function para cada Indivíduo;
3 while !Convergiu do
4   while Populacao/2 do
5     // Ciclo de Reprodução
6     Seleciona dois Indivíduos da Antiga Geração;
7     Recombina dois Indivíduos para Criação de Dois Novos;
8     Aplica o operador de mutação;
9     Incluir os Indivíduos na Nova Geração;
10    if População Convergiu then
11      Convergiu = verdadeiro;

```

2.3 Revisão da Literatura

Na literatura existem diversos métodos para realizar a segmentação de imagens, tais como: segmentação baseada em camadas, segmentação baseada em blocos, segmentação baseada em blocos por regiões, segmentação baseada em blocos por borda, segmentação baseada em blocos por pixel e segmentação baseada em blocos por modelo (ZAITOUN; AQEL, 2015).

Cada um desses métodos possui um conjunto de técnicas que possibilita realizar a segmentação de acordo com o problema apresentado.

A quantidade de métodos se dá pelo fato de não existir uma metodologia capaz de suprir todos os desafios presentes na segmentação de imagens. Devido a isso, a maioria das técnicas possui como propósito solucionar questões de uma categoria específica de problemas como, para citar apenas um dentre vários exemplos, a segmentação de imagens de ressonância magnética do ventrículo esquerdo do coração (CARVALHO, 2004).

Grande parte das abordagens para realizar a segmentação de imagens foi desenvolvida considerando o conhecimento do usuário a respeito da situação. Desse modo, o usuário se torna responsável por informar parâmetros para a aplicação iniciar, tais como: (i) o tamanho médio dos objetos presentes na imagem, (ii) a região em que a aplicação deverá iniciar a segmentação, (iii) intervenções durante o processo de segmentação, dentre outros.

Quando o processo de segmentação depende da intervenção humana para produzir resultados, pode acarretar em situações onde os parâmetros informados não serão ideais para todo o conjunto de imagem. A dependência da interação humana pode ocasionar a falta de adaptabilidade nas técnicas, e isso pode levar a instabilidades nos resultados, principalmente para conjuntos de imagens com baixo grau de similaridade.

Em relação aos métodos de segmentação de imagens citados nesse trabalho, todos apresentam dificuldades em alguns aspectos e pontos fortes em outros, o que lhes confere características únicas.

A técnica “*Size Interval Precision*” (SIP) apresentada por Ranefall e Wählby (2016), é uma proposta de segmentação de imagem por limiarização dos níveis de cinza baseada no tamanho do objeto. A técnica é embasada nas métricas *recall* e *maximum precision* considerando os objetos dentro do tamanho estipulado previamente, ou dentro do limite definido para objetos aglomerados (“*upper limit*”). A abordagem representa os dados por meio da *Component Tree* (apresentada na Seção 2.1.1) e emprega a técnica *Union-Find*.

Levando em conta a base de dados utilizada neste trabalho, SIP está entre as técnicas que apresentam melhores resultados. No entanto, a técnica é suscetível à instabilidade de resultados em bases cujas imagens apresentam baixa similaridade entre si. Isso pode ocorrer porque a técnica requer que o usuário informe um parâmetro de entrada (tamanho do objeto) para iniciar o processo de segmentação.

A técnica “*Per Object Ellipse Fit*” (POE), apresentada por Ranefall, Sadanandan e Wählby (2016), é uma proposta de segmentação de imagem por limiarização baseado no formato elíptico e no tamanho dos objetos. A estrutura de dados é representada pela *Component Tree* e a técnica calcula para os prováveis objetos dentro do tamanho pré-determinado o índice de curvatura elíptica. O resultado gerado classifica a estrutura analisada em três categorias, sendo elas: (i) objeto, (ii) plano de fundo e (iii) provável objeto. Ao final, a técnica POE analisa novamente as estruturas classificadas como “prováveis objetos” e os reclassifica. A técnica POE foi analisada com poucas imagens pelos autores, contudo, apresentou bons resultados e alta estabilidade. No que diz respeito aos parâmetros de entrada, a técnica consegue executar a segmentação sem o conhecimento prévio dos objetos, porém, ao informá-los corretamente, a técnica apresentou resultados melhores (RANEFALL; SADANANDAN; WÄHLBY, 2016).

A técnica *Real-time Adaptive Thresholding Using the Integral Image* produzida por Bradley e Roth (2007) é uma proposta de segmentação utilizando variações espaciais na iluminação da imagem. A técnica propõe comparar cada pixel da imagem com os pixels ao seu redor. Desse modo, a técnica aplica uma média móvel nos últimos pixels e os classifica como plano de fundo (*background*) quando seu valor é inferior à média dos pixels ao seu redor. O principal objetivo da técnica é criar uma representação binária classificando cada pixel como *escuro* (plano de fundo) ou *claro* (objeto).

A técnica *Region-based for Approximating an Arbitrary 2D* (RFOVE) apresentada por Panagiotakis e Argyros (2020) é uma proposta para determinar a possível quantidade de elipses sobrepostas. A técnica utiliza uma versão modificada da técnica *Real-Time Adaptive Thresholding* de Bradley e Roth (2007) para obter resultados melhores. Com a técnica, é possível identificar quando os objetos estão sobrepostos. O objetivo principal é obter o maior percentual de cobertura dos objetos elípticos.

A técnica *Saliency Optimization from Robust Background Detection* apresentada por Zhu et al. (2014) propõe que um determinado fragmento da imagem só poderá ser considerado como plano de fundo quando existir uma forte ligação entre o fragmento e a borda da imagem. Essa técnica partiu de uma definição conceitual, na qual os objetos e o plano de fundo em imagens naturais possuem traçados diferentes e que os objetos da imagem possuem pouca ou nenhuma conectividade com as bordas da imagem. A técnica implementada em Matlab apresenta tempo de execução médio de 0,25 segundo. Os resultados possuem *f-score* acima de 90% na análise dos próprios autores.

O método proposto por Otsu (1979) possui como característica a não parametrização e o fato de ser um método não supervisionado. O método realiza a segmentação de imagens a partir de um histograma de níveis de cinza e se caracteriza por definir a limiarização “ideal” a partir da integração das características do histograma (propriedades globais) e não pela diferença de suas características (propriedades locais como, por exemplo, vales). O método possui grande relevância na comunidade científica pela simplicidade e abrangência, a qual não se restringe a limiarização dos níveis de cinza em uma imagem.

Como se pode observar, cada técnica apresenta uma peculiaridade em seu desenvolvimento e todas irão esbarrar em alguma dificuldade. As técnicas em sua maioria possuem como propósito aprimorar resultados e propor novas metodologias para trabalhos futuros, que, por sua vez, podem ser empregadas em conjunto por outros trabalhos.

O presente trabalho se baseou em alguns fundamentos e os adaptou de trabalhos já existentes, como a utilização do intervalo de tamanho de Ranefall e Wählby (2016) em conjunto com a curvatura elíptica de Ranefall, Sadanandan e Wählby (2016) e somado com a utilização do algoritmo genético (BEASLEY; BULL; MARTIN, 1993). Nossa trabalho apresenta alta estabilidade nos resultados gerados e a segurança dos parâmetros de entrada serem apenas auxiliadores e não decisivos para o resultado.

Neste trabalho, o pré-processamento utiliza simplesmente *gaussian blur* para suavizar a imagem, seguido de erosão e dilatação. A segmentação faz uso da transformada Watershed e representa os dados por meio da Árvore dos Lagos Críticos.

O que se pode observar, portanto, é a utilização de premissas já existentes nos trabalhos que se destacam, com a vantagem de preservarmos a estabilidade e a previsibilidade nos resultados gerados, sem dependermos de uma entrada acurada do usuário para termos robustez.

Durante o desenvolvimento do nosso processo de segmentação utilizamos outras técnicas para compararmos os resultados obtidos. Referidos trabalhos estão citados na Tabela 2.3 e serão apresentados posteriormente na dissertação.

Ano	Autores	Técnica	Citação
1979	Nobuyuki Otsu	Otsu	Otsu (1979)
1997	Ying Xu, Edward C. Uberbacher	Minimum	Xu e Uberbacher (1997)
2000	R. Manduchi	MoG	Manduchi (2000)
2002	Yiyong Sun, Joonki Paik, Andreas Koschan, David Page	Triangle	Sun et al. (2002)
2006	Yan Li, Xiaoping Fan, Gang Li	Renyi Entropy	Y. Li, Fan e G. Li (2006)
2009	A. Singh, S. Sawan, M. Hanmandlu	Dual Background	Singh et al. (2009)
2007	Wenbing Tao, Hai Jin, Yimin Zhang	Mean Shift	Tao, Jin e Zhang (2007)
2010	A. El-Zaart.	ISODATA	El-Zaart (2010)
2011	Chunming Li, Rui Huang, Zhaohua Ding	MRI	C. Li et al. (2011)

Capítulo 3

Metodologia Proposta

Neste capítulo, apresentamos a metodologia desenvolvida durante o mestrado para segmentação de imagens. O desenvolvimento foi baseado em três grandes etapas. A primeira etapa, apresentada na Seção 3.2, consistiu na escolha e implementação de um algoritmo para geração da árvore dos lagos críticos capaz de gerar a estrutura da árvore da maneira que necessitamos para as etapas seguintes.

A segunda etapa, apresentada na Seção 3.3, correspondeu ao estudo de métricas que podem ser aplicadas na árvore dos lagos críticos de modo a selecionar determinados nós da árvore como regiões de interesse. As métricas discutidas são baseadas em tamanho e forma das componentes conexas. A seleção de componentes conexas, em última instância, é o que gera a segmentação fornecida ao usuário como resultado final.

A terceira etapa, apresentada na Seção 3.4, consiste na implementação de um algoritmo genético para otimizar o processo de seleção dos parâmetros da segmentação. Nesse contexto, o usuário não precisa conhecer o melhor conjunto de parâmetros a ser configurado, pois o algoritmo genético tentará efetuar a otimização a partir do intervalo fornecido. Vale ressaltar que o algoritmo genético também incluirá na busca valores fora do intervalo fornecido. Isso permite ao algoritmo genético encontrar boas soluções mesmo quando o usuário efetuar escolhas erradas de parâmetros.

3.1 Diagrama dos processos

Nesta seção, apresentamos os processos executados em um único diagrama. O diagrama não possui como finalidade explicar detalhadamente os processos, mas facilitar a compreensão das próximas seções e fornecer uma visão macro de todo o desenvolvimento.

A Figura 3.1 apresenta o diagrama contendo os processos e as tomadas de decisões feitas pelo algoritmo. Os processos estão representados por retângulos e as decisões estão representadas pelos losangos. As próximas seções oferecerão mais detalhes de cada processo e das decisões tomadas.

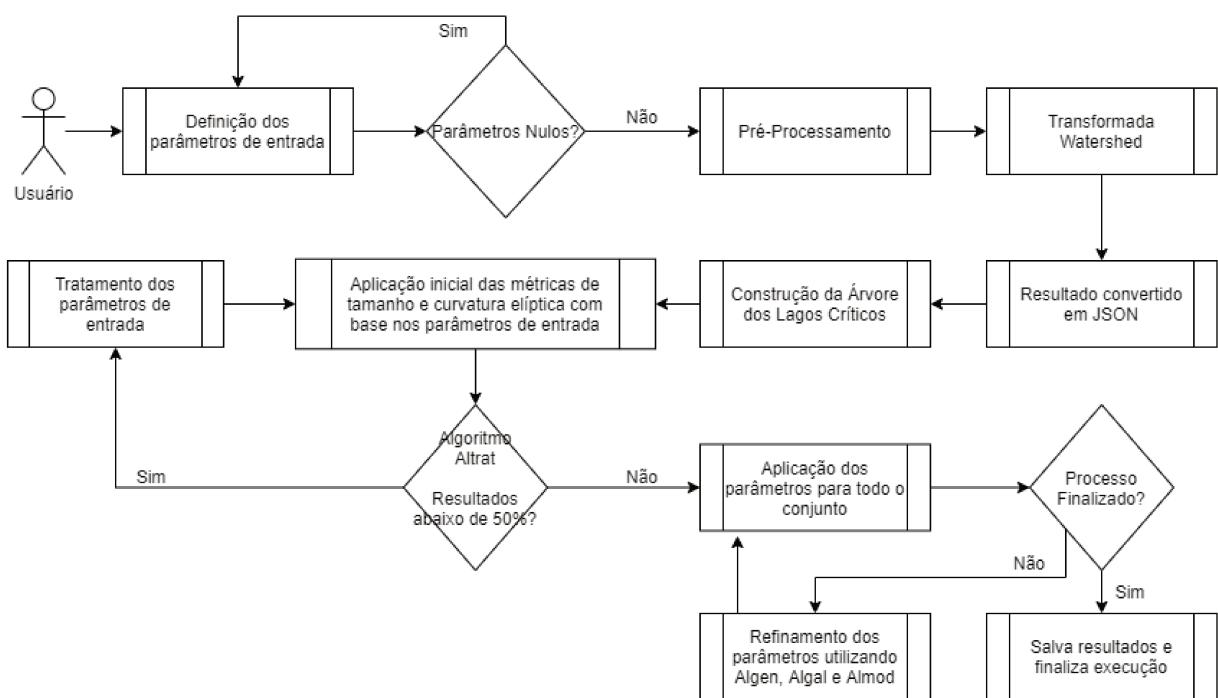


Figura 3.1: Diagrama dos processos envolvidos para realizar a segmentação proposta.

