UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Faculdade de Ciências e Tecnologia
Câmpus de Presidente Prudente

# Análise de Características em Imagens Digitais na Tarefa de Segmentação Interativa

Thiago Amorim de Faccio

Orientador: Prof. Dr. Wallace Correa de Oliveira Casaca

Programa: Matemática Aplicada e Computacional

Presidente Prudente, Junho de 2020

**UNIVERSIDADE ESTADUAL PAULISTA**

Faculdade de Ciências e Tecnologia de Presidente Prudente

Programa de Pós-Graduação em Matemática Aplicada e Computacional

# Análise de Características em Imagens Digitais na Tarefa de Segmentação Interativa

Thiago Amorim de Faccio

Orientador: Prof. Dr. Wallace Correa de Oliveira Casaca

Dissertação apresentada ao Programa de Pós-Graduação em Matemática Aplicada e Computacional da Faculdade de Ciências e Tecnologia da UNESP para obtenção do título de Mestre em Matemática Aplicada e Computacional.

Presidente Prudente, Junho de 2020

F138a

Faccio, Thiago Amorim de
Análise de características em imagens digitais na tarefa de segmentação interativa / Thiago Amorim de Faccio. -- , 2020
77 p. : tabs., fotos

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências Farmacêuticas, Araraquara,
Orientador: Wallace Correa de Oliveira Casaca

1. Processamento de imagens Técnicas digitais. 2. Imagens digitais. 3. Visão por computador. 4. Modelos matemáticos. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências Farmacêuticas, Araraquara. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

**UNIVERSIDADE ESTADUAL PAULISTA**

**Câmpus de Presidente Prudente**

## CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Análise de Características em Imagens Digitais na Tarefa de Segmentação Interativa

**AUTOR: THIAGO AMORIM DE FACCIO**

**ORIENTADOR: WALLACE CORREA DE OLIVEIRA CASACA**

Aprovado como parte das exigências para obtenção do Título de Mestre em MATEMÁTICA APLICADA E COMPUTACIONAL, pela Comissão Examinadora:

Prof. Dr. WALLACE CORREA DE OLIVEIRA CASACA Campus Experimental de Rosana / UNESP

Prof. Dr. JOSÉ ROBERTO NOGUEIRA **(PARTICIPAÇÃO VIA VIDEOCONFERÊNCIA)**
Departamento de Matemática e Computação / Faculdade de Ciências e Tecnologia de Presidente Prudente

Prof. Dr. HARLEN COSTA BATAGELO **(PARTICIPAÇÃO VIA VIDEOCONFERÊNCIA)**
Centro de Matemática, Computação e Cognição / Universidade Federal do ABC

Presidente Prudente, 15 de junho de 2020

*A meus pais, meus irmãos e meu amor,*
*dedico este trabalho.*

# Agradecimentos

Aqui venho para agradecer a todas as pessoas que de alguma forma ajudaram, direta ou indiretamente, durante a grande jornada que foi a realização deste trabalho. Concluir uma caminhada como essa é tão satisfatório quanto foi trabalhoso, e só foi possível graças a grande força que recebi de pessoas incríveis.

Antes de qualquer coisa, gostaria de agradecer aos meus pais, Adnir e Roseli, por todo suporte que prestaram, estando ao meu lado ou à distância, nunca me faltou amor e incentivo por parte deles, devido a isso me considero privilegiado.

Ao meu amor, Maria Carolina, que dividiu comigo todos os dias, desde o início da graduação até o fim do mestrado, essa experiência maravilhosa de estudos e pesquisas na matemática, não só como minha parceira dentro de casa, nos momentos de descontração, mas também como minha companheira de estudos.

Ao meu Orientador, Prof$^{o}$ Dr. Wallace Casaca, por toda atenção e disposição em me ajudar, respondendo aos chamados de socorro independente do dia ou hora, assim como a sua parceira Prof$^{a}$ Dra. Marilaine Colnago, que participou em diversos momentos deste trabalho.

Aos meus amigos mais próximos, aqueles que carrego desde a infância, e também aqueles que tive a oportunidade de conhecer já na fase adulta, que de alguma forma sustentaram minha sanidade com momentos de descontração e distração em meio a exaustivas horas de estudos e preocupações.

Aos meus familiares, por suas recorrentes festas e farras que ajudam a melhorar o humor de qualquer pessoa que esteja presente, e claro, por toda ajuda quando foi necessário.

Aos meus amigos de trabalho e graduação, que sempre que possível quebraram algum galho para que eu pudesse ganhar tempo para tocar meus estudos e desenvolver projetos paralelos.

E finalmente, aos meus amigos do Cabritas, que desde muito antes do início desta jornada veem proporcionando momentos bem humorados e carregados de renovação, uma galera que não mede esforços para vencer a distância e se reunir como um time, seja no mundo real, ou no mundo digital... Vai Cabritas!

Além destes, gostaria também de agradecer a Universidade Virtual do Estado de São Paulo (Univesp), pela oportunidade de exercer a função de facilitador de ensino, o que agregou na minha experiência como profissional e me proporcionou apoio financeiro.

*“ You can do anything you set your mind to”.*
***Benjamin Franklin***

# Resumo

A tarefa de segmentação de imagem está presente em diversos programas e aplicativos ao alcance de qualquer usuário de tecnologia. Apesar disso, pouco se fala sobre o funcionamento desta ferramenta, isto é, como ela é realmente executada quando levado em consideração não simplesmente o ponto de vista gráfico, mas também o processo matemático e computacional por trás desse tipo de tarefa. Visto isso, este trabalho faz uma breve abordagem sobre a matemática e as metodologias que regem técnicas de Processamento Digital de Imagens, com foco em métodos de Segmentação Interativa, e busca por meio do estudo de diferentes métodos clássicos, alcançar melhorias nos resultados deste tipo de tarefa. Tem-se como objetivo obter maior precisão na realização de segmentações através da análise de novas características extraídas da imagem, as quais são adotadas como elementos de entrada para o cálculo dos pesos do grafo associado à segmentação do método das Coordenadas de Laplace. A proposta analisou diferentes composições de *features* (bandas) de realce dos objetos e bordas de uma imagem, sendo essas bandas resultantes de técnicas como, por exemplo, modelos de mistura gaussiana, o algoritmo *expectation–maximization*, detector de bordas *Canny*, funções de distância local, entre outras. Assim, almejou-se atingir melhores resultados tanto do ponto de vista quantitativo, a partir de métricas de aferição de partições tais como o *Rand Index* (RI), *Variation ou Information* (VoI) e o *Dice Coefficient* (DICE), como também em termos qualitativos, seja para imagens com poucas marcações de entrada fornecidas pelo usuário ou com riqueza das mesmas.

Palavras-Chave: *Segmentação Interativa de Imagens, Análise de Features, Método das Coordenadas de Laplace, Processamento Digital de Imagens, Detector de Bordas.*

# Abstract

Image segmentation is an omnipresent task in many programs and applications commonly used by technology users. Despite its systematic use, the segmentation task has several issues not properly exploited in terms of the mathematical background, i.e., how the segmentation works not only in the visual interpretation aspect, but also considering the mathematical and computational processes behind this kind of task. Therefore, this manuscript takes a brief look at mathematics and the methodologies that govern Digital Image Processing techniques, mainly on Interactive Image Segmentation methods, and seeks through the study of different classic methods achieve improvements in the results of this type of task. Our goal is to reach high-precision segmentations by analysing new sets of features extracted from the image, which are taken as input for the weighted graph associated to the Laplacian Coordinates segmentation approach. The study investigated different combinations of features which highlight the objects and edges of the image. These features are computed by means of image processing techniques such as Gaussian mixture models, EM algorithm, Canny edge detector, local distance functions, among others. Thus, we aim to achieve better segmentation results both from a quantitative perspective, based on segmentation metrics such as Rand Index (RI), Variation or Information (VoI) and Dice Coefficient (DICE), as well as from qualitative view, including images with a reduced number of annotations.

Keywords: *Interactive Image Segmentation, Features Analysis, Laplacian Coordinates, Digital Image Processing, Edge Detector.*

# Lista de Figuras

# Lista de Tabelas

# Sumário

Capítulo

1

# Introdução

Caracteriza-se como processamento de imagens, o tratamento de dados a partir da realização de operações em imagens digitais. Essas imagens podem ser submetidas a processos matemáticos os quais permitem a manipulação das informações contidas nas mesmas, informações estas que estão descritas na Seção 1.1, possibilitando assim a criação de técnicas como, a segmentação de imagens, ou ainda, processos de aprendizagem de máquina ou de reconhecimento de padrões.

A datar de seus primeiros usos no início do Século XX, técnicas de Processamento Digital de Imagens (PDI) evoluíram passando pelo surgimento dos primeiros grandes computadores até a era digital, tendo se tornado cada vez mais versáteis e comuns no dia-a-dia de pessoas em todo o mundo.

Hoje, programas com grande capacidade de tratamento de imagens estão cada vez mais próximos dos usuários, como em computadores pessoais e celulares, tendo ainda amplo protagonismo nas redes sociais por meio da utilização de filtros e *apps* de edição de fotografias. Entretanto, não é só para uso em redes e meios de disseminação da informação que se sustentam as pesquisas baseadas em PDI. Tal campo de estudo tem oferecido soluções para diversas outras aplicações, em áreas como, por exemplo, medicina, auxiliando no estudo detalhado de ressonâncias (Bao e Chung [8]), ou ainda, na contagem de glóbulos brancos em amostras de sangue (Liu e colaboradores [42]).

Há, também, exemplos representativos em áreas como da segurança e monitoramento, onde há inúmeras aplicações como análise de imagens de satélites (Negri e colaboradores [50]) e até mesmo em frentes como geologia, arqueologia (Lin e colaboradores [41]), entre outras mais.

Em alguns casos, graças a programas com grande capacidade de processamento de imagens, é possível tornar o meio digital disponível a quem pensou que nunca teria a chance de utilizá-lo. Por exemplo, no trabalho de (Dias [26]), este utiliza uma câmera com o auxílio de um programa de PDI focado em um ponto azul, posicionado na testa do usuário, o qual segmenta cada imagem capturada e, em seguida, determina as coordenadas do cursor do mouse para que um usuário possa movê-lo pela tela do computador, possibilitando assim o uso da máquina para pessoas com lesões na medula.

No caso da aplicação de segmentação de imagens, esta busca promove a subdivisão da imagem em agrupamentos de pixels (*clusters*), que se relacionam através de diferentes características, especialmente a intensidade e o gradiente, com a intenção de aperfeiçoar

o reconhecimento da informação visual na imagem para a interpretação tanto por um agente humano quanto para fins de aprendizado de máquina.