Fonte: elaboração do autor.

3.2 Geração da Árvore dos Lagos Críticos

O algoritmo implementado nesta etapa é baseado no trabalho de Falcão, Stolfi e Alencar Lotufo (2004), que apresentaram uma abordagem baseada em grafos para implementar operadores de processamento de imagem baseados em conectividade. Essa abordagem chamada de IFT pelos autores (do inglês “Image Foresting Transform”) pode ser instanciada de várias formas para gerar estruturas hierárquicas comuns na área de processamento de imagens, bastando para tanto modificar algumas opções que os autores deixam em aberto.

De forma resumida, a IFT pode ser entendida como uma generalização do algoritmo de Dijkstra para encontrar os caminhos de custo mínimo em um grafo a partir de um vértice, sendo que essa generalização consiste em permitir que, ao invés de um vértice, o ponto de partida seja um conjunto de vértices. Nesta seção, apresentamos o modo como instanciamos a IFT para os nossos propósitos e ilustramos alguns passos do algoritmo por meio de um exemplo.

A Figura 3.2 e a Figura 3.3 exemplificam a construção da Árvore dos Lagos Críticos. O ponto de partida é a imagem representada pela matriz 7×7 mostrada na Figura 3.2(a). Foram utilizados limiares de níveis de cinza que variam de 5 (Figura 3.2(c)) a 9 (Figura 3.2(g)). A Vizinhança-4 define o trajeto da inundação para pixels vizinhos a um já inundado.

Inicialmente, os mínimos regionais são rotulados na Figura 3.2(b). Assumindo que nenhum método de supressão de super-segmentação foi utilizado, inicia-se o processo de inundação na Figura 3.2(c), e as componentes conexas próximas aos mínimos regionais aumentam progressivamente de área. Não existe junção de lago até o limiar 6, o que é representado pela Figura 3.2(d).

A primeira junção entre lagos ocorre no limiar 7. Essa junção cria a bacia B_1 a partir dos lagos L_3 , L_4 e L_5 , sendo que cada junção gerará um novo nó na ALC. A inundação continua e, no limiar 8, ocorre a junção entre a bacia B_1 e o lago L_2 , gerando a bacia B_2 . Por fim, a inundação é finalizada na Figura 3.2(g), quando toda a imagem está submersa e a última bacia B_3 é criada a partir de B_2 e L_1 .

Para construção da ALC na Figura 3.3, foi utilizado o mecanismo da transformada *Watershed* apresentado na Figura 3.2. Os lagos marcados como mínimos regionais L_1 , L_2 , L_3 , L_4 e L_5 correspondem aos nós folhas. A formação dos nós superiores ocorre pela junção desses lagos ao longo da inundação. Como exemplo, a bacia B_1 , criada pela junção dos lagos L_3 , L_4 e L_5 . Esse processo ocorre até que toda a imagem esteja imersa, não havendo mais regiões para concatenar.

O Algoritmo 2 apresenta um pseudocódigo do que foi implementando para gerar a árvore dos lagos críticos. O algoritmo recebe como entrada uma matriz de pixels que representa a imagem e um conjunto de pontos que iniciarão o processo de inundação. Esse conjunto de pontos é chamado de *Seed* e pode ser obtido computando os mínimos regionais. O algoritmo também precisa da relação de adjacência, que, no caso, foi definida como a **Vizinhança-4**.

As estruturas de dados auxiliares deste algoritmo são:

Algorithm 2: Entrada: uma matriz $Imagen$ de dimensões $N \times M$ que representa a imagem, um conjunto $Seed$ de coordenadas da imagem de dimensão K que identifica as células onde o processo de inundação inicia, e uma relação de adjacência Adj . Estruturas de Dados Auxiliares: uma fila de prioridade mínima $Queue$ e três matrizes $Costs$, $Paths$ e $Nodes$ de dimensões $N \times M$ para armazenar, respectivamente: (i) os custos dos menores caminhos, (ii) todos os caminhos de custo mínimo que levam até uma determinada célula e (iii) referências aos nós na árvore dos lagos críticos, respectivamente. Saída: raiz da árvore dos lagos críticos. A partir da raiz, todos os outros nós podem ser acessados facilmente.

```

// Inicialização das estruturas de dados.
1 for  $i \leftarrow 0$  to  $N$  do
2   for  $j \leftarrow 0$  to  $M$  do
3      $Costs[i][j] \leftarrow \infty$ 
4      $Paths[i][j] \leftarrow []$ 
5      $Nodes[i][j] \leftarrow \text{null}$ 

6 for  $i \leftarrow 0$  to  $K$  do
7    $cell \leftarrow Seed[i]$ 
8    $Queue.push(cell)$ 
9    $Costs[cell[0]][cell[1]] \leftarrow Imagem[cell[0]][cell[1]]$ 

10 while  $Queue$  is not empty do
11    $minCostCell \leftarrow Queue.pop()$ 
12    $i, j \leftarrow minCostCell$ 
    // A junção de duas bacias ocorre se um determinado nó é
    // alcançado por caminhos provenientes de mínimos regionais
    // diferentes.
13   if  $Paths[i][j]$  contém caminhos de dois nós diferentes em  $Seed$  then
    // Como esta célula foi inundada por duas fontes,
    // criamos um novo nó interno na Árvore dos Lagos
    // Críticos.
14     Step 1: criar um novo nó e atribuir a  $Nodes[i][j]$ .
15     Step 2: adicionar as coordenadas  $(i, j)$  ao novo nó.
16     Step 3: Criar uma aresta deste nó para os nós que estão em  $Paths[i][j]$ .

17   else
    // A célula foi inundada a partir de apenas uma fonte,
    // então simplesmente aumentamos a bacia com um pixel
    // extra.
18     Step 1: atribuir esta coordenada ao nó que está em  $Nodes[i][j]$ .
19   // Adicionamos na fila de prioridade  $Queues$  todas as células
   // que podem ser inundadas.
20   foreach  $cell \in Adj(minCostCell)$  do
21      $adj_i, adj_j \leftarrow cell$ 
22      $Costs[adj_i][adj_j] \leftarrow Imagem[adj_i][adj_j]$  // O custo deixará de ser
       //  $\infty$  e passará a ser a intensidade de cinza do pixel.
      $Paths[adj_i][adj_j].append(minCostCell)$  // Um caminho do nó
       // retirado da fila de prioridade até o nó vizinho é
       // criado.

23 return  $Nodes$ 
```

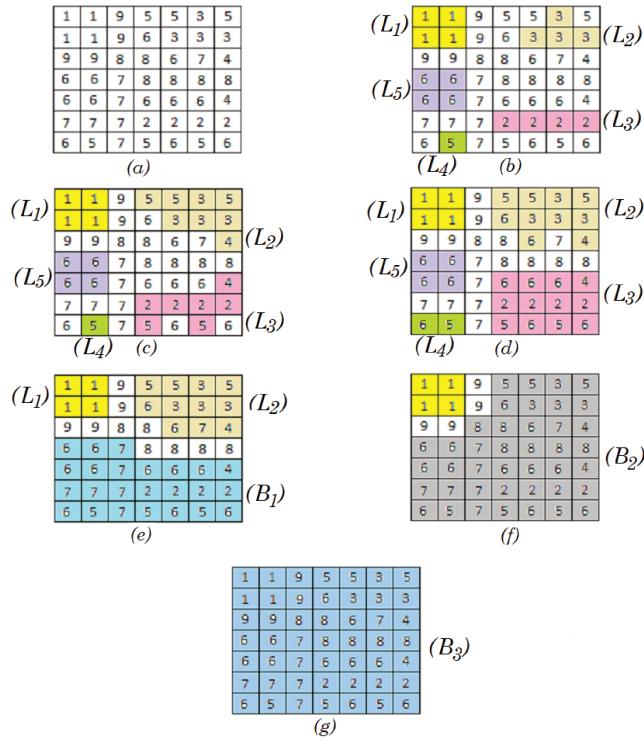


Figura 3.2: Processo de inundação da Árvore dos Lagos Críticos para uma imagem. Em (a) temos Matriz original; (b) apresenta o processo de rotulação dos mínimos regionais; Em (c), a inundação se inicia tendo 5 como limiar; Em (d), a inundação continua, contudo, nenhuma junção entre os lagos ocorre até o limiar 6; Em (e), a Bacia B_1 é formada a partir da junção dos lagos L_3 , L_4 e L_5 com o limiar 7; Em (f) temos a formação da segunda bacia B_2 a partir de B_1 e L_2 , o que ocorre com o limiar 8; Em (g), a imagem está completamente inundada e há a formação da última bacia B_3 a partir de B_2 e L_1 .

Fonte: elaboração do autor.

- Uma fila de prioridade mínima $Queue$ para que seja possível obter rapidamente o elemento com o menor nível de cinza ao lado da região inundada.
- Uma matriz de custos $Costs$ para armazenar o custo dos pixels já inundados ou que são adjacentes a um pixel inundado. Quando uma dada célula ainda não foi inundada e não está adjacente a uma região inundada, o seu custo é infinito.
- Uma matriz $Paths$ que contém todos os caminhos de custo mínimo que levam até uma célula a partir de uma posição em $Seed$.
- Uma matriz $Nodes$ que armazena referências a todos os nós da árvore dos lagos críticos.

As matrizes $Costs$, $Paths$ e $Nodes$ possuem dimensão $N \times M$, sendo que N e M são as dimensões da imagem de entrada. A saída do algoritmo é a matriz $Nodes$, que possui todas as informações necessárias e suficientes para criar a Árvore dos Lagos Críticos.

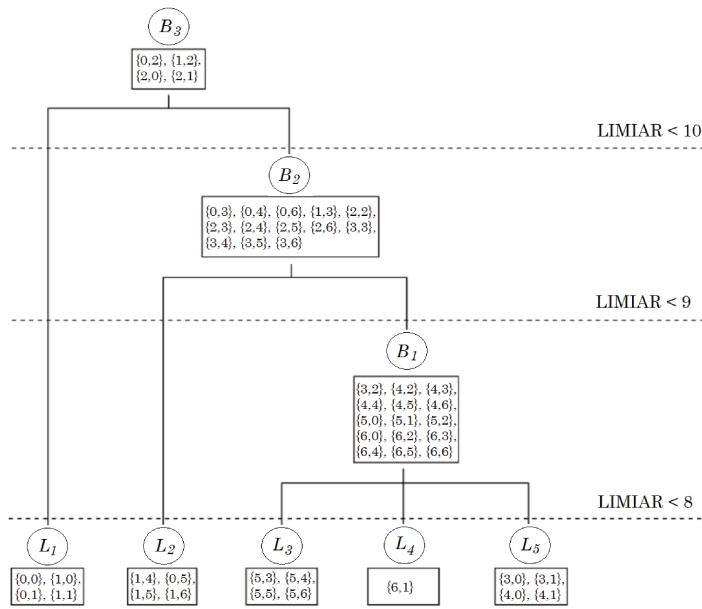


Figura 3.3: Representação da Árvore dos Lagos Críticos criada por meio da transformada de Watershed da Figura 3.2. Os pixels referentes a cada nó da árvore são mostrado no quadro abaixo de seu respectivo }lago/bacia". As bacias representam a junção de um ou mais lagos e possuem todos os pixels dos nós descendentes, bem como os pixels no momento da junção.

Fonte: elaboração do autor.

O algoritmo começa inicializando as matrizes *Costs*, *Paths* e *Nodes* (Linhas 1–5). Feito isso, as coordenadas em *Seed* são adicionadas na fila de prioridade *Queue* (Linhas 7–8). Essas coordenadas são mínimos regionais que representam o início da busca pelo caminho de menor custo. A matriz *Costs* recebe a intensidade de cinza dos mínimos locais, dado que consideramos que essas células são candidatas a serem inundadas.

Terminados os passos iniciais de configuração das estruturas de dados, inicia-se o bloco principal (Linhas 10–23), que permanece em repetição enquanto a fila de prioridade não estiver vazia. A cada iteração dessa repetição, a célula de menor custo é retirada da fila de prioridade, lembrando que o custo neste algoritmo é definido como a intensidade de cinza das células já inundadas.

Quando uma célula é removida da fila de prioridade, dizemos que ela foi inundada, sendo que pode participar de uma junção caso seja adjacente a duas bacias (nesse caso, um novo nó é adicionado na árvore dos lagos críticos), ou fará parte de um lago já existente (nesse caso, a coordenada da célula inundada é adicionada em um nó já existente na árvore dos lagos críticos). Por legibilidade, usamos as variáveis i e j para representar as coordenadas da célula de menor custo na iteração corrente (Linha 12).