## 1.1 Representação Digital de uma Imagem

Quando falamos em manipular as informações presentes em uma imagem digital a partir do uso de técnicas como a de segmentação, estamos, na verdade nos referindo aos dados que são interpretados pelo computador na hora de armazenar ou reproduzir tais imagens.

Antes de entender como é representada uma imagem na tela de dispositivos digitais, temos que entender primeiramente como é interpretado um pixel pelo computador. Um pixel é o menor ponto de uma imagem, e este possui apenas uma cor, sendo o conjunto de vários pixels responsável por compor uma imagem completa. Se estamos falando de pixels em tons de cinza, para o computador, esta informação não passa de um número, um número que representa a intensidade da cor presente no pixel em questão.

Em uma interpretação de uma imagem de 8 *bits*, na qual cada pixel corresponde a um *byte*, os pixels em tons de cinza podem ser representados por números inteiros que variam de 0 (ausência de cor) até 255 (intensidade máxima de cor). Neste caso, um pixel representado pelo valor 0 indica a cor preta, que representa a ausência de cor, enquanto um pixel na máxima intensidade, 255, é da cor branca. Todos os demais valores representam tons intermediários de cinza, em 256 variedades, incluindo o preto e o branco, como representado na Figura 1.1.

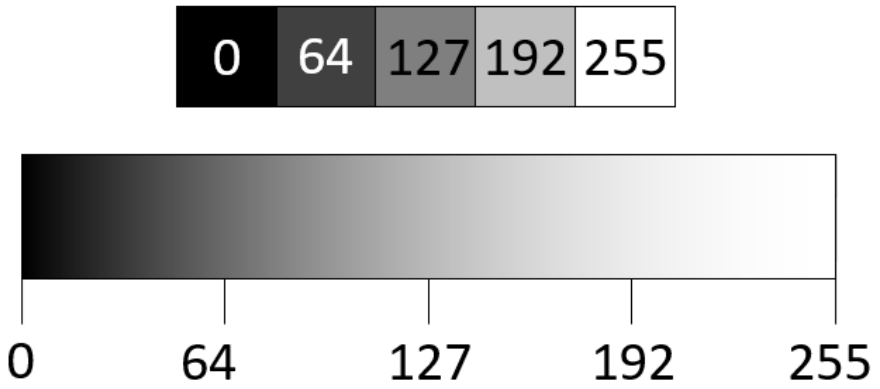
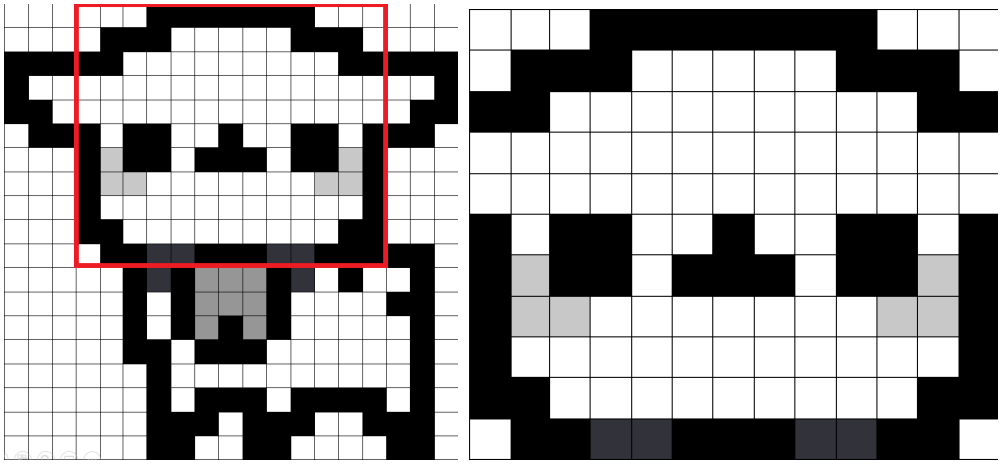



Figura 1.1: Representação digital da intensidade dos pixels de uma imagem.

A partir da concepção de pixel, podemos pensar na representação de uma imagem digital. Assim, uma imagem é composta de um conjunto de pixels, ou seja, sua representação é dada por diversos valores escalares posicionados um ao lado do outro, compondo assim uma matriz, onde cada elemento desta matriz representa a intensidade de um pixel que compõe a imagem. Para melhor ilustração deste conceito, vide a Figura 1.2 e a representação digital da área destacada, na Figura 1.3.

O mesmo é válido quando consideramos uma fotografia em tons de cinza, a qual é, na verdade, uma imagem digital, conforme destacado na Figura 1.4.

Com relação às dimensões de uma imagem digital, está é sempre representada em termos da largura×altura, ou ainda, pixels na horizontal×pixels na vertical. Por exemplo, um vídeo nada mais é do que uma sequência de imagens, reproduzidas sistematicamente



(a) Imagem completa. (b) Área destacada.

Figura 1.2: Imagem em tons de cinza com pixels de larga escala.

$$\begin{pmatrix}
255 & 255 & 255 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 255 & 255 & 255 \\
255 & 0 & 0 & 0 & 255 & 255 & 255 & 255 & 255 & 0 & 0 & 0 & 255 \\
0 & 0 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 0 & 0 \\
255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\
255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\
0 & 255 & 0 & 0 & 255 & 255 & 0 & 255 & 255 & 0 & 0 & 255 & 0 \\
0 & 200 & 0 & 0 & 255 & 0 & 0 & 0 & 255 & 0 & 0 & 200 & 0 \\
0 & 200 & 200 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 200 & 200 & 0 \\
0 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 0 \\
0 & 0 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 & 0 & 0 \\
255 & 0 & 0 & 50 & 50 & 0 & 0 & 0 & 50 & 50 & 0 & 0 & 255
\end{pmatrix}$$

Figura 1.3: Matriz que representa a área destacada da Figura 1.2.

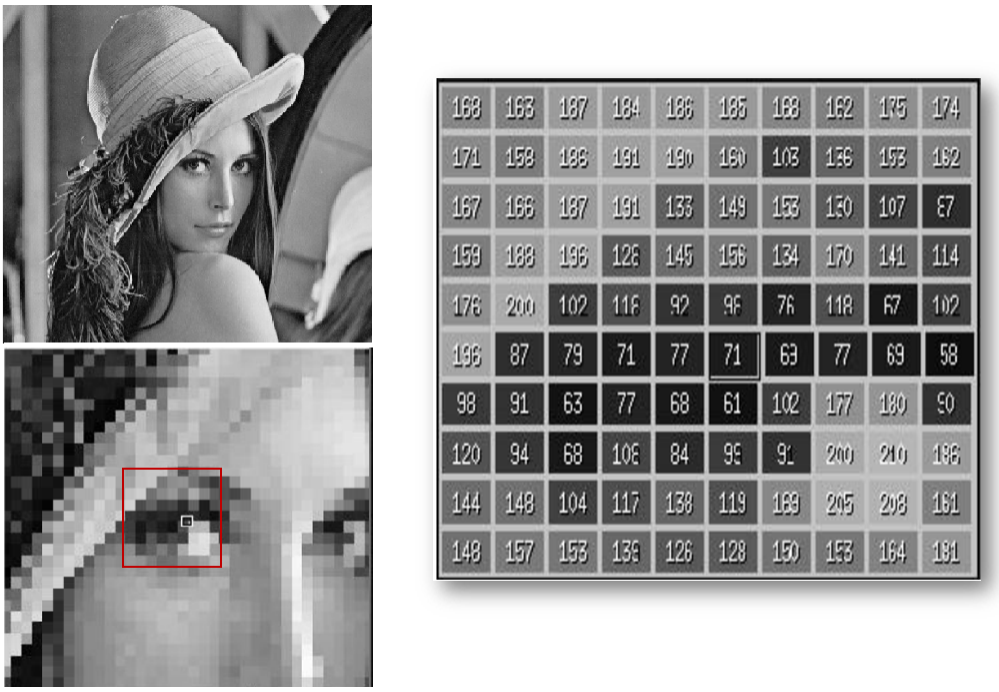



Figura 1.4: Representação digital 10×10 da área destacada pelo quadrado vermelho na fotografia digital.

na tela de um dispositivo digital. Neste caso, cada uma dessas imagens recebe o nome de quadro, ou ainda, *frame*. Uma resolução de vídeo considerada de alta definição (*Full* HD), e popularmente representada em sites de videos por 1080p, possui em cada quadro uma imagem de resolução 1920×1080, ou seja, cada imagem que compõe o vídeo possui 1920 pixels de largura e 1080 pixels de altura.

Acabamos de relacionar imagens com matrizes, e quando falamos de dimensões de matrizes, esta é dada em termos de linhas×colunas, de forma contrária a representação da resolução de uma imagem. Uma imagem no formato 1080, como dito anteriormente, tem resolução de 1920×1080, e se considerarmos essa imagem em tons de cinza – para facilitar a analogia – esta possui então um total de 2.073.600 pixels. Logo, para representá-la em sua forma matricial, será necessário uma matriz com o mesmo número de elementos. Por outro lado, como a dimensão da matriz é dada por linhas×colunas, a matriz que representa uma imagem de resolução 1920×1080 pixels, possui na verdade dimensões 1080×1920. A imagem representada na Figura 1.2 (b), por exemplo, possui resolução 13×11 pixels, enquanto a matriz que a representa, disposta na Figura 1.3, possui dimensões 11×13.

No caso de uma imagem colorida, a representação desta é bastante semelhante ao caso de tons de cinza, contudo, deve ser levado em consideração algum sistema de representação de cores. No caso mais comum, a imagem é definida a partir do chamado sistema RGB (*Red*, *Green*, *Blue*) de cores, em que cada pixel não é representado apenas por um escalar, mas sim por um vetor de três elementos, no qual cada termo representa a intensidade de luz vermelha, verde e azul, respectivamente, presentes neste pixel. Logo, a matriz que representa uma imagem colorida possui três dimensões. Por exemplo, considerando, mais uma vez, a resolução 1080, teríamos uma matriz de dimensões 1080×1920×3, na qual a terceira dimensão faz alusão às 3 bandas que compõem a imagem, a saber: cor vermelha, cor verde, e a terceira na cor azul que, quando combinadas, formam uma imagem colorida, como é descrito com mais detalhes na Seção 3.2.1.