A árvore dos lagos críticos possui vários nós que representam a junção das bacias. Essa junção ocorre sempre que a célula retirada da fila de prioridade foi inundada a partir de dois lagos diferentes. Para saber se isso ocorreu, a matriz de caminhos *Paths* é acessada para verificar se existe mais de um caminho começando em sementes diferentes que inundam a célula (Linha 13). Caso um novo nó seja necessário, o processo de criação segue os três passos apresentados nas linhas 14–16. Caso contrário, a célula é simplesmente inundada e apenas adicionamos as coordenadas no nó atual, indicando que esta célula faz parte da bacia.

Por fim, a fila de prioridade recebe todos os nós adjacentes à célula de custo mínimo, o que indica que os adjacentes são agora candidatos à inundação na próxima iteração.

3.3 Métricas para identificação dos objetos

Esta seção apresenta as métricas para identificar os objetos de interesse na imagem utilizando como estrutura representativa a árvore dos lagos críticos. As métricas utilizadas foram adaptadas de Ranefall, Sadanandan e Wählby (2016) e Ranefall e Wählby (2016), os quais utilizam como principais característica de seleção o formato e o tamanho dos objetos. As métricas foram originalmente aplicadas no contexto da *Component Tree*, mas se mostraram apropriadas também para a árvore dos lagos críticos.

Nosso algoritmo solicita ao usuário um intervalo mínimo e máximo de tamanho dos objetos de interesse. O tamanho se refere ao número de pixels que se espera que os objetos de interesse possuam. A partir desse intervalo, verificamos se uma dada componente conexa (obtida a partir de um nó da árvore dos lagos críticos) pode representar um objeto de interesse. Se uma componente conexa está dentro do intervalo fornecido pelo usuário, ela recebe uma pontuação $score_tamanho = 1$, o valor máximo possível.

Se uma componente conexa está fora do intervalo fornecido pelo usuário, ela ainda pode ser considerada pelo algoritmo. Sejam lim_inf e lim_sup os limitantes inferiores e superiores fornecidos pelo usuário para a quantidade pixels dos objetos de interesse, respectivamente, as componentes conexas no intervalo estendido $\left[\frac{2}{3}lim_inf, \frac{4}{3}lim_sup\right]$ serão consideradas, mas algumas serão penalizadas por excederem o limite fornecido pelo usuário. Isso é feito com a intenção de diminuir a sensibilidade dos resultados em relação a esse parâmetro, dado que o intervalo inserido pode conter imprecisões.

Assim, a pontuação é obtida de duas formas, o que depende do fato de o objeto se encontrar dentro ou fora do intervalo definido pelo usuário. A equação 3.1 apresenta o funcionamento do score do tamanho.

$$score_tamanho = \begin{cases} 1 & \text{se } limite_inferior \leq valor \leq limite_superior \\ \frac{limite_superior}{tamanho_objeto} & \text{se } valor \geq limite_superior \\ \frac{tamanho_objeto}{limite_inferior} & \text{se } limite_inferior \geq valor \end{cases} \quad (3.1)$$

A partir dessa pontuação, o algoritmo irá percorrer os nós da *Árvore dos Lagos Críticos* em busca de componentes conexas que se enquadram no intervalo estendido. Qualquer componente fora do intervalo estendido estará automaticamente descartada. As componentes que se enquadram no intervalo original fornecido pelo usuário terão o *score* de tamanho automaticamente definido com o valor máximo (valor 1). Os objetos que estiverem fora do intervalo original, mas nos limites estendidos, terão *score* menor do que 1, o que será calculado utilizando o tamanho do objeto e o valor do limite excedido.

Na análise estatística dos resultados conduzida no Capítulo 4, trabalhamos com as células de curvaturas elípticas *Hoeschst 33342* apresentadas por Coelho, Shariff e Murphy (2009b). Por esse motivo, iremos apresentar o que foi desenvolvido assumindo que os objetos que queremos segmentar têm formatos próximos do elípticos. Esse comportamento pode ser alterado para outras formas, caso fosse decidido segmentar outros objetos.

Para todas as componentes conexas dentro do intervalo estendido de tamanho, computamos o quanto próximo elas estão de um formato puramente elíptico. Para executar esse cálculo, computamos uma elipse ideal que circunscreve o objeto. Feito isso, é possível obter um número de zero até um, sendo zero o valor que corresponde a um objeto com formato muito longe de uma elipse, e um o valor concedido a um objeto elíptico. O cálculo para o *score* da forma é obtido com a fórmula:

- Cálculo do *score* da forma utilizando a curvatura elíptica ideal e a área do objeto.

$$score_forma = \frac{area_objeto}{area_elipse} \quad (3.2)$$

Em nosso algoritmo, verificamos o tamanho do objeto e qual seria a elipse ideal ao redor do objeto. A partir desse ponto, calculamos a área de cobertura do objeto em relação à área calculada da elipse ideal. A Figura 3.4 exemplifica este cenário e apresenta uma lista dos objetos encontrados com maior frequência durante a segmentação. A imagem foi adaptada de Ranefall, Sadanandan e Wählby (2016) e aponta o valor estimado para cada um dos tipos de objetos encontrados. Como mencionado anteriormente, quanto mais perto do valor 1 (apresentado abaixo do objeto), mais próximo de ser uma elipse perfeita. O pseudocódigo para calcular a curvatura elíptica ideal é apresentado no Algoritmo 3. O mesmo foi adaptado de Burger e James Burge (2008).



Figura 3.4: Resultados dos cálculos de proximidade de uma curvatura elíptica. Os números abaixo das figuras representam a estimativa de alguns objetos comuns e referida informação foi adaptada de Ranefall, Sadanandan e Wählby (2016).

Computado o *score* da forma, um limiar irá determinar quais objetos estão dentro do formato aceitável. Em seguida, *score_tamanho* e *score_forma* são unificados em uma média ponderada:

- Cálculo para o score final.

$$\text{score} = \frac{a \times \text{score_tamanho} + b \times \text{score_forma}}{a + b} \quad (3.3)$$

As variáveis *a* e *b*, bem como a variável *limiar_forma* que define um limiar para que objetos sejam descartados por causa da forma que possuem, são configuradas utilizando um algoritmo genético, apresentado na Seção 3.4. Nesse contexto, o algoritmo genético tem liberdade de privilegiar ora o tamanho (pesos mais altos para *a*), ora a forma (pesos mais altos para *b*). Além disso, pode escolher os *thresholds* que julgar mais apropriados.

3.4 Algoritmo Genético

O algoritmo genético em nosso trabalho foi implementado devido à necessidade de computar os melhores valores dos parâmetros mencionados na seção anterior. Em decorrência desse

Algorithm 3: A função **momentum** calcula os primeiros momentos do objeto e recebe como parâmetro de entrada: (i) o conjunto de pixel que representa o objeto, (ii) um valor inteiro que será utilizado para elevar a coordenada “x” e (iii) um valor inteiro que será utilizado para elevar a coordenada “y” a uma dada potência. A função **computeEllipseFit** é responsável por calcular a curvatura elíptica do objeto e recebe como parâmetro o conjunto de pixel que representa o objeto. A função realiza a somatória das coordenadas invocando a função **momentum**.

```

1 Function momentum (pixels, powx, powy)
2   mom  $\leftarrow$  0;
3   foreach el  $\in$  pixels do
4     mom  $\leftarrow$  (el[0]powx  $\times$  el[1]powy)
5   return mom;
6 Function computeEllipseFit (pixels)
7   m00  $\leftarrow$  momentum(pixels, 0, 0); // Corresponde ao tamanho do objeto
8   xs  $\leftarrow$  momentum(pixels, 1, 0); // Soma das coordenadas do eixo x
9   ys  $\leftarrow$  momentum(pixels, 0, 1); // Soma das coordenadas do eixo y
10  xxs  $\leftarrow$  momentum(pixels, 2, 0); // Soma das coordenadas do eixo x2
11  yys  $\leftarrow$  momentum(pixels, 0, 2); // Soma das coordenadas do eixo y2
12  xys  $\leftarrow$  momentum(pixels, 1, 1); // Soma das coordenadas x  $\times$  y
13  xsxs  $\leftarrow$  (xs)2;
14  xsys  $\leftarrow$  xs * ys;
15  ysys  $\leftarrow$  (ys)2;
16  m02  $\leftarrow$  yys - (ysys/m00)
17  m11  $\leftarrow$  xys - (xsys/m00)
18  m20  $\leftarrow$  xxs - (xsxs/m00)
19  M[0, 0]  $\leftarrow$  [m02];
20  M[0, 1]  $\leftarrow$  [m11];
21  M[1, 0]  $\leftarrow$  [m11];
22  M[1, 1]  $\leftarrow$  [m20];
23  lambda1, lambda2 = eigenvalue(M)
24  a = 2  $\times$   $\sqrt{\lambda_1/m_{00}}$ 
25  b = 2  $\times$   $\sqrt{\lambda_2/m_{00}}$ 
26  return m00/( $\pi \times a \times b$ )

```

cenário, criamos duas abordagens distintas: (i) a primeira abordagem executa rotinas de maneira semi-aleatória e (ii) a segunda executa rotinas fundamentadas nas características de um algoritmo genético. Implementamos essas duas versões para obter um efeito comparativo, pois queríamos validar o quanto a execução de um algoritmo genético seria superior a de um algoritmo que executa de maneira semi-aleatória. Denominamos os algoritmos semi-aleatório e algoritmo genético de Algal e Algen, respectivamente.

O algoritmo Algal possui uma rotina de execução simples, composta por dois passos. Um pseudocódigo do Algal é apresentado no Algoritmo 4. No primeiro passo, é solicitado do usuário um tamanho estimado dos objetos de interesse, por exemplo, 100 pixels. Após receber o valor de entrada, o algoritmo cria um intervalo de tamanho que representa 33% do valor informado. Em outras palavras, se for recebido o valor de 100 pixels, o intervalo seria de 67 até 133 pixels de tamanho. No segundo passo, o algoritmo Algal itera calculando a eficiência dos parâmetros em cada execução. Desse modo, é criada uma lista com as melhores segmentações.

Algorithm 4: Pseudocódigo do algoritmo Algal. O processo é iniciado solicitando ao usuário um tamanho estimado dos objetos. Em seguida verificamos a existência de imagens, e realizamos a segmentação para cada imagem de acordo com a quantidade de repetições. O resultado de cada repetição é salvo em uma lista e no final da última execução, o melhor resultado é selecionado de acordo com a métrica Almod.

```

1 Início;
2 Recebe Parâmetros Iniciais;
3 foreach Imagem do
4   while Contador < Repetições do
5     // A segmentação ocorre dentre dos limites definidos
      // pelo usuário, não existindo nenhum reajuste de valor,
      // ao final, executamos a avaliação do resultado pelo
      // algoritmo Almod.
6     executarSegmentacao();
7     avaliarResultado();
8     ranquearResultados();
Fim;
```

Para definir qual a melhor segmentação, uma pontuação aqui denominada de Almod é fornecida para cada uma delas da seguinte forma: $Almod(\mathcal{B}) = \sum_{i=1}^n \sum_{j=1}^m |\mathcal{I}_{i,j} - 255 \times \mathcal{B}_{i,j}|$, onde \mathcal{I} é a imagem original e \mathcal{B} é a imagem segmentada.

Criamos Almod para avaliar pixel a pixel os resultados gerados nas duas abordagens. Essa métrica calcula para cada pixel a diferença entre o valor de cinza da imagem original e seu correspondente na imagem gerada após o processo de segmentação. Note que a matriz

segmentada possui apenas zeros e uns, o que obriga a multiplicar pelo número 255 para se adequar ao domínio dos valores dos pixels da imagem original. Isso torna possível identificar os bons resultado.

A Figura 3.5 ilustra o funcionamento do *Almod*. A matriz no lado esquerdo representa a imagem original e a matriz no lado direito representa a imagem segmentada. Os pixels com valores 55 representam um objeto (*foreground*) na imagem original, e os pixels com valores de 155 e 255 representam o plano de fundo (*background*). Quanto maior a diferença entre os pixels da imagem original com seus respectivos representantes na imagem segmentada, maior a chance de o resultado não ter sido bom.

Imagen Original (IO)						Imagen Gerada (IG)					
1	2	3	4	5	6	1	2	3	4	5	6
1	55	55	55	55	55	255	255	255	255	255	255
2	155	255	55	55	255	255	255	255	255	255	255
3	55	55	55	55	55	255	0	0	0	0	0
4	155	255	55	55	255	255	255	255	255	255	255

Figura 3.5: Diferença entre o módulo da imagem original vs imagem gerada.

Fonte: elaboração do autor.