## 1.2 Segmentação de Imagens Digitais

Grande parte dos trabalhos relacionados com a segmentação tem como objetivo a delimitação de objetos ou regiões de uma figura, através de métodos que fazem a leitura da imagem e buscam por pontos de fronteiras/bordas, possibilitando assim a extração de objetos/alvos da imagem a partir dos limites de tais objetos.

Quando os pontos de fronteira de uma imagem são devidamente delineados, a imagem fica subdividida em agrupamentos de pixels, os quais recebem o nome de partições ou *clusters*. Com relação à literatura de segmentação de imagens, podemos dividir as técnicas existentes em duas categorias básicas:

- Técnicas que promovem *clusters* binários (Figura 1.5).
- Técnicas que promovem múltiplas partições - *multi-clusters* (Figura 1.6).

O processo de segmentação da imagem pode ser realizado de duas maneiras distintas: de forma automática, e a partir da participação de um usuário, a qual facilita a obtenção do resultado esperado.

Na abordagem automática, o método é estruturado para interpretar a imagem e criar agrupamentos de acordo com a semelhança entre os pixels. Em razão disso, tais métodos são chamados de técnicas de segmentação automática. Já na proposta que envolve

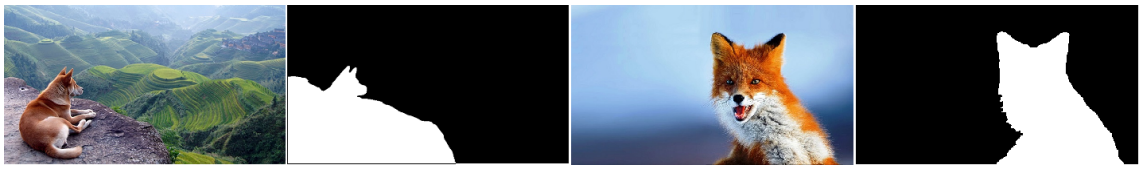


Figura 1.5: Segmentação binária de uma imagem. Da esquerda para a direita tem-se: imagem 1, clusterização da imagem 1, imagem 2, clusterização da imagem 2.

Figura 1.6: Separação da imagem em diversos agrupamentos de pixels, levando em consideração suas características como intensidade, cor, gradiente, entre outras (Imagem adaptada de Casaca [17]).

a participação do usuário, isto é, os chamados métodos de segmentação interativa (Capítulo 2), a classificação dos pixels em *clusters* acontece também a partir das características dos pixels que compõem a imagem, no entanto, a segmentação é orientada a partir dos pixels demarcados pelo usuário, os quais auxiliam no processo de segmentação das áreas da imagem que devem ou não ser segmentadas.

## 1.3 Objetivos

A pesquisa conduzida nessa dissertação retrata a tentativa de melhorar a precisão na segmentação de imagens utilizando diferentes técnicas de PDI, cada qual produzindo uma *feature* (banda da imagem), a qual é fornecida como um parâmetro para o cálculo de uma matriz de pesos, utilizada no processo de segmentação da imagem. Tal matriz faz parte do processo de solução de um determinado segmentador, o qual emprega um funcional de energia de forma a obter um sistema linear que, quando solucionado, retorna um mapa escalar da imagem. Por fim, a partir do mapa gerado, a imagem é então binarizada de modo a produzir a segmentação desejada. Além disso, como objetivo secundário, procurou-se aumentar a precisão da segmentação de forma a evitar um aumento significativo do custo computacional do processo de segmentação como um todo. O processo descrito acima está ilustrado na Figura 1.7, e será explorado com um maior nível de detalhamento durante este trabalho.

Nesta pesquisa, foi explorado o processo de segmentação interativa guiada por *seeds* (*brushes*/pinceladas), que são fornecidas por um usuário a fim de guiar o segmentador na tarefa de clusterização de imagens pretendidas. Essas *seeds* constituem a "personificação" do interesse do usuário com relação aos objetos presentes na imagem. Assim, no caso da metodologia aqui empregada, as *seeds* se diferenciam em termos de cores: a cor verde representa pixels dentro dos objetos de interesse da imagem, enquanto que a cor vermelha define uma porção de pixels presentes no fundo da imagem, ou a área de descarte. Em

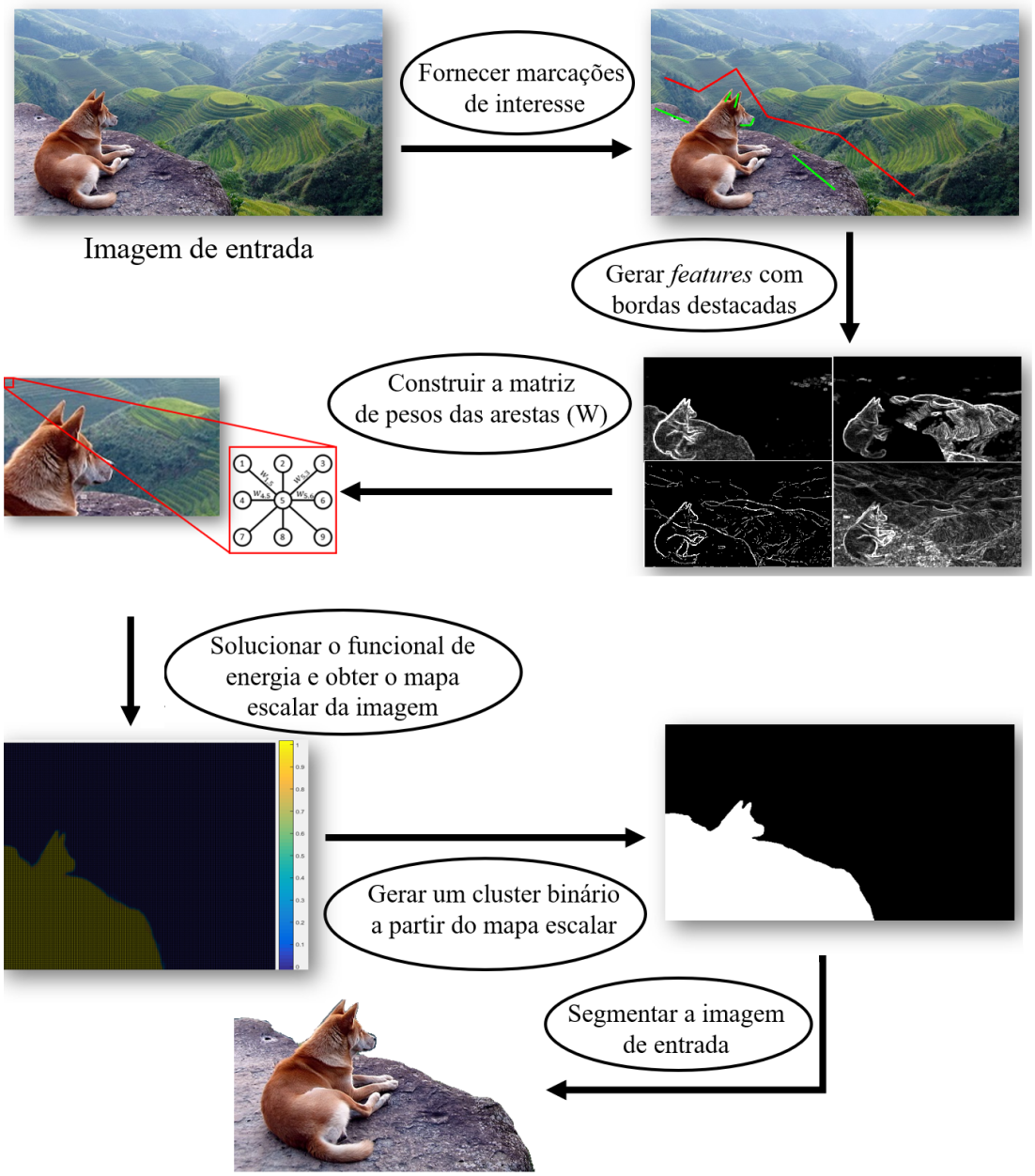
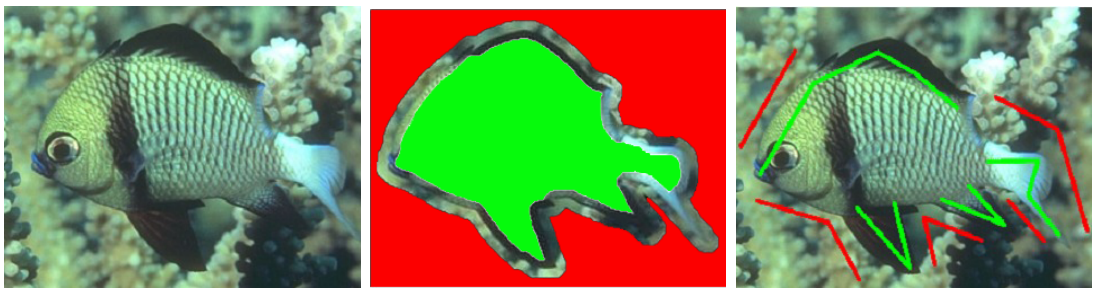



Figura 1.7: Processo de segmentação realizado neste trabalho.



(a) Imagem de entrada (b) *Seeds* densas (c) *Seeds* esparsas

Figura 1.8: Modelos de *seeds*.

particular, neste trabalho, foi adotado o método de segmentação das *Coordenadas de Laplace* (Casaca e colaboradores [20, 19]) como protótipo de teste.

Desta forma, a análise conduzida nesta pesquisa permite aperfeiçoar a tarefa de segmentação interativa, melhorando assim a acurácia dos resultados esperados. Além disso, a pesquisa empregou conjuntos de *seeds* esparsas, isto é, marcações com menor grau de distribuição na imagem (um exemplo deste tipo de *seed* é apresentado na figura 1.8), o que viabiliza a utilização do segmentador em circunstâncias reais de uso, como, por exemplo, a partir de mecanismos que possibilitem o "deslizar de um dedo" sobre a tela de um celular, assim como marcações rápidas feitas por um mouse em um computador.

Portanto, a ideia implementada neste trabalho pretende combinar um *pipeline* de segmentação robusto e de fácil manuseio por parte dos usuários, e que permita a obtenção de resultados mais precisos com relação ao estado-da-arte. Assim, a combinação de técnicas de PDI tais como aquelas avaliadas nesta pesquisa possibilitou atingir uma segmentação de alta acurácia e com custo computacional aceitável, que neste trabalho, procuramos sempre manter abaixo de 10 segundos por imagem. Além disso, a segmentação obtida procura refletir os anseios do usuário, visto que a imagem é particionada em "Objeto" (ou "*Foreground*"), e "Fundo" (ou "*Background*") conforme as marcações definidas pelo mesmo. Sendo assim, a segmentação ocorre de forma interativa e objetiva, gerando resultados de alta precisão (vide Figura 1.9, por exemplo).