No algoritmo Algen, dividimos a execução em quatro etapas. A primeira etapa é responsável por valorar os parâmetros iniciais (intervalos máximos de tamanho conforme recebidos do usuário, parâmetros a , b e *limiar_forma*) e definir a população inicial. A cada nova geração, metade da população será renovada. Por exemplo, supondo que a população inicial tenha 50 indivíduos, a geração seguinte manterá 25 dos 50 indivíduos iniciais. Essa abordagem é chamada de *elitismo* e é usada principalmente para não perdermos os melhores cromossomos entre uma geração e outra.

Para definir a população inicial, geramos características aleatórias para cada indivíduo da população tendo como base o tamanho dos objetos informado na primeira etapa. As características geradas são: peso do *score* do tamanho a , peso do *score* da forma b , dimensões mínimas e máximas e limite para descarte de componentes pela forma.

Após a criação dos indivíduos da primeira geração, iniciamos o processo de escolha das componentes conexas a partir da árvore dos lagos críticos. Esse processo de escolha é realizado percorrendo todos os nós da árvore dos lagos críticos e pontuando individualmente cada componente conexa conforme a fórmula do *score*. Os nós com os melhores *scores* serão

selecionados, gerando, ao final, a segmentação de acordo com os cromossomos dos indivíduos.

Ao término da segmentação, uma *fitness function* pontua a utilidade de cada indivíduo. A *fitness function* aqui utilizada é a métrica A1mod apresentada anteriormente. Seja \mathcal{I} a imagem de entrada e \mathcal{B} a imagem limiarizada obtida ao final da computação, a *fitness function* realiza a seguinte operação: $fitness_function = \sum_{i=1}^n \sum_{j=1}^m |\mathcal{I}_{i,j} - 255 \times \mathcal{B}_{i,j}|$. Note que quanto menor o valor da *fitness function*, melhor o resultado da segmentação e mais adaptado será considerado o cromossomo.

Computada a *fitness function*, excluímos 50% da população que corresponde aos piores indivíduos. Em seguida, iniciamos o processo de criação dos novos indivíduos. Os novos indivíduos serão uma média das características dos pais. Para realizar o cruzamento, pareamos o melhor indivíduo da população inicial com o último melhor. Por exemplo, caso tenhamos uma população remanescente de 16 indivíduos realizaremos o cruzamento entre o primeiro e o décimo-sexto; segundo com o décimo-quinto; e assim sucessivamente.

Gerada a segunda geração de indivíduos, iniciamos o processo de mutação. Nessa etapa, cada indivíduo da lista (novos e antigos) têm 10% de chance de sofrer mutações em seus genes. A mutação sofrida será uma alteração de até 15% do valor naquele gene. Por exemplo, supondo que o valor em um gene seja o número 100, o mesmo pode variar de 85 a 115. Após a mutação, o processo se repete iterativamente e termina quando um número pré-definido de gerações tiver sido alcançado.

O Algoritmo 5 apresenta o pseudocódigo do A1gen. Em resumo, iniciamos a execução do algoritmo recebendo o parâmetro de entrada, definindo a população inicial que será processada e o número de gerações (Linhas 2–4). Em sequência, parametrizamos a população inicial e executamos a segmentação para cada indivíduo da população.

Após a segmentação para toda a população, iniciamos a função para selecionar a melhor metade (essa comparação é feita utilizando o resultado gerado por cada iteração na Linha 8). Em seguida, executamos a função para gerar novos indivíduos a partir dos melhores. Nessa etapa, é realizada uma média simples entre as duas extremidades da população. Depois da criação dos novos indivíduos, iniciamos o processo de mutação, no qual existe uma chance de mutação de 10% por indivíduo (Linha 28). Quando um individuo sofre mutação, cada parâmetro sofre uma alteração que varia de 5% à 15% (Linha 29). Por fim, iniciamos uma nova geração e repetimos o procedimento até que todas as gerações tenham sido executadas.

Algorithm 5: Pseudocódigo do algoritmo Algen. O processo é iniciado solicitando ao usuário os parâmetros inciais. Em seguida a população e as gerações são definidas de acordo com a quantidade de imagens a serem segmentadas. Na função principal, executamos a segmentação para cada indivíduo da população, comparamos os resultados gerados utilizando o algoritmo Almod e armazenamos em uma lista. Na sequência excluímos a pior metade da população, criamos novos indivíduos por meio do cruzamento cromossômico e avaliamos a possibilidade de mutação para cada indivíduo. A função principal se repete até que todos os indivíduos de todas as gerações tenham sido processados.

```

1 1 Início;
2 Recebe Parâmetros Iniciais;
3 Define População Inicial;
4 Define Quantidade de Gerações;
5 Main Inicio Algen
6   while Existe Geração do
7     while Existe População do
8       executarSegmentação(Individuo);
9       avaliarResultado(Individuo);
10      armazenar(Individuo);
11      // Neste ponto executamos a segmentação de acordo com
12      // as características de cada indivíduo, as quais são
13      // modificadas ao longo das gerações, e avaliamos os
14      // resultados por meio do algoritmo Almod.
15      SelecionarMelhoresIndividuos();
16      CriarNovosIndividuos();
17      ExecutaChanceMutação();
18 Function CarregarParametros
19   if Primeira População then
20     Define Intervalo dos Parâmetros;
21     Criação dos Indivíduos da Primeira População;
22 Function SelecionarMelhoresIndividuos
23   Ordena os Indivíduos da População pela Métrica do Módulo;
24   Exclusão da Pior Metade da População;
25 Function CriarNovosIndividuos
26   Inicia Processo de Junção dos Indivíduos;
27   Inclusão dos Novos Indivíduos na População;
28 Function ExecutaChanceMutação
29   while Existe População do
30     Mutacao = Geração Aleatória de 0.00 to 1.00;
31     if Mutacao <= 0.10 then
32       AtributoMutacao = Geração Aleatória de 0.85 to 1.15;
33       Atributos do Indivíduo × AtributoMutacao;
34   Fim;

```

3.5 Estabilidade dos Parâmetros de Entrada

No uso do algoritmo genético, valores imprecisos de parâmetros de entrada podem ser fornecidos accidentalmente pelo usuário. Para situações semelhantes a essa, criamos um mecanismo que avalia os cinco primeiros resultados e calcula a medida F (F -Score) antecipadamente. Essa verificação é executada apenas para medir a qualidade do parâmetro de entrada (intervalo de tamanho), sendo executada antes do início do algoritmo *Algen* ou do algoritmo *Algat*.

A verificação consiste na análise dos cinco primeiros resultados. Nos casos em que os cinco primeiros resultados apresentam valores de medida F (F -Score) inferiores a 50%, interrompemos o processo de segmentação e iniciamos o processo para tratamento do parâmetro de entrada.

O pseudocódigo para tratar os parâmetros de entrada está em Algoritmo 6 e consiste na busca do tamanho médio dos objetos em que se obtenha resultados na medida F (F -Score) minimamente aceitáveis. O algoritmo funciona em ciclos de execuções com tamanhos pré-fixados, que aumentam gradualmente ao longo da execução do algoritmo. O critério de parada é atingido quando obtemos cinco resultados sequenciais acima de 50% na medida F (F -Score).

Algorithm 6: Iniciamos a execução do *Altrat* recebendo os parâmetros de entrada e em seguida definimos um contador que será utilizado como critério de parada. O algoritmo calcula as próximas cinco medidas F e caso todas sejam acima de 50% encerra o refinamento inicial dos parâmetros. Caso algum dos cinco resultados seja inferior a 50% de medida F , o algoritmo irá recalcular os valores inciais e tentar novamente.

```

1 Início
2 Recebe Parâmetros Iniciais;
3 while contador < 5 do
4   contador ← Executa e Armazena o Resultado de Cinco Segmentações;
5   if contador < 5 then
6     | Recalcula Valores Inicias;
7 Fim;
```

Esse tratamento preserva a boa qualidade dos resultados em detrimento do tempo de execução. Quando o algoritmo inicia o tratamento dos parâmetros, o tempo de execução aumenta, pois, o algoritmo interpreta que o valor informado pelo usuário não corresponde ao tamanho dos objetos na imagem e, a partir desse ponto, inicia o processo para encontrar um valor inicial que gere resultados na medida F (F -Score) minimamente aceitáveis.

Capítulo 4

Resultados

Neste capítulo, apresentamos os resultados obtidos utilizando o nosso processo de segmentação. O capítulo foi dividido em cinco seções. Na Seção 4.1, apresentamos algumas das características das bases que utilizamos para o processo de segmentação e na Seção 4.3, tratamos do modo como os resultados são avaliados e como abordamos o processo de avaliação.

A partir da Seção 4.4, tratamos dos resultados dos algoritmos que produzimos e, especificamente, avaliamos a eficiência da métrica *Almod* e seu potencial para classificar os resultados. Na Seção 4.5, realizamos medições entre os dois algoritmos de segmentação produzidos neste trabalho, o *Algen* e o *AlgAl*. Vale ressaltar que as medições foram subdivididas em três partes, sendo elas: (i) evolução dos resultados, (ii) tempo de execução e (iii) estabilidade.

Por fim, a Seção 4.6 apresenta resultados das comparações de nossos algoritmos com outros na literatura. Essa parte também foi subdividida em três, sendo elas: (i) estudo da volatilidade, (ii) desempenho utilizando a Medida *F* e (iii) capacidade de processamento.

4.1 Base de Dados

O primeiro conjunto de imagens utilizado neste trabalho foi o das células *Hoechst 33342*. O conjunto foi fornecido pelo laboratório Murphy, que categorizou as células em dois tipos, sendo eles: (i) *3T3* e (ii) *U20S*.

As imagens disponibilizadas no site do laboratório se encontram nos seguintes formatos: *PNG* (*Portable Network Graphics*), *XCF* (*Experimental Computing Facility*) e *PSD* (*Photoshop*

Document), e, para sua obtenção, o laboratório Murphy utilizou dois microscópios distintos. Algumas dessas imagens são apresentadas na Figura 4.1 para fins de ilustração.

O conjunto das células *Hoechst 33342* contabilizam um total de 50 imagens, contudo, apenas 48 dessas imagens apresentam seus respectivos *ground-truths*. Vale mencionar que *ground-truth* consiste no contorno manual das células (objetos) contidas nas imagens e está representado na Figura 4.2. Esse conjunto de imagens é um *benchmark* bastante consolidado, já citado em 177 artigos até o momento da escrita deste trabalho.

O segundo conjunto de dados utilizado para a construção deste trabalho corresponde ao agrupamento de células adenocarcinoma do cólon HT-29, obtidas pela empresa {{American Type Culture Collection (ATCC)}. Lindblad e Bengtsson (2001) realizaram a correção do plano de fundo das imagens e delinearam o *ground-truth* manualmente utilizando a ferramenta *live-wire* do *ImageJ*. A Figura 4.3 apresenta essas células.

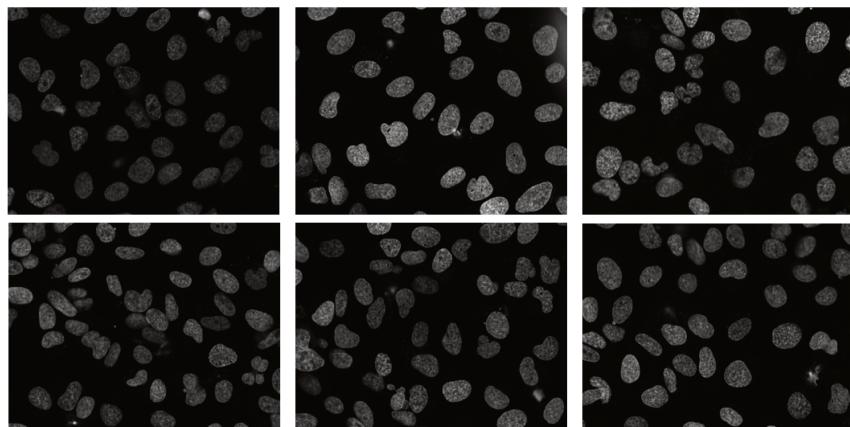


Figura 4.1: Imagens de células *Hoechst 33342* (COELHO; SHARIFF; MURPHY, 2009a).

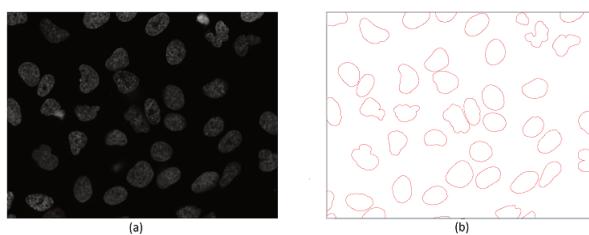


Figura 4.2: Exemplo de segmentação manual de células *Hoechst 33342*. Em (a) temos a imagem original e em (b) temos o *ground-truth* segmentado manualmente a partir da imagem original (COELHO; SHARIFF; MURPHY, 2009b).

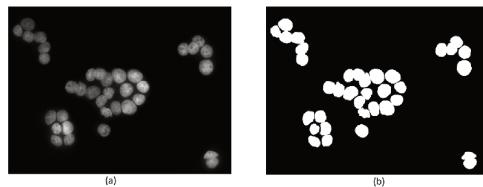


Figura 4.3: Células adenocarcinoma. Em (a), temos a imagem original. Em (b) temos o *ground-truth* (LINDBLAD; BENGTSSON, 2001).

4.2 Etapas e Resultados da Segmentação

Nesta seção apresentamos as etapas até a segmentação final de uma imagem e dois resultados obtidos pelo nosso processo de segmentação, sendo o primeiro referente ao Laboratório Murphy e o segundo referente a empresa ATCC. As segmentações escolhidas como exemplo possuem medida F de 84%, a qual representa nossa média dos valores obtidos na medida F .

Em nossa segmentação processamos cada objeto individualmente e posteriormente remontamos toda a imagem. Esse processo está exemplificado na Figura 4.4, a qual demonstra a segmentação de cada objeto da imagem.

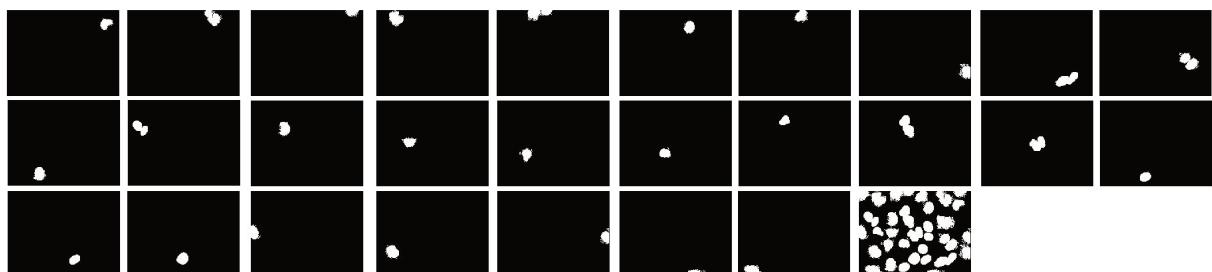


Figura 4.4: Todas as nossas segmentações são realizadas por etapa. Esta imagem é um exemplo de como segmentamos utilizando o algoritmo *Algen*. Ressaltamos que esses paços foram gravados única e exclusivamente como exemplo, pois, em uma execução normal as segmentações ficam em memória e não são gravadas fisicamente.

Fonte: elaboração do autor.

O resultado médio do nosso processo de segmentação possui uma medida F superior a 80%, sendo que a maioria desses resultados são encontrados nas últimas gerações do algoritmo genético, que denominamos de *Algen*. Nas Figuras 4.5 e 4.6 apresentamos o *ground-truth* e a imagem segmentada consecutivamente.

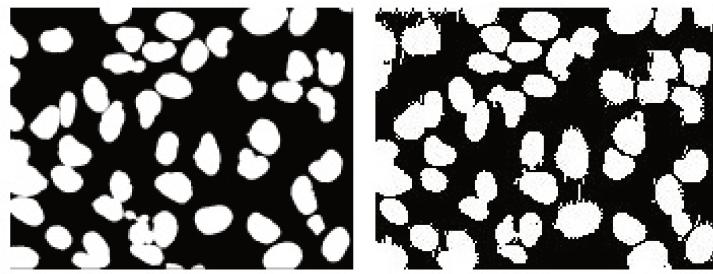


Figura 4.5: Resultado da segmentação utilizando o algoritmo *Algen* para a base de dados do Laboratório *Murphy*

Fonte: elaboração do autor.

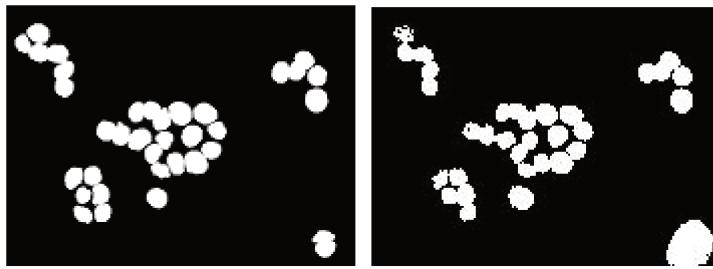


Figura 4.6: Resultado da segmentação utilizando o algoritmo *Algen* para a base de dados da empresa } }American Type Culture Collection (ATCC)"

Fonte: elaboração do autor.

4.3 Formato da Análise Comparativa

Em nosso processo de segmentação, utilizamos uma abordagem semelhante a de Ranefall e Wählby (2016), e assumimos que pequenas variações na segmentação realizadas manualmente para gerar o *ground-truth* interferem muito pouco no resultado e, portanto, essas pequenas diferenças não são significativas para estimar a eficiência da segmentação. Por esse fato, assumimos que as grandes diferenças entre nossos resultados e a segmentação manual realizada pelo laboratório Murphy são os verdadeiros problemas.

Durante a realização de nosso processo de segmentação, não temos qualquer conhecimento prévio do *ground-truth*, e por esse motivo solicitamos como parâmetro de entrada um tamanho estimado dos objetos contidos na imagem. Após completar o processo de segmentação, avaliamos a qualidade dos resultados por meio da medida *F* (*F-Score*), a qual realiza uma comparação pixel a pixel entre o *ground-truth* e a segmentação obtida pelo algoritmo *Algen* e pelo algoritmo *Algat*.

Para computar a medida *F*, os pixels são classificados em uma das seguintes categorias:

True Positives: pixels pertencentes a um objeto na imagem segmentada pelo algoritmo que também pertencem a um objeto no *ground-truth*.

True Negatives: pixels pertencentes ao plano de fundo na imagem segmentada pelo algoritmo que também pertencem ao plano de fundo no *ground-truth*.

False Positives: pixels pertencentes a um objeto na imagem segmentada pelo algoritmo que pertencem ao plano de fundo no *ground-truth*.

False Negatives: pixels pertencentes ao plano de fundo na imagem segmentada pelo algoritmo que pertencem a um objeto no *ground-truth*.

Após realizar a classificação dos pixels, calculamos o *recall* e o *precision* conforme as equações 4.2 e 4.1, respectivamente. O cálculo do *recall* é realizado utilizando a soma de todos os pixels *true positives* dividido pela soma conjunta dos pixels *false negatives* com os pixels *true positives*. O cálculo do *precision* é composto pela soma de todos os pixels *true positives* dividido pela soma conjunta dos pixels *false positives* com os pixels *true positive*.

$$\text{Recall} = \frac{\text{true positives}}{\text{false positives} + \text{true positives}} \quad (4.1)$$

$$\text{Precision} = \frac{\text{true positives}}{\text{false negatives} + \text{true positives}} \quad (4.2)$$

Após computarmos o *recall* e o *precision*, calculamos a medida *F* aplicando a Equação 4.3. A medida *F* aqui apresentada é a média harmônica entre *recall* e *precision*.

$$F = \frac{2 \times (\text{recall} \times \text{precision})}{(\text{recall} + \text{precision})} \quad (4.3)$$

Exemplificamos a importância da medida *F* na Figura 4.7. O exemplo demonstra três resultados do nosso processo de segmentação para a mesma imagem, sendo que cada segmentação possui um valor específico de Medida *F*.

É possível notar que quando a Medida *F* é muito baixa (por exemplo, 40%), os resultados estão longe do desejado. Quando a medida *F* aumenta para valores maiores do que 60%, encontramos mais semelhança com o *ground-truth*.

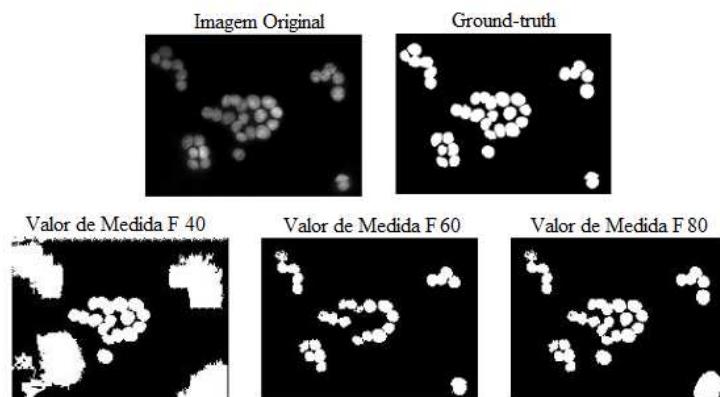


Figura 4.7: Ilustração do resultados de Medidas F
Fonte: elaboração do autor.

A medida F foi aplicada como métrica em mais de 26 trabalhos que também utilizaram a base de dados do laboratório Murphy, por esse motivo, foi possível realizar comparações com os resultados obtidos por outras técnicas. As comparações são expostas nas seções 4.4, 4.5 e 4.6.

4.4 Avaliação da Métrica Almod

Na Seção 4.4, iremos discutir a métrica *Almod*, que possui como característica principal pontuar a qualidade da segmentação realizada. Essa métrica foi usada pelo algoritmo *AlgAl* para selecionar os melhores resultados e também corresponde à *fitness function* do algoritmo *Algen*. A pergunta a ser respondida aqui é: “Segmentações com melhores pontuações são mais próximas do *ground-truth*?” Se a resposta for negativa, então todos os esforços dos dois algoritmos seriam em vão, dado que estaríamos otimizando uma função que não corresponde ao objetivo proposto.

O *Almod* funciona como um classificador de resultados e possui como funcionalidade calcular o módulo da diferença entre a imagem original e a imagem processada, portanto, quanto menor o valor gerado por *Almod* maior deverá ser o resultado da medida F .

Para computar o desempenho da métrica *Almod*, executamos nosso processo de segmentação para todas as 48 imagens originais do laboratório Murphy. A operação foi realizada utilizando o algoritmo *Algen* com uma população de 16 indivíduos ao longo de sete gerações. Como resultado da execução, obtivemos 5.376 segmentações, sendo 112 por imagem original. Na execução, nossa métrica foi capaz de encontrar, em 56% dos casos, a solução com melhor medida F existente. Em 17% dos casos, nossa métrica falhou em

encontrar o melhor resultado, porém por uma diferença menor que 1%. Nos demais casos, 27%, a métrica *Almod* encontrou resultados com diferenças superiores a 2% do melhor resultado. A Figura 4.8 apresenta os percentuais.

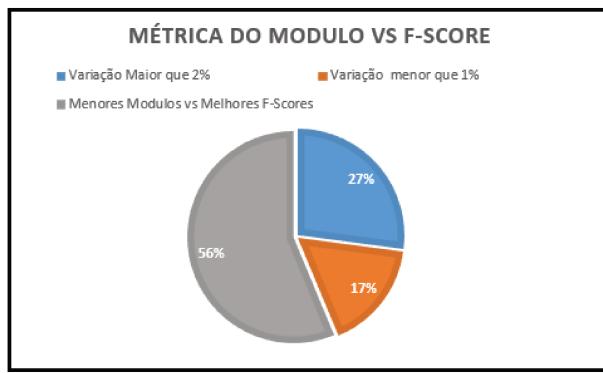


Figura 4.8: Resultados das imagens com menores módulos para uma população de 16 indivíduos ao longo de sete gerações.

Fonte: elaboração do autor.

Continuando com a análise das 5.376 segmentações, realizamos comparações entre três cenários, sendo o primeiro cenário representado pelos melhores valores da medida *F* que obtivemos em nosso processo de segmentação, o segundo cenário representado pelos piores valores da medida *F* que obtivemos em nosso processo de segmentação e o terceiro cenário representa os valores obtidos pela classificação do algoritmo *Almod*.

A diferença entre o *melhor resultado possível* e o *pior resultado possível* foi de 21%, ao passo que a diferença entre o *melhor resultado possível* e o resultado produzido pela métrica *Almod* foi de 2%. A Figura 4.9 representa essa relação entre os cenários.

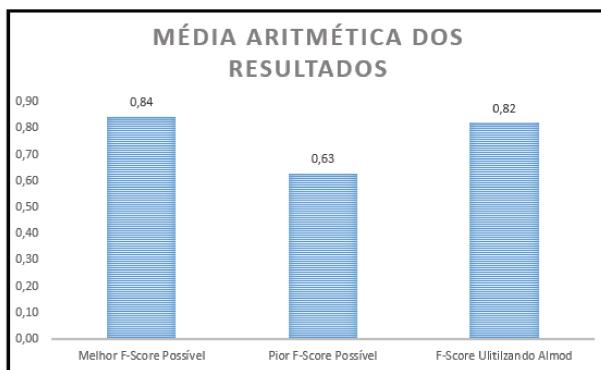


Figura 4.9: Resultado comparativo que apresenta a medida *F* entre os três cenários.

Fonte: elaboração do autor.

Dados os resultados mostrados nas Figuras 4.8 e Figuras 4.9, concluímos que o algoritmo *Almod* fornece uma heurística capaz de guiar os algoritmos em direção à melhor segmentação, apesar de não ser 100% preciso.