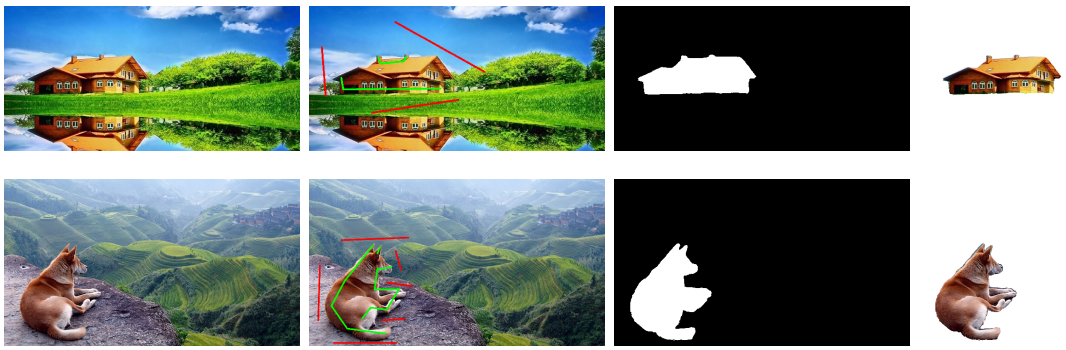


Figura 1.9: Segmentação com o uso de *seeds* fornecidas pelo usuário. Da esquerda para a direita tem-se: imagem original, *seeds* fornecidas, divisão da imagem em *clusters*, resultado da segmentação.

Capítulo

2

# Segmentação Interativa

## 2.1 Métodos Clássicos de Segmentação Interativa

Conforme mencionado anteriormente, são denotados por Métodos de Segmentação Interativa aqueles que contam com a participação do usuário na tarefa de segmentação de uma imagem, isto é, no processo de dividi-la em agrupamentos de pixels. Esse processo é guiado a partir do interesse do usuário em um determinado objeto ou partição de uma imagem, que é conduzido por meio de marcações (*seeds*) fornecidas pelo mesmo. Uma discussão desse tipo de aplicação pode ser vista no trabalho de Yuvaraj e colaboradores [65].

Existem duas abordagens principais nesse tipo de segmentação dirigida: a primeira delas é uma abordagem baseada na fronteira dos objetos, e a outra é direcionada a partir das regiões que compõem as imagens. Em ambos os casos, a provisão das *seeds* por parte do usuário acontece, entretanto, de maneira distinta. Em abordagens do tipo de fronteira, o usuário normalmente fornece marcações de interesse em um objeto, e estas se espalham pela imagem como em um processo de crescimento até que atinjam os limites/bordas deste objeto. Este processo é conhecido na literatura como *Seeded Region Growing*. No segundo caso, que são as abordagens baseadas em regiões, a ideia do *Seeded Region Growing* também é aplicada, porém, o usuário deve fornecer previamente as *seeds* não só para o objeto de interesse, mas também para a região que o contorna, ou seja, as marcações funcionam como rótulos tanto para o objeto como para o fundo da imagem.

Em linhas gerais, tais métodos empregam as marcações de entrada para construir um grafo de afinidade, que faz corresponder cada pixel da imagem a um nó, e trata cada conexão entre dois nós como uma aresta, geralmente de peso positivo, cujo valor depende da intensidade, da cor, do gradiente, vizinhanças, ou ainda de outros fatores do par de pixels em questão.

Antes da ideia principal a ser explorada neste trabalho ser introduzida, será descrito alguns dos métodos da literatura baseados em *seeds*, os quais realizam a segmentação de uma imagem a partir da propagação dessas marcações e dos modelos matemáticos associados a cada método.

Desta forma, focaremos nos métodos que tomam como premissa matemática a minimização de funcionais de energia em grafos de afinidade, a exemplos daqueles categorizados e discutidos por Couprie e colaboradores [21]. De fato, esses autores demonstraram

que grande parte das técnicas clássicas de segmentação interativa tais como *Graph Cuts* (Boykov e Funka-Lea [63], Boykov e Jolly [13]), *Randon Walker* (Grady e Funka-Lea [36], Grady e Sinop [35]), *Shortest Paths* (Bai e Sapiro [4]), ou ainda, *Watersheds* (Monteiro e Campinho [49], Cousty e colaboradores [22]), podem ser formulados a partir de um único funcional de energia mais genérico. Mais especificamente, o seguinte funcional de energia contendo pequenas alterações nas variáveis $p$ e $q$ foi apresentado por Couprie e colaboradores [21]:

$$E_{PWS}(x) = \sum_{i \in B} w_{B_i}^p |x_i - x_B|^q + \sum_{i \in F} w_{F_i}^p |x_i - x_F|^q + \frac{1}{2} \sum_{i \in V} \left( \sum_{j \in N(i)} w_{ij}^p |x_i - x_j|^q \right) , \quad (2.1)$$

em que a solução de $E_{PWS}(x)$ retorna o vetor real $x = (x_i)$ de forma que cada valor $x_i$ está associado a um nó do grafo de afinidade derivado da imagem, gerando assim um mapa escalar para a mesma. Na equação (2.1), $V$ representa o conjunto dos vértices do grafo, que estão relacionados com os pixels da imagem a ser segmentada, $x_B$ e $x_F$ são os valores de referência associados aos pixels semeados pelo usuário, sendo $B$ e $F$ os conjuntos de pixels das *seeds* de fundo (*Background*) e do objeto (*Foreground*), respectivamente, $w_B$ e $w_F$ seus termos regularizadores, $w_{ij}$ é o peso relacionado à aresta que liga o par de nós $(i, j)$, enquanto $N(i)$ representa as arestas $(i, j)$ contidas na vizinhança do nó $i$ em questão.

Para um melhor entendimento de cada um dos métodos supracitados, uma breve descrição de cada técnica será feita nas próximas seções de forma a relacionar cada uma delas com a variação dos parâmetros $p$ e $q$ da equação (2.1).

### 2.1.1 Graph Cuts

Esta abordagem é baseada no clássico método de clusterização *Graph Cut*, apresentado originalmente por (Boykov e Jolly [13]), cujo trabalho foi reproduzido inúmeras vezes em outras pesquisas como, por exemplo, (Boykov e Funka-Lea [63], Hower e colaboradores [37], Rother e colaboradores [55], Vicente e colaboradores [61]), entre outras. Este método, assim como nos outros casos que veremos a seguir, recai na minimização de um funcional de energia que, como mostra Couprie e colaboradores [21], assume a equação (2.1) quando os parâmetros $p$ e $q$ se comportam de forma que $p$ é uma constante finita não nula, e $q = 1$.

Definimos, neste caso, o funcional de energia do *graph cuts*, $E_{GC}$, construído com base na imagem de entrada e nas informações oferecidas pelo usuário:

- O grafo de afinidade $G = (V, E, W_E)$, construído a partir da imagem de entrada.

  Os nós $i \in V$ representando os pixels da imagem.

  As arestas $(i, j) \in E$ relacionando os pares de pixels que fazem parte de uma mesma vizinhança.

  Os pesos $w_{ij} \in W_E$ referentes a cada aresta do grafo.

- *Seeds* fornecidas pelo usuário.

  Marcações sobre o objeto de interesse ($F$).

  Marcações sobre o que se considera como fundo da imagem ($B$).

Assim, deve-se então minimizar o seguinte funcional de energia:

$$E_{GC}(x) = \sum_{i \in F \cup B} D_i(x_i) + \lambda \sum_{(i,j) \in E} V_{i,j}(x_i, x_j)\,, \tag{2.2}$$

onde $D_i$ representa os dados fornecidos nas regiões de interesse da imagem, $x_i$ é um escalar relacionado a um pixel da imagem, $V_{i,j}$ um termo regularizador (ou, ainda, termo de suavização), e $\lambda$ um parâmetro para ajustes do método.

De uma forma simplificada, podemos afirmar que $D_i$ verifica se os valores escalares relacionados aos pixels $x_i$ estão próximos dos pixels $x_F$ e $x_B$ de um dos conjuntos $F$ ou $B$, respectivamente, isto é, das *seeds* inicialmente fornecidas pelo usuário, enquanto $V_{i,j}$ é uma função do argumento $|x_i - x_j|$, que calcula o módulo da diferença dos valores de dois pixels da imagem. Aliando esses conceitos aos pesos $w_{ij}$, é possível reescrever o funcional a ser minimizado na equação (2.2) da seguinte maneira:

$$E_{GC}(x) = \sum_{i \in F} |x_i - x_F| + \sum_{i \in B} |x_i - x_B| + \sum_{(i,j) \in E} w_{ij} |x_i - x_j|. \tag{2.3}$$

A solução do funcional acima resulta no menor custo para a tarefa de corte mínimo com relação aos pesos das arestas do grafo (Juan e Boykov [39]). Em termos efetivos, a solução é obtida a partir de dois algoritmos tradicionais da área: o *Min-Cut* e o *Max-flow*. O processo de solução que combina esses dois algoritmos ficou conhecido na literatura como *Min-cut/Max-flow algorithm*. Tal abordagem encontra-se descrita em detalhes em (Boykov e Kolmogorov [14]).

Servindo de base para diversos outros trabalhos, a obra introdutória de Boykov e Jully [13] foi refinada por diversos outros autores, inclusive com a participação de alguns colaboradores que atuaram diretamente com Boykov em pesquisas sobre o tema, como foi o caso do trabalho de Rother e colaboradores [55], que sugeriram uma segmentador utilizando um aperfeiçoamento do *Graph Cut*. Mais especificamente, tal metodologia considerou uma área retangular a qual limitava o objeto-alvo na imagem (vide Figura 2.1 para uma ilustração).

(a) (b)

Figura 2.1: Segmentação limitada a uma área retangular tal como proposta por Rother e colaboradores [55] via *Graph Cut*. (a) Representação da área que deve ser segmentada com relação ao restante da imagem. (b) Resultado da segmentação aplicada à região selecionada.