4.5 Resultados do *Algen* e *Algai*

Apresentaremos os resultados obtidos pela execução do *Algen* e do *Algai*. Para tornar a comparação justa, limitamos o espaço de busca que as abordagens podem explorar em 112 segmentações e ambas as execuções receberam os mesmos parâmetros de entrada, portanto, não existe vantagem em relação ao número de execuções realizadas ou a parâmetros de entrada.

Na execução do algoritmo *Algai*, 112 iterações para cada imagem original produziu um total de 5.376 segmentações. A mesma quantidade de segmentações foi obtida pelo algoritmo *Algen* operando com uma população de 16 indivíduos ao longo de sete gerações, o que resulta em $16 \times 7 = 112$ resultados por imagem original e 5.376 segmentações exploradas considerando todas as imagens do conjunto. Em ambos os casos, as imagens marcadas como resultado final foram selecionadas com a métrica *Almod*.

O algoritmo *Algai* produziu resultados inferiores aos produzidos pelo algoritmo *Algen*. O algoritmo *Algai* obteve uma medida *F* igual ou superior à 80% em 67% dos resultados (32 imagens das 48). Em 27% dos casos, o valor ficou entre 75% e 79% (13 imagens das 48). Em apenas 6% dos casos, o algoritmo *Algai* obteve um intervalo entre 70% e 74% (3 imagens das 48). A Figura 4.10 ilustra esses resultados.

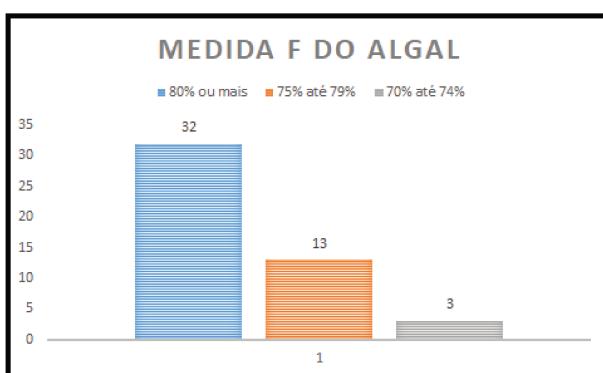


Figura 4.10: Resultados obtidos utilizando o algoritmo *Algai*.
Fonte: elaboração do autor.

O algoritmo *Algen* obteve uma medida F igual ou superior à 80% em 75% dos resultados (36 casos de 48). Em 23% dos resultados, o valor ficou entre 75% e 79% (11 casos de 48). Em apenas 2% dos casos, obteve um intervalo entre 72% e 74%. A Figura 4.11 ilustra esses resultados.

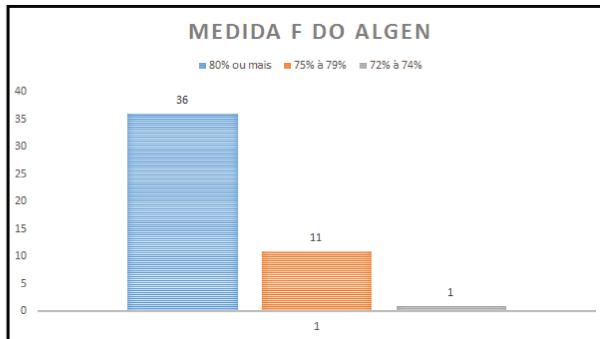


Figura 4.11: Resultados obtidos utilizando o algoritmo *Algen*.

Fonte: elaboração do autor.

4.5.1 Análise por Instâncias do Laboratório Murphy

Durante as execuções de ambas as abordagens (*Algen* e *Algai*), constatamos que o algoritmo *Algen* foi superior ao algoritmo *Algai* na maioria das execuções, o que foi causado pelo refinamento sucessivo dos indivíduos por meio da métrica *Almod* ao longo da execução do *Algen*. Apresentaremos agora um perfil dos melhores resultados considerando as imagens obtidas do Laboratório Murphy.

Para facilitar o entendimento, classificamos os resultados em três níveis: (i) $n1$, resultados que possuem como medida F valores superiores à 80%, (ii) $n2$, resultados cuja medida F está no intervalo de 75% à 79%, e (iii) $n3$, resultados que obtiveram medida F entre 70% e 74%.

Durante a execução, existem pequenas diferenças entre os resultados produzidos pelo *Algen* e pelo *Algai*. O menor resultado produzido pela execução *Algen* foi de 72,5% de medida F , ao passo que na execução do *Algai* o menor resultado foi de 70,5% de medida F . Em relação aos níveis de classificação, o *Algen* apresentou resultados melhores nos níveis $n1$, $n2$ e $n3$ produzindo uma melhora de 8%, 4% e 4%, respectivamente.

Com o intuito de comparar as abordagens (*Algen* e *Algai*), geramos uma visualização gráfica que colore uma determinada região de acordo com a abordagem de maior valor. Para representar a abordagem *Algai*, utilizamos a cor azul e para representar a abordagem *Algen* utilizamos a cor laranja.

A Figura 4.12 apresenta a comparação entre os valores produzidos por ambas as abordagens considerando cada imagem individualmente. A abordagem *Algal* (azul) se mostrou melhor em 10 instâncias de 48, o que representa 21% do total. A abordagem *Algen* apresentou melhores resultados em 38 instâncias de 48, o que torna sua predominância na Figura 4.12 evidente, pois representa 79% do total.

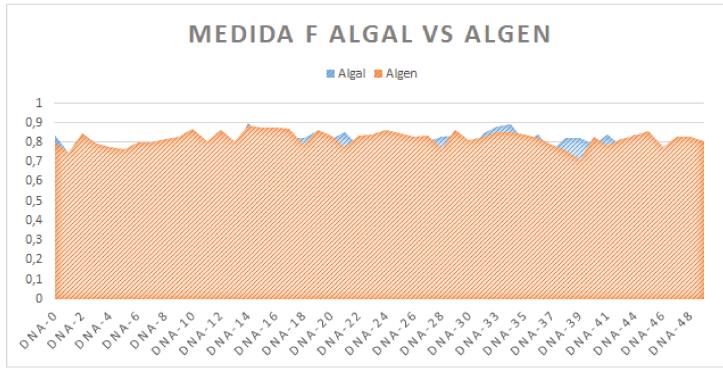


Figura 4.12: Comparação entre os resultados da medida F para os algoritmos *Algen* e *Algal*. O gráfico é colorido com base na técnica que gerou maior valor no eixo Y, o qual representa o valor obtido pela medida F , já o eixo X representa todas as 48 imagens do conjunto fornecido pelo laboratório Murphy.

Fonte: elaboração do autor.

4.5.2 Estabilidade dos Resultados e Tempo de Execução

Em nosso processo de segmentação, assumimos a mesma perspectiva de Ranefall e Wählby (2016), os quais reconhecem que as pequenas diferenças ocasionadas pela linha de contorno entre a segmentação manual (*Ground Thruth*) e os resultados gerados pelos algoritmos possuem menor relevância quando comparado com objetos faltantes ou excedentes. Os objetos faltantes e os objetos excedentes representam os verdadeiros erros no processo de segmentação, ao passo que as linhas de contorno definida pela segmentação manual podem sofrer variações (viés) do quanto perto o contorno é traçado em relação ao objeto.

Nossa principal finalidade é de encontrar todos os objetos da imagem, e apenas os objetos. Após executar o processo de segmentação repetidas vezes, consideramos que o contorno dos objetos podem variar o valor da medida F em até 23%. Portanto, em nosso processo de segmentação para o conjunto de imagens do laboratório Murphy, classificamos como bons resultados acima de 77%.

Um de nossos objetivos consiste em criar um processo de segmentação que seja consistente, independe da imagem processada, pois, segundo Ranefall e Wählby (2016),

muitos dos métodos de segmentação falham em situações diferentes das projetadas inicialmente.

Nosso principal objetivo é encontrar os objetos independente do viés do contorno. Com esse intuito, reduzimos o tamanho das imagens processadas a fim de obtermos múltiplos resultados e encontrar o melhor, de modo a obtermos resultados estáveis com maior frequência. A redução do tamanho da imagem nos possibilitou obter resultados do processo de segmentação, para uma única imagem, em menos de 0,19 segundos.

Com esse tempo de execução, criamos mecanismos que asseguram o bom resultado da segmentação, como por exemplo, a estabilidade dos parâmetros de entrada, o algoritmo de métrica *Almod* e os algoritmos *Algen* e *Algai*.

Calculamos o desempenho do nosso processo de segmentação em um computador contendo 8GB de memória RAM, processador i3-4170 com 3.70GHz, Windows 10 e 1TB de memória secundária. Para essas características, processamos 5.376 segmentações em aproximadamente 17 minutos, ou uma imagem a cada 0,19 segundos.

4.6 Comparação entre Métodos de Segmentação

Na Seção 4.6, apresentamos dois estudos comparativos, sendo o primeiro utilizando a estabilidade dos resultados e o segundo utilizando os resultados obtidos pela medida *F*. Os estudos foram realizados utilizando como base de dados as imagens do Laboratório Murphy e as imagens da empresa }}American Type Culture Collection (ATCC)". Os resultados apresentados ao longo desta seção foram sintetizados, pois, o estudo contou com 27 técnicas, sendo que as medições foram realizadas de maneira individual para cada imagem do conjunto, totalizando 1.296 resultados.

Os resultados desta seção foram calculados a partir da execução conjunta entre os algoritmos *Algen* e *Algai*. Essa abordagem foi adotada devido ao resultado evidenciado na Figura 4.12, a qual permite deduzir que ambos algoritmos se complementam na produção de resultado, e, por esse motivo, incluímos a abordagem de execução conjunta.

4.6.1 Estudo da Volatilidade

Iniciamos o estudo da volatilidade com o conjunto de imagens fornecido pelo Laboratório Murphy. Esse conjunto apresenta alta similaridade entre as imagens, porém em alguns poucos

casos as imagens apresentam diferenças consideráveis na intensidade dos níveis de cinza e na quantidade de células por imagem.

No primeiro estudo, analisamos a amplitude dos resultados (medida F) para cada técnica, dessa forma, o cálculo da amplitude foi realizado individualmente e resume-se na subtração do maior valor obtido pelo menor. Várias técnicas apresentaram uma variação de amplitude média de 56% e, por esse motivo, consideramos os resultados instáveis uma vez que toda medição foi realizada para um único conjunto de imagem que possui alta similaridade. Na Figura 4.13, apresentamos a variação entre mínimo e máximo de cada técnica.

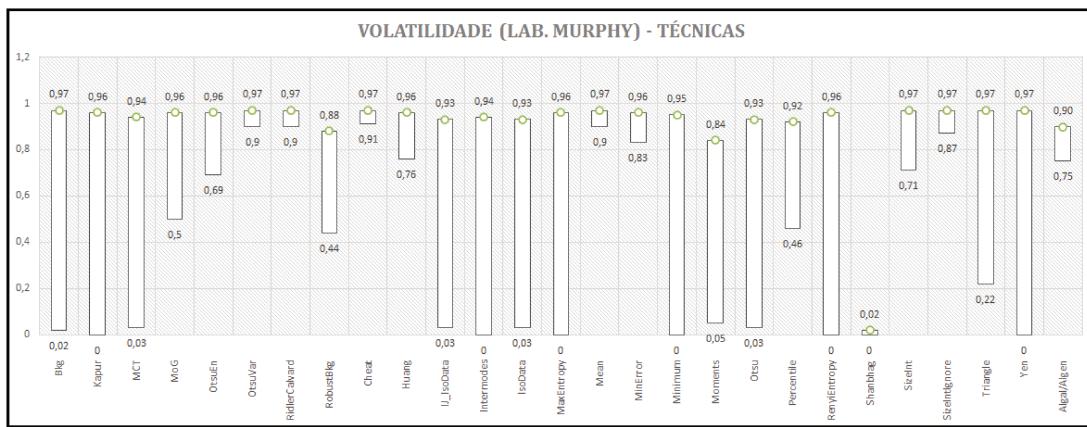


Figura 4.13: Volatilidade dos resultados.

Fonte: elaboração do autor.

Em seguida, comparamos o resultado médio obtido pelas técnicas com o resultado obtido pela execução conjunta dos algoritmos *Algal* e *Algen*. A Figura 4.14 apresenta a comparação da amplitude entre nossa técnica com a média aritmética das demais técnicas.

O mesmo estudo foi realizado para o conjunto de imagens da empresa ATCC. Esse conjunto possui um total de seis imagens e apresenta variação alta na intensidade dos níveis de cinza. Em razão da baixa quantidade de imagens, a medição da volatilidade foi prejudicada. Neste cenário as técnicas apresentaram uma redução da volatilidade nos resultados, tanto como média de variação 21%. A Figura 4.15 apresenta a volatilidade de cada técnica.

Nossa abordagem produziu uma volatilidade alta para o conjunto da empresa ATCC. Em nossas execuções, a volatilidade média para nossa abordagem foi de aproximadamente 24%, ao passo que a volatilidade média (média aritmética) para as demais técnicas foi de aproximadamente 21%. A Figura 4.16 apresenta a comparação da volatilidade entre nossa técnica e a média aritmética das demais técnicas.

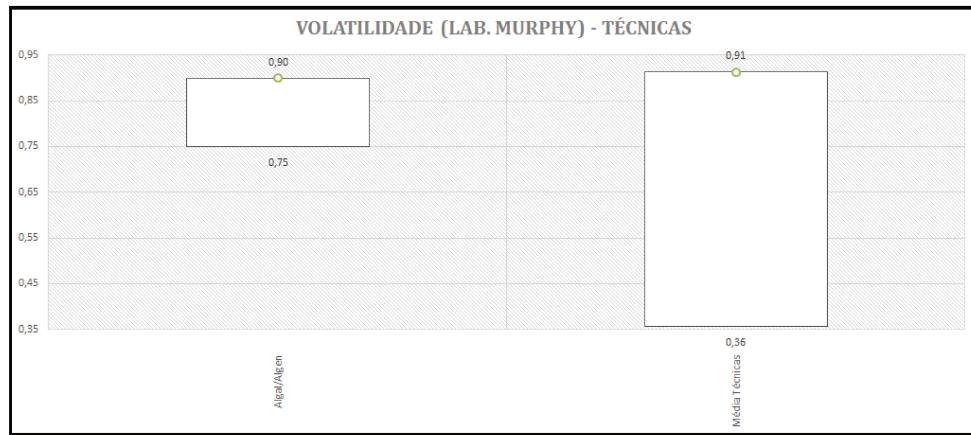


Figura 4.14: Volatilidade dos resultados obtidos por nossa técnica em comparação com a volatilidade média das demais técnicas para o conjunto de dados do Laboratório Murphy.

Fonte: elaboração do autor.

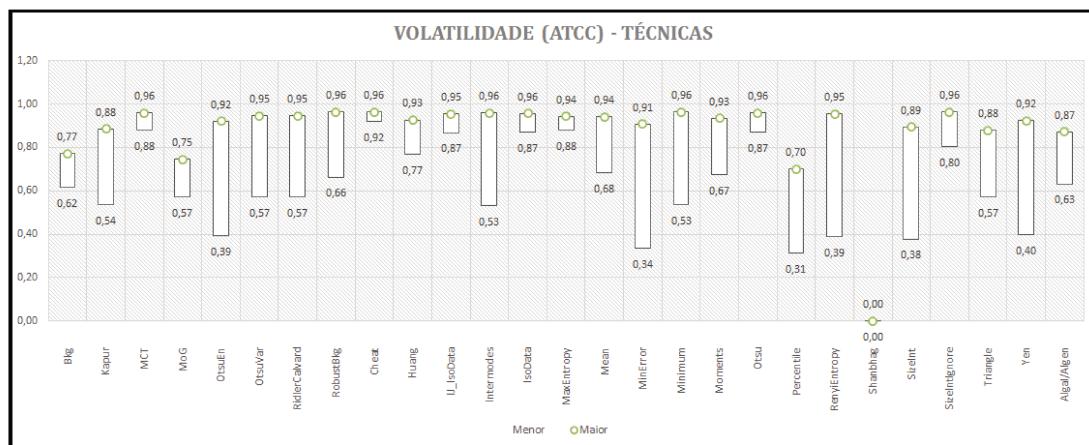


Figura 4.15: Volatilidade das técnicas utilizando o conjunto de dados da empresa ATCC.

Fonte: elaboração do autor.

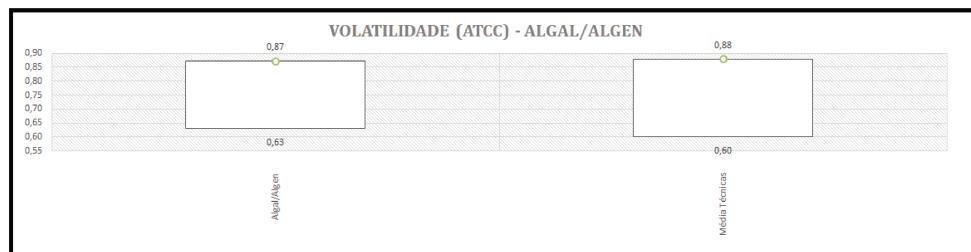


Figura 4.16: Volatilidade dos resultados obtidos por nossa técnica em comparação com a volatilidade média das demais técnicas para o conjunto de dados da empresa ATCC.

Fonte: elaboração do autor.

Por fim, disponibilizamos as informações utilizadas para construir o estudo da volatilidade. Referidas informações estão em formato de tabela e podem ser consultadas na Figura 4.17

VOLATILIDADE CONJUNTO LAB. MURPHY				VOLATILIDADE CONJUNTO ATCC			
	Menor Resultado	Média	Melhor Resultado		Menor Resultado	Média	Melhor Resultado
Bkg	0,02	0,7	0,97	Bkg	0,62	0,70	0,77
Kapur	0	0,88	0,96	Kapur	0,54	0,70	0,88
MCT	0,03	0,88	0,94	MCT	0,88	0,92	0,96
MoG	0,5	0,69	0,96	MoG	0,57	0,69	0,75
OtsuEn	0,69	0,92	0,96	OtsuEn	0,39	0,58	0,92
OtsuVar	0,9	0,95	0,97	OtsuVar	0,57	0,75	0,95
RidlerCalvard	0,9	0,95	0,97	RidlerCalvard	0,57	0,75	0,95
RobustBkg	0,44	0,66	0,88	RobustBkg	0,66	0,90	0,96
Cheat	0,91	0,95	0,97	Cheat	0,92	0,95	0,96
Huang	0,76	0,93	0,96	Huang	0,77	0,90	0,93
IJ IsoData	0,03	0,87	0,93	IJ IsoData	0,87	0,91	0,95
Intermodes	0	0,53	0,94	Intermodes	0,53	0,91	0,96
IsoData	0,03	0,87	0,93	IsoData	0,87	0,92	0,96
MaxEntropy	0	0,09	0,96	MaxEntropy	0,88	0,92	0,94
Mean	0,9	0,94	0,97	Mean	0,68	0,82	0,94
MinError	0,83	0,91	0,96	MinError	0,34	0,54	0,91
Minimum	0	0,53	0,95	Minimum	0,53	0,91	0,96
Moments	0,05	0,67	0,84	Moments	0,67	0,88	0,93
Otsu	0,03	0,87	0,93	Otsu	0,87	0,92	0,96
Percentile	0,46	0,7	0,92	Percentile	0,31	0,52	0,70
RenyiEntropy	0	0,39	0,96	RenyiEntropy	0,39	0,87	0,95
Shanbhag	0	0	0,02	Shanbhag	0,00	0,00	0,00
SizeInt	0,71	0,89	0,97	SizeInt	0,38	0,62	0,89
SizeIgnore	0,87	0,93	0,97	SizeIgnore	0,80	0,93	0,96
Triangle	0,22	0,88	0,97	Triangle	0,57	0,75	0,88
Yen	0	0,4	0,97	Yen	0,40	0,85	0,92
Algal/Algen	0,75	0,86	0,90	Algal/Algen	0,63	0,72	0,87

Figura 4.17: Tabulação dos dados de volatilidade para o conjunto de dados do Lab. Murphy e da empresa ATCC.

Fonte: elaboração do autor.

4.6.2 Estudo dos Resultados da Medida F

Nesta sub-seção, apresentamos os resultados dos desempenhos das técnicas de modo separado para cada conjunto de imagens. Os resultados foram separados em três grupos: (i) no primeiro grupo, levamos em consideração apenas os valores da execução do algoritmo *Algal*, (ii) no segundo grupo, levamos em consideração apenas os resultados produzidos pelo algoritmo *Algen* e (iii) no terceiro grupo, levamos em consideração a execução conjunta entre *Algen* e *Algal*.

Na primeira série de experimentos (execução do *Algal*) obtivemos, para 94% das instâncias, valores de medida F superiores a 75% e uma média geral de 80%, sendo que o pior resultado obtido foi de 70,5% e o melhor resultado com valor de 90,0%. Para a segunda série de experimentos (execução do *Algen*), obtivemos, para 98% das instâncias, valores de medida F superiores a 75% e uma média geral de 84%, sendo que o pior resultado obtido foi de 72,5%

e o melhor resultado obtido com valor de 89,0%. Na terceira série de experimentos (execução conjunta do *Algal* e *Algen* utilizando *Almod* como comparativo), obtivemos, para 100% das instâncias, valores de medida *F* superiores a 75% e uma média geral de 86%, sendo que o pior resultado obtido foi de 75% e o melhor resultado com valor de 90,0%. A Figura 4.18 apresenta um gráfico de linha com os resultados das três séries que apresentamos.



Figura 4.18: Demonstração dos resultados obtidos por meio da execução do algoritmo *Algal*, *Algen* e da execução conjunta entre o *Algal* e *Algen*.

Fonte: elaboração do autor.

No conjunto de dados fornecido pelo Laboratório Murphy, em 96% das instâncias os valores obtidos estiveram acima de 60%, com uma média geral em 73%, sendo que o pior resultado obtido foi de 44% e o melhor resultado com valor de 90%. A Figura 4.19 apresenta um gráfico de linha com resultado individual para cada imagem e comparmos com o nosso resultado.

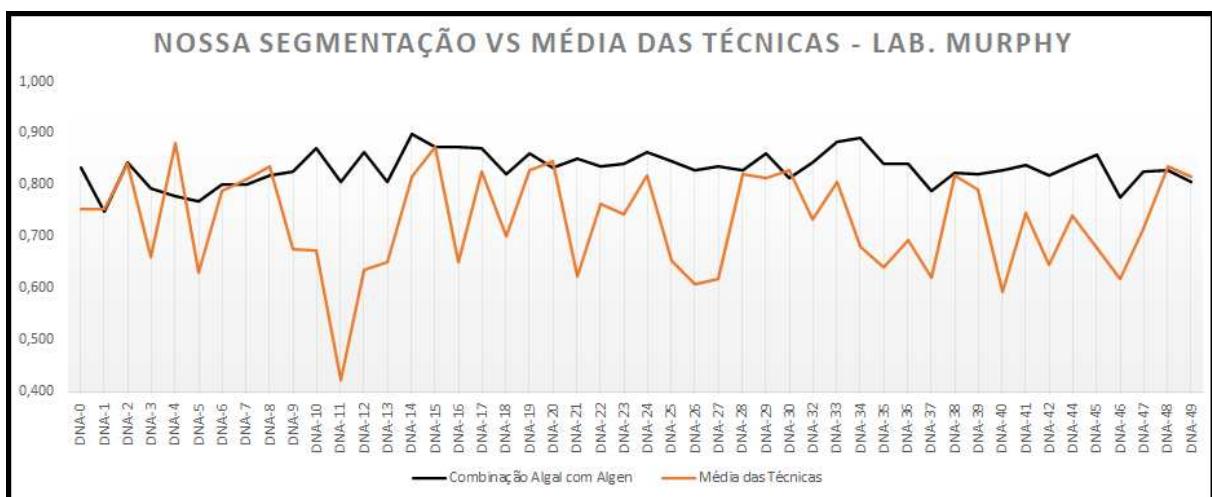


Figura 4.19: Demonstração da média dos resultados obtidos pelas técnicas em comparação com os resultados obtidos por nossa segmentação.

Fonte: elaboração do autor.

Para o conjunto de imagens fornecido pela empresa ATCC, realizamos o mesmo procedimento e obtivemos três conjuntos de resultados. O primeiro resultado foi levando em consideração apenas os valores da execução do algoritmo *Algal*. O segundo resultado levou em consideração apenas os resultados produzidos pelo algoritmo *Algen*. O terceiro resultado foi originado a partir da execução conjunta entre *Algen* e *Algal*.

Para a primeira série de experimentos (execução do *Algal*), obtivemos para 67 das instâncias valores de medida *F* superiores a 65% e uma média geral de 69%, sendo que o pior resultado obtido foi de 62,5% e o melhor resultado com valor de 87,5%. Para a segunda série de experimentos (execução do *Algen*), obtivemos para 83% dos valores de medida *F* superiores a 65% e uma média geral de 72%, sendo que o pior resultado obtido foi de 63% e o melhor resultado com valor de 88%. Para a terceira série de experimentos (execução conjunta do *Algal* e *Algen* utilizando *Almod* como comparativo), obtivemos para 100% dos valores de medida *F* superiores a 65% e uma média geral de 73,5%, sendo que o pior resultado obtido foi de 66% e o melhor resultado com valor de 88%. A Figura 4.20 apresenta um gráfico de linha com os resultados das três séries que apresentamos.