### 2.1.2 Random Walks

Mais uma vez, levando em consideração os parâmetros $p$ e $q$ no funcional de energia descrito na equação (2.1), se o valor de $p$ for igual a 1, e adotando, nessa ocasião, $q = 2$, então o funcional (2.1) irá assumir a forma de outro método de segmentação interativo: o algoritmo *Random Walker* (RW). Assim, o modelo RW assuma a seguinte forma (2.4):

$$E_{RW}(x) = \sum_{i \in F}(x_i - x_F)^2 + \sum_{i \in B}(x_i - x_B)^2 + \sum_{(i,j) \in E} w_{ij}(x_i - x_j)^2. \tag{2.4}$$

Como o próprio nome em inglês sugere, o *Random Walker* considera os vértices do grafo de afinidade $G = (V, E, W_E)$, ou seja, os pontos $i \in V$ que representam os pixels da imagem de entrada, como se um caminhante fosse, de forma aleatória, tentar prosseguir até um dos pontos $i \in F$ ou $i \in B$ das *seeds* fornecidas pelo usuário, definindo assim a rotulação desse "caminhante" a partir da probabilidade dele atingir algum desses vértices.

A segmentação através do RW foi introduzida inicialmente por Grady e Funka-Lea [36], e tinha como principal propósito a segmentação de imagens na área da medicina, mais precisamente, no estudo de ressonâncias. Este trabalho foi estendido e aperfeiçoado alguns anos depois por Grady [33] e, desde então, tem sido utilizado com frequência na área de segmentação interativa, conforme demonstrado em (Grady e Sinop [35], Andrews e colaboradores [3], Bao e Chung [8], Cui e colaboradores [23]).

O objetivo desse método era melhorar alguns aspectos na tarefa de segmentação com relação a métodos interativos anteriores, principalmente, com relação ao desempenho computacional. Assim, como cálculos computacionais mais rápidos, o método foi capaz de realizar uma segmentação arbitrária a partir da interação do usuário de forma mais efetiva, além de também prover ferramentas mais intuitivas que maximizassem a interação com o usuário.

Considere o conjunto $V$ de nós do grafo sobre uma malha quadriculada tal como apresentado na Figura 2.2. Vamos então considerar que exista uma probabilidade relacionada ao peso $w_{ij}$ atribuído a uma aresta de uma partícula que se move de um pixel até o outro pela aresta $(i, j)$. Por se tratar de probabilidades, temos que a soma das probabilidades da partícula se mover em todas as direções a partir de cada pixel deve ser 1, e ainda que as arestas que possuírem um ponto de fronteira do objeto tem sua probabilidade diminuída drasticamente, dificultando assim a passagem desta partícula através dessa "parede" ilusória. Dadas as informações de entrada que definem as rotulações do objeto-alvo e do fundo da imagem pelo usuário, cada pixel $i \in V$ não rotulado será avaliado como uma partícula que se desloca nesse mapa livremente e aleatoriamente, tentado alcançar um dos pixels rotulados como representado na Figura 2.2.

Dessa forma, tem-se então calculada a probabilidade da partícula originada de um pixel não rotulado atingir um pixel rotulado. Sejam $x_F$ o valor do rótulo para o conjunto de *seeds* $F$, pertencentes ao objeto, e $x_B$ o valor do rótulo para o conjunto de *seeds* $B$, determinadas como fundo da imagem. Assim, se denotarmos $P_F(i)$ e $P_B(i)$ as probabilidades dessa partícula originada de $i$ atingir um pixel rotulado como objeto ou fundo respectivamente, então $i$ será classificado como:

$$i \longrightarrow \begin{cases} F, & \text{se } P_F(i) \geq P_B(i) \\ B, & \text{caso contrário} \end{cases}. \tag{2.5}$$

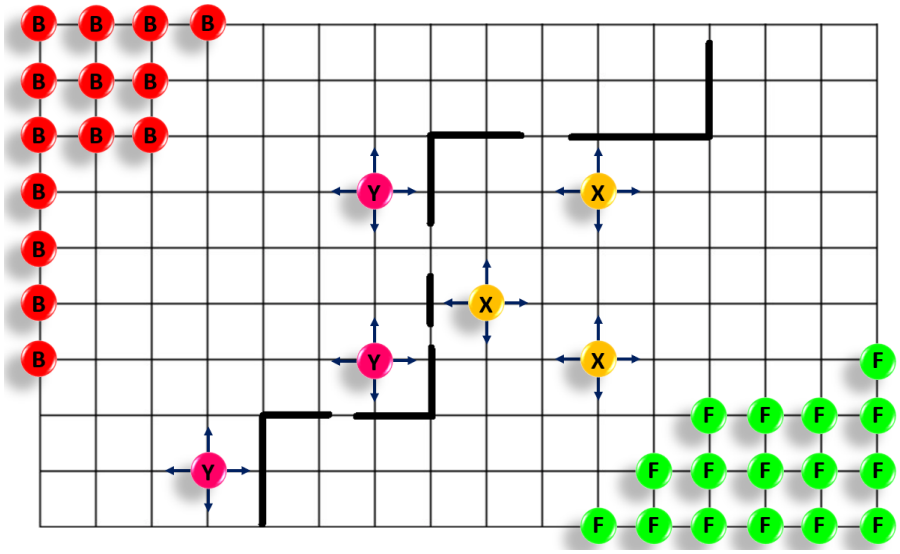



Figura 2.2: Representação do mapa de pontos sobre uma malha quadriculada do processo de caminhada de uma partícula no método *Random Walker*.

Observando a Figura 2.2, é possível ter um *insight* visual de que o método pode "preservar-se" de uma situação em que a fronteira é falha, ou ainda que possua imperfeições. Note que, mesmo um ponto rotulado na imagem como $Y$ podendo se mover livremente para todos os lados, ele tem uma fronteira que impossibilita a passagem do mesmo para o lado direito da imagem, onde estão os rótulos do objeto $(F)$, a chance dele se conectar a um ponto rotulado como $(B)$ continua sendo maior. Logo, o mais provável que aconteça na Figura 2.2 é que os pontos à direita da fronteira se tornem parte do objeto, e os pontos à esquerda da fronteira sejam considerados como parte do fundo da imagem (Figura 2.3). Tal resultado leva à rotulação ilustrada na Figura 2.4.


Figura 2.3: Atribuição esperada de rótulos aos pontos $Y$ e $X$ da Figura 2.2 levando em consideração as bordas encontradas na imagem.

Com relação ao funcional representado na equação (2.4), se considerarmos que $x_i = x_F$ se $i \in F$, e de forma análoga, $x_i = x_B$ se $i \in B$, e sem perda de generalidade assumir que $x_F = 1$ e $x_B = 0$, podemos então determinar $\overline{F}$ e $\overline{B}$ duas matrizes diagonais com elementos $(i, i) = 1$ para os $i's$ inicialmente rotulados como $F$, e $B$ cada um em sua respectiva matriz $\overline{F}$ e $\overline{B}$. Desta forma, o funcional (2.4) pode ser então reescrito na forma matricial através da seguinte expressão:

$$E_{RW}(x) = x^T L x + (1 - x)^T \overline{F}(1 - x) + x^T \overline{B} x, \tag{2.6}$$

em que se adota uma matriz $D$ diagonal cujos valores são representados pelas valências de cada pixel $i \in V$, e outra matriz $W$ esparsa, construída a partir dos pesos $w_{ij}$ relacionados às arestas $(i, j)$. Note que, neste caso, podemos escrever $L = D - W$, que é a matriz Laplaciana do grafo a qual aparece na equação (2.6).

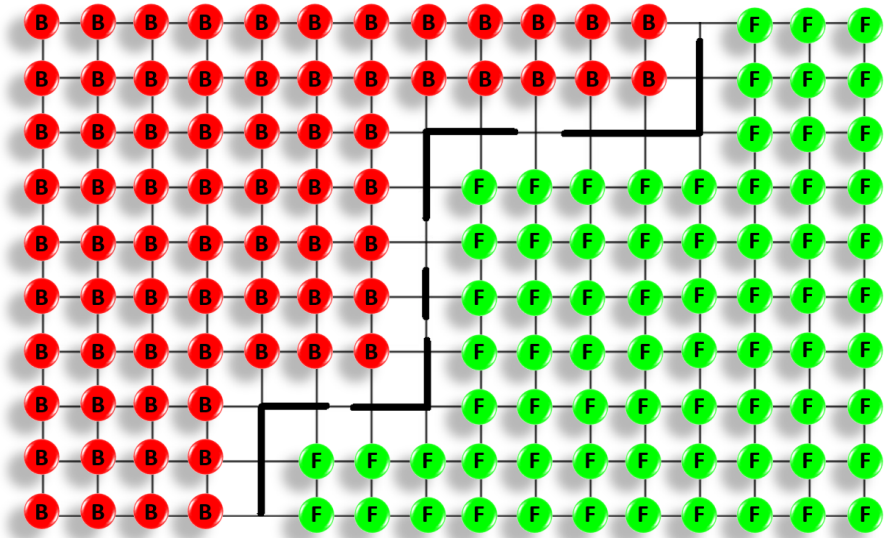



Figura 2.4: Efeito de propagação das *seeds* após a atribuição de rótulos de "*Foreground*" e "*Background*" ao pontos ainda não rotulados.

Embora a equação (2.6) possa ser minimizada a partir da localização de seu ponto de mínimo global, existem ainda outras formas de se obter o vetor solução $x$. Por exemplo, (Grady [33]) leva em consideração as probabilidades dos pixels atingirem um certo rótulo $F$ ou $B$ e, dessa forma, atribui rótulos aos pixels ainda não rotulados a partir dessas probabilidades:

$$D^{-1}Wx = x\,. \tag{2.7}$$

Os processos de solução do RW levam em consideração que as probabilidades se assemelham às segmentações produzidas por *Seeded Region Growing* (Adams e Bischof [1]), porém, existem algumas propriedades interessantes quanto à solução proposta por Grady. Por exemplo, apesar de a descrição fazer parecer que o problema seja estocástico, na verdade ele é determinístico e existe uma relação com algumas interpretações físicas: se definirmos um conjunto de rótulos como $1V$ e o outro como $0V$, a probabilidade de um certo pixel atingir um dos conjuntos rotulados pode ser calculada de forma semelhante a encontrar a tensão de uma rede de resistores pelas leis de Kirchhoff (Doyle e Snell [28]).

### 2.1.3 Shortest Path Forest

Este método emprega uma ideia muito difundida na análise de grafos, isto é, este explora o uso de caminhos mínimos em grafos a fim de cumprir com a segmentação da imagem.

O *Shortest Path Forest* (Bai e Sapiro [4]) é um método de segmentação interativo que interpreta a tarefa de particionamento como um problema de encontrar o menor caminho em um grafo. Imagine que o peso $w_{ij}$ relacionado à aresta $(i, j)$, que liga um par de pixels (representados pelos nós do grafo), seja menor quando características como intensidade, cor, gradiente, entre outras, são semelhantes, e de maneira oposta, $w_{ij}$ seja maior quando existe um grande contraste entre os pixels $i$ e $j$. Assim, o método calcula a soma mínima dos pesos entre um pixel não rotulado $i$ e os pixels rotulados pelas marcações inicias. Em seguida, é atribuído a $i$ o mesmo rótulo do pixel que está mais próximo (com relação à distância em termos dos pesos entre os dois).

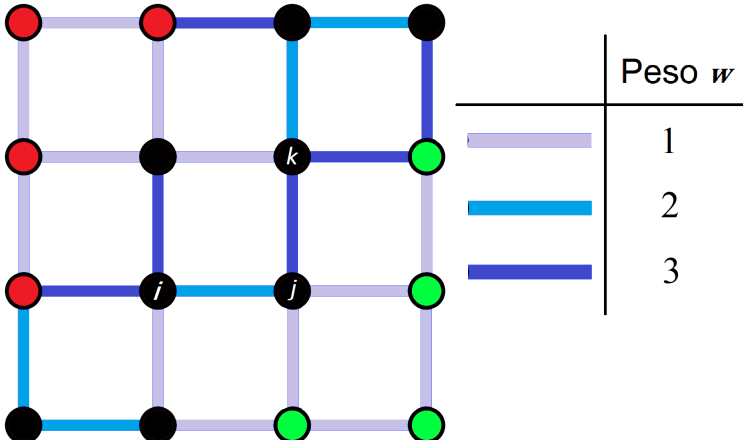



Figura 2.5: Grafo representando o método do menor caminho para a atribuição de rótulos para os nós $i, j$ e $k$. Sendo os nós indicados em preto aqueles que representam os pixels ainda não rotulados, os nós em verde os pixels com o rótulo $x_F$ de objeto, e finalmente, os nós em vermelho os pixels com rótulo $x_B$ de fundo da imagem.

No Grafo ilustrado na Figura 2.5, ao levar em consideração o peso $w$ de cada aresta, é possível prever que os nós $i, j$ e $k$ do grafo irão receber os seguintes rótulos:

$$\begin{cases} i & \longrightarrow x_F, \\ j & \longrightarrow x_F, \\ k & \longrightarrow x_B. \end{cases} \tag{2.8}$$

Note que, apesar de $i$ estar mais próximo de um pixel rotulado como $x_B$, a aresta entre eles tem maior peso, ou seja, o contraste entre eles é maior, e portanto o menor caminho o leva para um pixel com rótulo $x_F$, tornando-o parte do objeto após a segmentação.

No que concerne os funcionais de energia de abordagens do tipo *Shortest Paths*, segundo Couprie e colaboradores [21], na maioria dos trabalhos relacionados, ambas as constantes $p$ e $q$ divergem no funcional de referência (2.1), ou seja, $p \to \infty$, assim como $q \to \infty$.

Apesar de apresentar menor custo computacional, realizar a segmentação por um método de menor caminho torna a eficiência muito dependente do posicionamento das *seeds*. Eventualmente um pixel com características muito semelhantes com o objeto-alvo pode estar longe de uma *seed* rotulada como objeto e, contrariamente, estar próximo a uma com rótulo de fundo, sendo assim equivocadamente descartada da solução como objeto. Nguyen e colaboradores [51] apresentaram uma abordagem diferente a qual torna a escolha das sementes menos importante para o resultado final da segmentação, porém, tal mecanismo depende de uma pré-segmentação para funcionar de forma efetiva.

### 2.1.4 Watersheds

O termo *watershed*, em sua forma literal, nada mais é do que um termo hidrólogo, que se refere a bacias hidrográficas, ou melhor, as linhas que separam uma bacia da outra.

Bacias hidrográficas são regiões que são separadas por linhas chamadas de divisores de água, que constituem todas as fontes de uma bacia hidrográfica. Neste caso, água da chuva, de rios ou afluentes e até mesmo subterrânea, tendem a escoar naturalmente do

ponto mais alto desta região até o ponto mais baixo, onde acontece o acúmulo desta bacia. O que difere duas bacias vizinhas é para onde a água está escoando, ou seja, os limites da região em que toda a água escorre para o mesmo lugar. A Figura 2.6 ilustra a concepção empregada por modelos construídos nesse ramo de pesquisa.

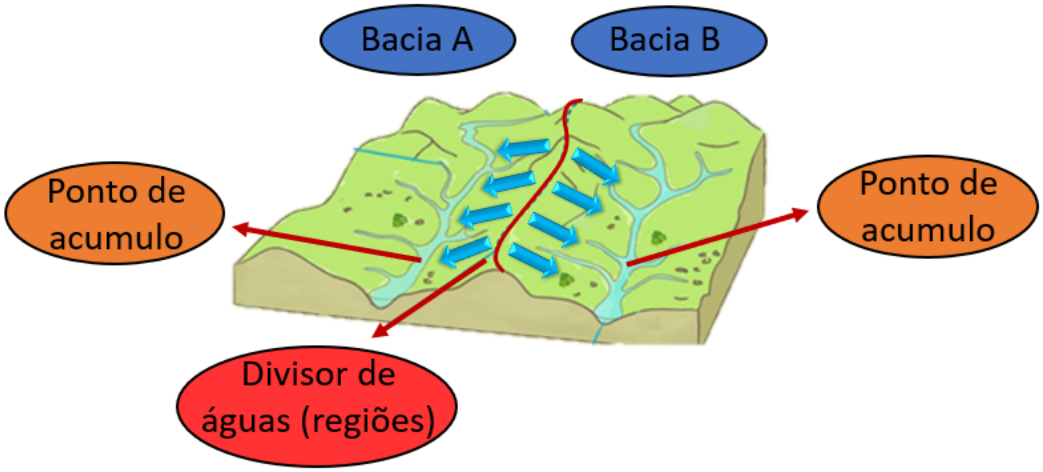



Figura 2.6: Ilustração de duas bacias vizinhas, seu divisor e seus pontos de acumulo (Adaptado de França [29]).

A literatura a respeito das técnicas que levam em consideração o tipo de abordagem *watersheds* é bastante abrangente, sendo possível explorar desde problemas que envolvem análise topológica (Bertrand [10]) até o nosso foco, a segmentação de imagens, como, por exemplo, (Monteiro e Campilho [49], Bai e Urtasun [5]).

A técnica de segmentação de imagens chamada de *Watersheds* pode ser interpretada como se a imagem fosse uma grande área com diversas bacias, em que hipoteticamente uma inundação de diversas fontes é "criada" e espalhada pela imagem (Figura 2.7). Desta forma, o que se procura é basicamente as linhas que dividirão estas bacias. Conforme a água vai subindo, águas de diferentes regiões ficarão próximas de se encontrarem e, neste momento, é criada uma linha a qual dividirá as regiões. Finalmente, o divisor de águas (Figura 2.8), que no ponto de vista da segmentação, será considerada como sendo uma fronteira da imagem.

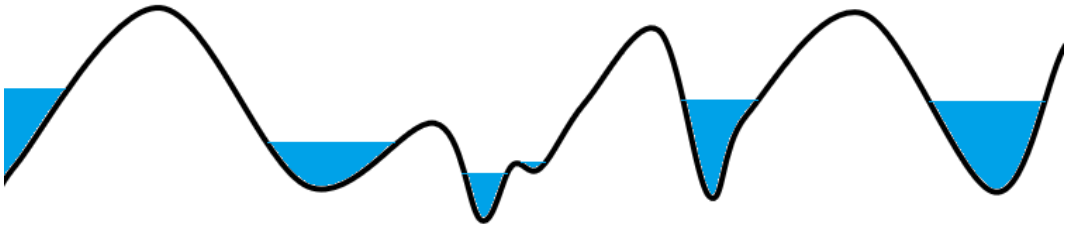


Figura 2.7: Início do alagamento provocado em uma região.

O final deste processo gera a divisão da imagem em diferentes agrupamentos (*clusters*), de pixels que são considerados próximos ou semelhantes, criando assim a segmentação desejada da imagem.

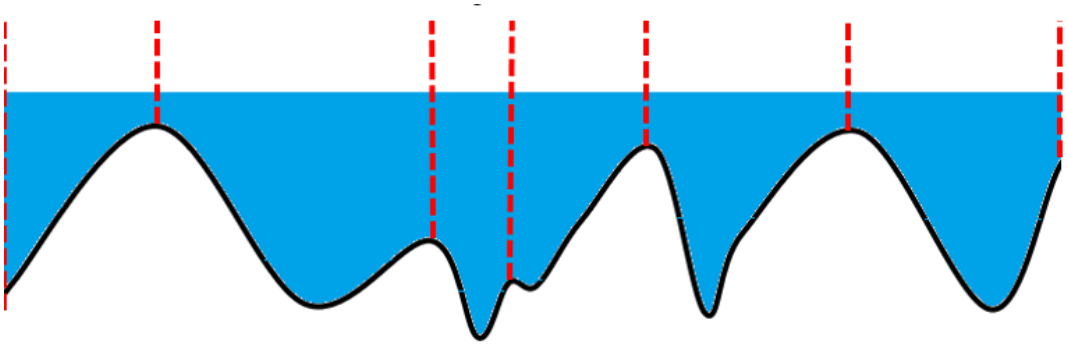


Figura 2.8: Divisão das bacias hidrográficas (regiões da imagem) após a água se espalhar por completo.

O processo descrito acima foi inicialmente proposto na área de segmentação de imagens por Cousty e colaboradores [22], e o cálculo dos divisores de águas, ou ainda, dos limites das regiões da imagem, pode ser realizado, por exemplo, com a ajuda de um dentre dois algoritmos clássicos: o Algoritmo de Kruskal [40] ou o Algoritmo de Prim [53].

Ainda utilizando este tipo de abordagem, Couprie e colaboradores [21] desenvolveram uma nova versão do método tradicional apresentado inicialmente por Cousty e colaboradores [22]. Este recebeu o nome de *Power Watersheds*, cujo segmentador correspondente pode assumir duas formas com respeito ao funcional da equação (2.1): uma quando $q = 1$, e outra quando $q = 2$, porém, em ambos os casos, o valor de $p$ diverge. Segundo os autores, o *Power Watersheds* se mostrou tão rápido quanto a proposta original, no entanto, este apresentou maior robustez quando comparado a todos os métodos dessa categoria que foram testados na época em que sua proposta foi publicada.