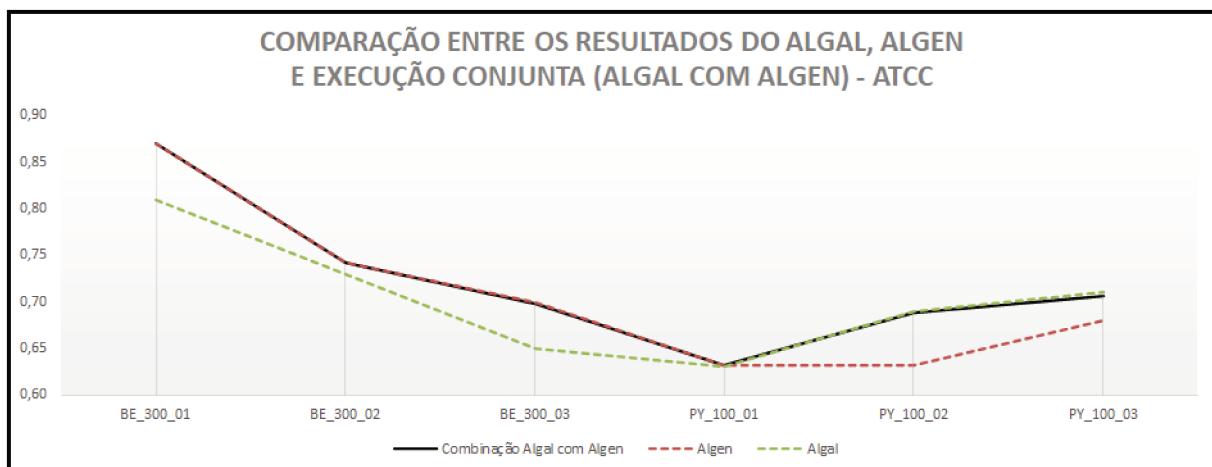


Figura 4.20: Demonstração dos resultados obtidos por meio da execução do algoritmo *Algal*, *Algen* e da execução conjunta entre o *Algal* e *Algen*.

Fonte: elaboração do autor.

No conjunto de dados fornecido pela empresa ATCC, as técnicas obtiveram, em 100% das instâncias, valores acima de 70% com uma média geral de 74%, sendo que o pior resultado foi de 72% e o melhor resultado com valor de 77%. A Figura 4.21 apresenta um gráfico de linha com resultado individual para cada imagem em comparação com o resultado obtido por nossa segmentação.

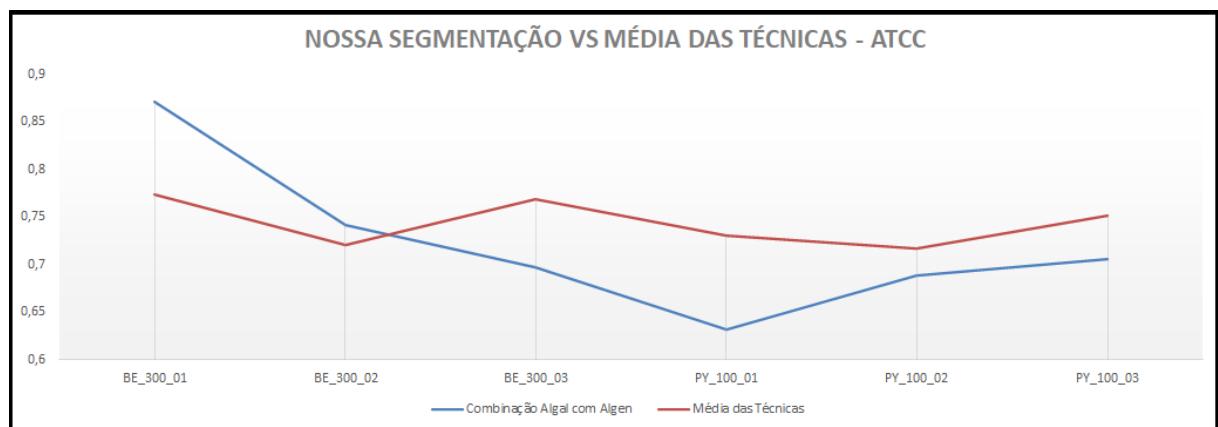


Figura 4.21: Demonstração da média dos resultados obtidos pelas técnicas em comparação com os resultados obtidos por nossa segmentação.

Fonte: elaboração do autor.

Capítulo 5

Considerações Finais

Neste trabalho, apresentamos a fundamentação, o desenvolvimento e os resultados de um novo processo de segmentação. O processo abrange uma etapa de pré-processamento, seguido da segmentação e representação de forma hierárquica, sendo que toda a estrutura está envolta por algoritmo genético que auxilia todas as etapas e garante melhores resultados.

A etapa de pré-processamento foi criada para tratamento dos ruídos e das distorções. Esta etapa é iniciada com a utilização do *gaussian blur* para suavizar os objetos da imagem e após a suavização geramos o gradiente morfológico (diferença entre dilatação e erosão). A aplicação da suavização reduziu a quantidade de objetos não identificados e trouxe ganhos na medida F para os conjuntos que utilizamos.

Após a etapa de pré-processamento iniciamos a segmentação por meio da transformadas Watershed hierárquica e convertemos o resultado obtido para a estrutura da Árvore dos Lagos Críticos, que possui representação hierárquica dos valores e possibilita a correção de prováveis erros com base nas métricas definidas. Tais métricas representam uma conjunção de medidas baseadas no tamanho dos objetos e na curvatura elíptica, que proporcionam melhores resultados quando aplicadas em conjunto. Atrelados à estrutura hierárquica da *ALC*, proporcionam maior ênfase no ajuste e na correção de erros, de modo a aprimorar o processo de segmentação.

Por fim, elaboramos uma estrutura iterativa baseada em algoritmo genético e modelos para avaliação de resultados preliminares. Esta estrutura possui como objetivo o ajuste contínuo dos parâmetros e permite ao algoritmo genético identificar objetos faltantes e objetos excedente, resultando na redução das falhas no processo de reconhecimento dos objetos.

Os princípios norteadores deste trabalho foram o tempo de processamento e a preocupação em encontrar todos os objetos, e apenas estes. Como não estávamos interessados em acertar todos os pixels dos contornos, decidimos segmentar imagens em escalas reduzidas, o que pode ter resultado em uma definição não tão precisa dos contornos com consequente penalidade na medida F utilizada como métrica de qualidade.

Nosso trabalho contribui com soluções promissoras para a segmentação de imagens e esclarece algumas dúvidas da comunidade científica, contudo, deixamos como trabalho futuro a implementação de novas características na identificação dos objetos, tais como a utilização do gradiente, da textura e de outras formas geométricas. Também sugerimos como aprimoramento a detecção de sobreposição de objetos, pois nossa técnica apresenta dificuldades em detectar sobreposição de área de interesse e por esse motivo seria interessante aprimorar os resultados a partir de detectores de bordas e detectores de sobreposição de objetos. Por fim, ressaltamos que nossa técnica foi aplicada única e exclusivamente para bases de dados de imagens de células e, portanto, existe uma restrição referente ao contexto que será segmentado. Acreditamos que referida restrição pode ser solucionada a partir do aprimoramento do trabalho com novas características e funcionalidades.

Referências bibliográficas

- BEASLEY, D.; BULL, D. R.; MARTIN, R. R. An overview of Genetic Algorithms: Pt1, Fundamentals. In:
- BEUCHER, S. Watershed, Hierarchical Segmentation and Waterfall Algorithm. In: **Mathematical Morphology and Its Applications to Image Processing**. Edição: Jean Serra e Pierre Soille. Dordrecht: Springer Netherlands, 1994. p. 69–76. ISBN 978-94-011-1040-2. DOI: 10 . 1007 / 978 - 94 - 011 - 1040 - 2 _ 10. Disponível em: <https://doi.org/10.1007/978-94-011-1040-2_10>.
- BRADLEY, D.; ROTH, G. Adaptive Thresholding using the Integral Image. **J. Graphics Tools**, v. 12, p. 13–21, jan. 2007. DOI: 10 . 1080 / 2151237X.2007 . 10129236.
- BURGER, W.; JAMES BURGE, M. **Principles of Digital Image Processing: Fundamental Techniques**. [S.l.: s.n.], jan. 2008.
- CARVALHO, M. A. G. **Análise hierárquica de imagens através da árvore dos lagos críticos**. 2004. Tese (Doutorado) – Unicamp.
- COELHO, L. P.; SHARIFF, A.; MURPHY, R. F. Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. [S.l.: s.n.], jun. 2009. p. 518–521. DOI: 10 . 1109 / ISBI . 2009 . 5193098.
- COELHO, L. P.; SHARIFF, A.; MURPHY, R. F. Nuclear segmentation in microscope cell images: A hand-segmented dataset and comparison of algorithms. **2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro**, p. 518–521, 2009.
- DARWIN, C. **On the Origin of Species by Means of Natural Selection**. London: Murray, 1859. or the Preservation of Favored Races in the Struggle for Life.
- DENG, Y.; LIU, Y.; ZHOU, D. An Improved Genetic Algorithm with Initial Population Strategy for Symmetric TSP. **Mathematical Problems in Engineering**, v. 2015, p. 1–6, out. 2015. DOI: 10 . 1155 / 2015 / 212794.
- FALCÃO, A. X.; STOLFI, J.; ALENCAR LOTUFO, R. de. The Image Foresting Transform: Theory, Algorithms, and Applications. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, Washington, DC, USA, v. 26, n. 1, p. 19–29, jan. 2004. ISSN 0162-8828. DOI: 10 . 1109 / TPAMI . 2004 . 1261076. Disponível em: <<https://doi.org/10.1109/TPAMI.2004.1261076>>.

- LI, C. et al. A Level Set Method for Image Segmentation in the Presence of Intensity Inhomogeneities With Application to MRI. **IEEE Transactions on Image Processing**, v. 20, n. 7, p. 2007–2016, 2011. DOI: 10.1109/TIP.2011.2146190.
- LI, Y.; FAN, X.; LI, G. Image Segmentation based on Tsallis-entropy and Renyi-entropy and Their Comparison. In: 2006 4th IEEE International Conference on Industrial Informatics. [S.l.: s.n.], 2006. p. 943–948. DOI: 10.1109/INDIN.2006.275704.
- LINDBLAD, J.; BENGTSSON, E. A comparison of methods for estimation of intensity nonuniformities in 2D and 3D microscope images of fluorescence stained cells, jan. 2001.
- MANDUCHI, R. Mixture models and the segmentation of multimodal textures. v. 1, 98–104 vol.1, fev. 2000. DOI: 10.1109/CVPR.2000.855805.
- OTSU, N. A Threshold Selection Method from Gray-Level Histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, 1979.
- PANAGIOTAKIS, C.; ARGYROS, A. Region-based Fitting of Overlapping Ellipses and its application to cells segmentation. **Image and Vision Computing**, Elsevier, v. 93, p. 103810, 2020.
- PEDRINI, H.; SCHWARTZ, W. Análise de imagens digitais: princípios, algoritmos e aplicações. **THOMSON PIONEIRA**, 9, 15, 20, 22, 24, 29 e 42, 2008. ISSN 0162-8828. Disponível em: <<http://books.google.com.br/books?id=13KAPgAACAAJ>>.
- RANE FALL, P.; SADANANDAN, S. K.; WÄHLBY, C. Fast adaptive local thresholding based on ellipse fit. In: 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI). [S.l.: s.n.], abr. 2016. p. 205–208. DOI: 10.1109/ISBI.2016.7493245.
- RANE FALL, P.; WÄHLBY, C. Global gray-level thresholding based on object size. v. 89, p. 385–390, abr. 2016.
- SINGH, A. et al. An Abandoned Object Detection System Based on Dual Background Segmentation, p. 352–357, 2009. DOI: 10.1109/AVSS.2009.74.
- SUN, Y. et al. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. In: PROCEEDINGS. International Conference on Image Processing. [S.l.: s.n.], 2002. v. 3, 825–828 vol.3. DOI: 10.1109/ICIP.2002.1039099.
- TAO, W.; JIN, H.; ZHANG, Y. Color Image Segmentation Based on Mean Shift and Normalized Cuts. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, v. 37, n. 5, p. 1382–1389, 2007. DOI: 10.1109/TSMCB.2007.902249.
- XU, Y.; UBERBACHER, E. C. 2D image segmentation using minimum spanning trees. **Image and Vision Computing**, v. 15, n. 1, p. 47–57, 1997. ISSN 0262-8856. DOI: [https://doi.org/10.1016/S0262-8856\(96\)01105-5](https://doi.org/10.1016/S0262-8856(96)01105-5). Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0262885696011055>>.
- EL-ZAART, A. Images thresholding using ISODATA technique with gamma distribution. **Pattern Recognition and Image Analysis**, v. 20, p. 29–41, mar. 2010. DOI: 10.1134/S1054661810010037.

ZAITOUN, N. M.; AQEL, M. J. Survey on Image Segmentation Techniques. **Procedia Computer Science**, v. 65, p. 797–806, 2015. International Conference on Communications, management, and Information technology (ICCMIT'2015). ISSN 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2015.09.027>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050915028574>>.

ZHU, W.; LIANG, S.; WEI, Y.; SUN, J. Saliency Optimization from Robust Background Detection. In: THE IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], jun. 2014.