## 2.2 Segmentação via Coordenadas de Laplace

Casaca e colaboradores [20, 17] desenvolveram uma técnica de segmentação denominada *Coordenadas de Laplace* (LC - *Laplacian Coordinates*), que foi recentemente aperfeiçoada em [19], e que opera através do fornecimento de *seeds*, isto é, através de marcações (*brushes*) fornecidas pelo usuário a fim de segmentar a imagem (vide Figura 2.9 para uma ilustração). Esse método procura minimizar a distância de cada $x_i$, que é o elemento do mapa de saliência ($x_i$) associado a cada nó/pixel $i$, com relação aos seus vizinhos mais próximos no grafo-base construído a partir da imagem.

Tal formulação resultou no algoritmo de Segmentação de Imagens Coordenadas de Laplace, que procura promover os seguintes passos durante a tarefa de segmentação:

- Definir as *seeds* na imagem a partir dos *brushes* operados pelo usuário;
- Construir um grafo de afinidade;
- Estruturar e solucionar o funcional de energia das Coordenadas de Laplace;
- Segmentar a imagem (isto é, caracterizar os pixels entre "objeto" e "fundo").

O primeiro passo acima constitui na definição das *seeds*, sendo este processo uma tarefa a ser realizada pelo usuário: as (*seeds*) são fornecidas de forma interativa e, então, gravadas pelo programa de modo a indicar quais são as regiões que contém o "objeto", cujo rótulo associado é $F$, e o "fundo" da imagem, cujo rótulo é $B$.

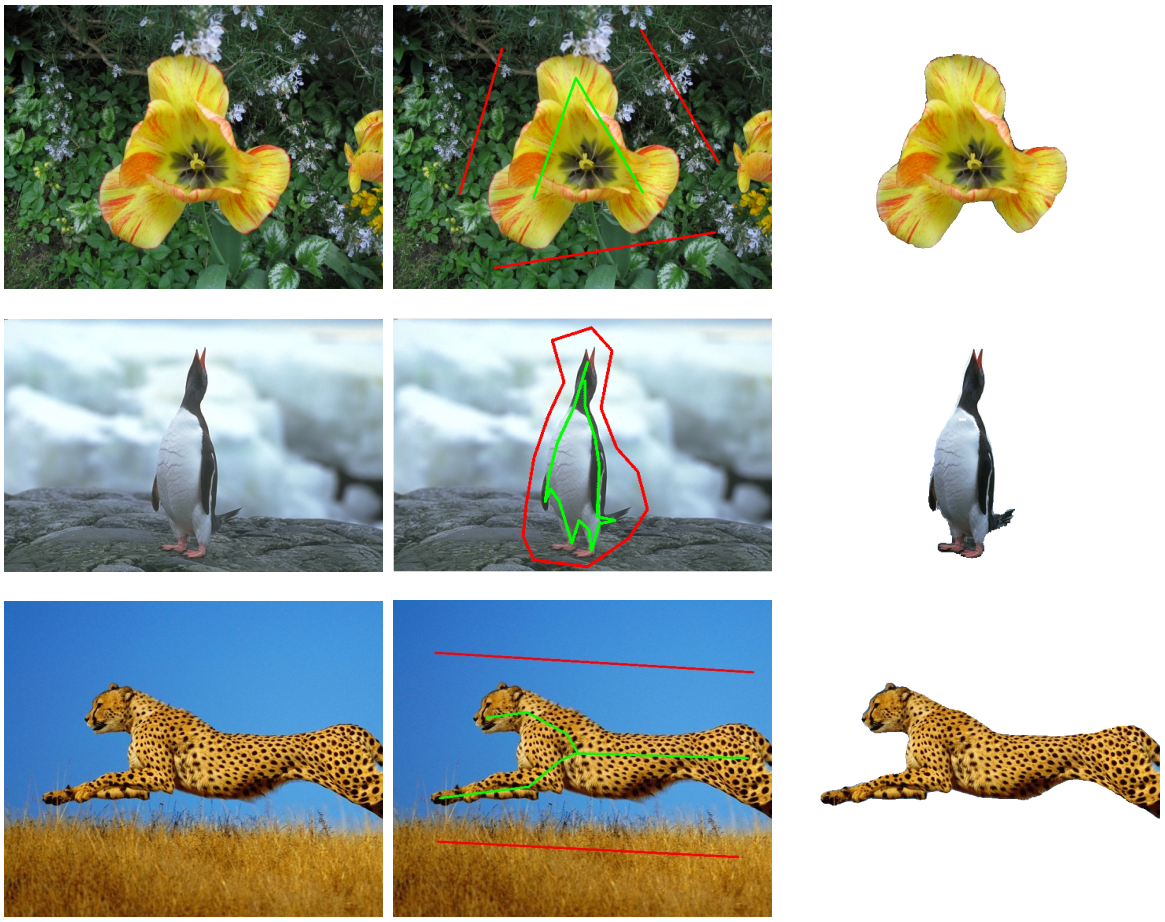


Figura 2.9: Exemplo de segmentação a partir da técnica Coordenadas de Laplace. Da esquerda para direita, tem-se: a imagem original, imagem com as *seeds* que representam o objeto (em verde), e o fundo (em vermelho), e o resultado da segmentação.

### 2.2.1 O Funcional de Energia

Concluído o primeiro passo de alocação das *seeds*, tem-se como próxima etapa a construção do grafo de afinidade a partir das características encontradas na imagem $I$. Assim, seja $I$ a imagem de entrada a ser processada. No caso da imagem ser colorida, podemos denotar o vetor RGB de um pixel $i$ como $I_i = (R_i, G_i, B_i)$, onde cada uma das componentes desse vetor representam as componentes de luminosidade vermelha, verde e azul de um pixel $P_i \in I$ (o modelo RGB é descrito de forma mais detalhada na Seção 3.2.1). Já para uma imagem em tons de cinza, basta tomar o valor $I_i$ que representa a intensidade original do pixel na referida escala.

A partir da imagem $I$ e seus componentes de cor, a construção do grafo de afinidade é então dada da seguinte maneira:

$$G = (V, E, W_E)\,, \tag{2.9}$$

em que $V$ representa o conjunto de nós $i \in V$ do grafo, ou seja, os pixels $P_i$ da imagem $I$, $E$ representa o conjunto de arestas do grafo, e neste caso, considera-se uma vizinhança de 8 pixels, e por fim, $W_E$ representa o conjunto dos pesos $w_{ij}$ das arestas $(i, j) \in E$. Além disso, define-se como $N(i)$ o conjunto $\{j : (i, j) \in E\}$, que representa todos os índices dos pixels $P_j$ que estão na vizinhança de $P_i$, tal como ilustrado na Figura 2.10. Desta forma, a valência associada ao pixel $P_i$ é dada por:

$$d_i = \sum_{j \in N(i)} w_{ij} \tag{2.10}$$

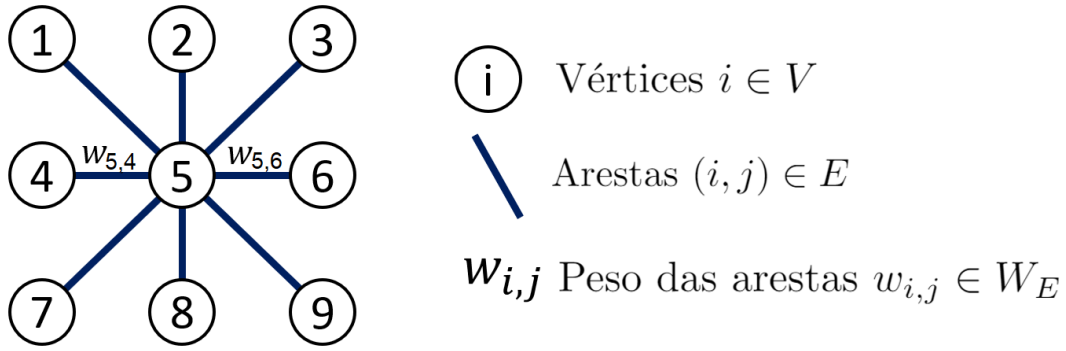



Figura 2.10: Representação da vizinhança de um pixel $P_5$.

**Obtenção dos Pesos do Grafo e da Segmentação da Imagem**

O cálculo dos valores dos pesos $w_{ij} \in W_E$ de cada aresta $(i,j) \in W$ pode ser realizado a partir de diferentes características dos pixels da imagem, tais como intensidade, gradiente, e contorno (Cai e colaboradores [15], Casaca e colaboradores [18]). Tendo como um dos objetivos obter uma implementação mais simples do segmentador, o método descrito por Casaca e colaboradores [18] leva em consideração apenas a intensidade do pixel no momento do cálculo do peso $w_{ij} = w(P_i, P_j)$ relacionado à aresta que liga o par de pixels $(P_i, P_j)$. Assim, tal valor é obtido da seguinte maneira:

$$w_{ij} = \exp\left(-\frac{\beta ||I_i - I_j||^2_\infty}{\sigma}\right), \quad \sigma = \max_{(i,j)\in E} ||I_i - I_j||_\infty , \tag{2.11}$$

em que $\beta$ é um escalar que representa um parâmetro de ajuste. Note que, neste caso, os pesos são todos positivos e simétricos, isto é, $w_{ij} = w_{ji}$. Além disso, visando evitar valores $w_{ij}$ nulos em função do truncamento nos cálculos numéricos, Grady [34] sugere adotar uma constante $\varepsilon = 10^{-6}$, onde $\varepsilon$ é o valor mínimo que deve ser somado à $w_{ij}$.

Se considerarmos:

$$\alpha = \alpha(i,j) = \left(\frac{||I_i - I_j||^2_\infty}{\sigma}\right), \tag{2.12}$$

então, na equação (2.11), o valor de $w_{ij}$ pode ser representado por $e^{-\beta\alpha}$. Neste caso, note que o peso atribuído à aresta $(i,j)$ decairá de forma mais rápida conforme o valor de $\beta$ aumenta, ou seja, $\beta$ tem influência direta no valor do peso das arestas. Assim, um estudo foi dirigido sobre o referido parâmetro de ajuste $\beta$ com a finalidade de buscar melhorias na computação dos pesos $w_{ij}$ e, por conseguinte, no resultado final da segmentação.

Considere as informações fornecidas pelo usuário (por meio da inserção das *seeds*), isto é, o conjunto de índices $F = \{i : P_i \in I$ rotulado como objeto$\}$ e o conjunto homólogo $B = \{i : P_i \in I$ rotulado como fundo$\}$. Desta forma, a partir dos valores $w_{ij}$ da matriz de pesos $W$, podemos definir o funcional de energia das Coordenadas de Laplace $E_{LC}$ tal

como proposto por Casaca e colaboradores [20]:

$$E_{LC}(x) = \sum_{i \in F} ||x_i - x_F||_2^2 + \sum_{i \in B} ||x_i - x_B||_2^2 + \sum_{i \in V} \left|\left| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right|\right|_2^2 . \tag{2.13}$$

A solução de $E_{LC}(x)$ retorna o vetor real $x = (x_i)$, cujos elementos estão relacionados com os pixels $P_i$ da imagem, gerando assim uma mapa escalar para a mesma, a vetorização da matriz que representará a solução do problema de segmentação. Finalmente, $x_F$ e $x_B$ são valores reais pré-definidos no funcional de energia, onde deve-se ter $x_F \neq x_B$. Neste caso, note que, esses valores correspondem aos pixels dos conjuntos $F$ e $B$ respectivamente. Sendo assim, se assumirmos $x_F \geq x_B$, podemos então atribuir os rótulos $y_i \in \{x_F, x_B\}, i \in V$, que definem se um pixel faz parte do objeto de interesse ou do fundo da imagem tal como segue:

$$y_i = \begin{cases} x_F, & \text{se} \quad x_i \geq \frac{x_F + x_B}{2} \\ x_B, & \text{caso contrário} \end{cases} , \tag{2.14}$$

obtendo assim o resultado definitivo da segmentação.

### Minimizando a Energia das Coordenadas de Laplace

O funcional de energia $E_{LC}$, descrito na equação (2.13), pode ser divido em dois termos-chaves:

$$E_{LC}(x) = T_{(F,B)}(x) + LC(x) \tag{2.15}$$

com,

$$T_{(F,B)}(x) = \sum_{i \in F} ||x_i - x_F||_2^2 + \sum_{i \in B} ||x_i - x_B||_2^2 \tag{2.16}$$

$$LC(x) = \sum_{i \in V} \left|\left| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right|\right|_2^2 , \tag{2.17}$$

com $T_{(F,B)}(x)$ representando o termo de fidelidade relacionado às *seeds* (conjuntos $F$ e $B$), e $LC(x)$ representando o termo central das Coordenadas de Laplace, isto é, a componente que controla o ajuste nas vizinhanças de cada pixel da imagem de forma que cada valor $x_i$ associado ao pixel $P_i$ deverá se aproximar da média ponderada de sua vizinhança $N(i)$.

Note que o termo $LC(x)$ pode ser reescrito na forma matricial como segue:

$$\sum_{i \in V} \left|\left| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right|\right|_2^2 = \sum_{i \in V} \left( \sum_{j \in N(i)} w_{ij} x_i - \sum_{j \in N(i)} w_{ij} x_j \right)^2 = \sum_{i \in V} \left( \sum_{j \in N(i)} w_{ij} (x_i - x_j) \right)^2$$

$$= \left( \sum_{j \in N(1)} w_{1j} (x_1 - x_j) \right)^2 + \left( \sum_{j \in N(2)} w_{2j} (x_2 - x_j) \right)^2 + ... + \left( \sum_{j \in N(n)} w_{nj} (x_n - x_j) \right)^2 =$$

$$= \underbrace{\left( \sum_{j \in N(1)} w_{1j}(x_1 - x_j), \sum_{j \in N(2)} w_{2j}(x_2 - x_j), ..., \sum_{j \in N(n)} w_{nj}(x_n - x_j) \right)}_{\mathbf{v}^t} .\mathbf{v} = (\mathbf{Lx})^t(\mathbf{Lx}),$$

Logo,

$$\sum_{i \in V} \left|\left| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right|\right|_2^2 = (\mathbf{Lx})^t(\mathbf{Lx}) = ||\mathbf{Lx}||_2^2 \,, \tag{2.18}$$

em que $\mathbf{L}$ é a matriz Laplaciana do grafo base (já comentada brevemente em 2.1.2), a qual é determinada por $\mathbf{L} = \mathbf{D} - \mathbf{W}$, sendo $\mathbf{D}$ uma matriz diagonal contendo os pesos da valência em seus elementos da diagonal, $D_{ii} = d_i$ (a equação (2.10) descreve os valores $d_i$), e $\mathbf{W}$ a matriz dos pesos $w_{i,j}$, que corresponde à matriz (ponderada) de adjacência do grafo base (a equação (2.11) descreve os valores $w_{ij}$). Neste caso, note que $\mathbf{W}$ é esparsa por construção, uma vez cada nó do grafo se conecta com no máximo 8 vizinhos. Além disso, se a imagem de entrada tem dimensão $b \times h$, com $n = (b \cdot h)$, então $\mathbf{W}$ tem dimensão $n \times n$. Por exemplo, para uma imagem de resolução $500 \times 500$ a matriz de entrada teria tal dimensão, e a matriz $\mathbf{W}$ seria de dimensão $250.000 \times 250.000$ (esparsa), cujos elementos são definidos da seguinte forma:

$$\mathbf{W}_{ij} = \begin{cases} w_{ij}, & \text{se} \quad (i,j) \in E \\ 0, & \text{caso contrário} \end{cases} . \tag{2.19}$$

Sendo assim, os elementos de $\mathbf{L}$ (de tamanho $n \times n$) são definidos da seguinte forma:

$$\mathbf{L}_{ij} = \begin{cases} d_i, & \text{se} \quad i = j, \\ -w_{ij}, & \text{se} \quad (i,j) \in E \\ 0, & \text{caso contrário} \end{cases} \tag{2.20}$$

Por fim, para minimizar o funcional $E_{LC}$ em (2.13), podemos reescrever a equação (2.13) em sua forma matricial tal como segue:

$$\begin{aligned} E_{LC}(x) &= \sum_{i \in F} ||x_i - x_F||_2^2 + \sum_{i \in B} ||x_i - x_B||_2^2 + \sum_{i \in V} \left|\left| d_i x_i - \sum_{j \in N(i)} w_{ij} x_j \right|\right|_2^2 \\ &= \sum_{i \in F} (x_i - x_F)^2 + \sum_{i \in B} (x_i - x_B)^2 + ||\mathbf{Lx}||_2^2 \\ &= \sum_{i \in F} (x_i^2 - 2x_i x_F + x_F^2) + \sum_{i \in B} (x_i^2 - 2x_i x_B + x_B^2) + (\mathbf{Lx})^t(\mathbf{Lx}) \\ &= \sum_{i \in S} x_i^2 - 2\left( \sum_{i \in F} x_i x_F + \sum_{i \in B} x_i x_B \right) + \#F x_F^2 + \#B x_B^2 + \mathbf{x}^t(\mathbf{L}^t\mathbf{L})\mathbf{x} \\ &= \mathbf{x}^t\mathbf{L}^2\mathbf{x} + \mathbf{x}^t\mathbf{I}_S\mathbf{x} - 2\mathbf{x}^t b + (\#F x_F^2 + \#B x_B^2) \end{aligned} \tag{2.21}$$

ou simplesmente,

$$E_{LC}(x) = \mathbf{x}^t(\mathbf{I}_S + \mathbf{L}^2)\mathbf{x} - 2\mathbf{x}^t b + c \,, \tag{2.22}$$

sendo $S = F \cup B$ o conjunto de todas as *seeds* independente do rótulo, $c$ uma constante real, e $\mathbf{I}_S$ uma matriz diagonal, construída da seguinte forma:

$$\mathbf{I}_S = \begin{cases} 1, & \text{se} \quad i \in S \\ 0, & \text{caso contrário} \end{cases} . \tag{2.23}$$

Já no caso do vetor $b$, este possui os valores correspondentes às *seeds*:

$$b(i) = \begin{cases} x_B, & \text{se} \quad i \in B, \\ x_F, & \text{se} \quad i \in F \\ 0, & \text{caso contrário} \end{cases} . \tag{2.24}$$

Como a expressão obtida na equação (2.22) é quadrática, minimizar tal funcional é, na realidade, equivalente a encontrar o vetor solução $\mathbf{x}$ que resolve o seguinte sistema linear:

$$(\mathbf{I}_S + \mathbf{L}^2)\mathbf{x} = b\,. \tag{2.25}$$

Tal como foi definido pela equação (2.11), os elementos $w_{ij} \in W$ são positivos e simétricos, isto é, $w_{ij} = w_{ji}$, de modo que a matriz $\mathbf{L}$ é simétrica e semi-definida positiva. Além disso, foi demonstrado em [19] que, se o grafo $G$ é conectado, então a matriz real $(\mathbf{I_S} + \mathbf{L^2})$ é definida positiva. Além disso, ela é também é esparsa, garantindo ao sistema (2.25) ótimas propriedades matemáticas, tornando possível, por exemplo, a resolução via fatoração de *Cholesky* [24].

Capítulo

3

# Análise de Features na Tarefa de Segmentação Interativa

Como a intenção desta pesquisa é aperfeiçoar o cálculo dos valores $w_{ij}$ de modo a melhorar o resultado final da segmentação, serão fornecidos como valores de entrada para o cálculo de $w_{ij}$ configurações distintas de bandas (*features*) resultantes do processamento da imagem de entrada através dos métodos descritos neste capítulo. Sendo assim, podemos então reescrever a equação (2.11) de forma a acomodar todas as bandas testadas, isto é, através da adição de uma variável de profundidade $k$:

$$w_{ij}^k = \exp\left(-\frac{\beta||I_i^k - I_j^k||_\infty^2}{\sigma_k}\right), \quad \sigma_k = \max_{(i,j)\in E} ||I_i^k - I_j^k||_\infty \tag{3.1}$$

onde $k$ representa a $k$-ésima *feature* computada da imagem.

Neste capítulo (e no subsequente), serão descritas em detalhes as bandas (*features*) que serviram de parâmetro para o cálculo da matriz de pesos $W$ construída tal como proposta acima.

Cada *feature* $k$ é produzida a partir do processamento da imagem original de entrada, isto é, a qual se deseja segmentar. Assim, a imagem passa por um determinado número de métodos de extração de bordas ou destacamento de fronteiras, além de filtros de pré-segmentação de modo a realçar os limites dos objetos presentes na imagem (vide exemplo apresentado na Figura 3.1).

Figura 3.1: Imagem original acompanhada de algumas *features* extraídas da imagem, que representam diferentes leituras por métodos de PDI aplicados a fim de destacar os limites do objeto de interesse na imagem.

Os métodos de PDI aplicados tem como objetivo acentuar as bordas da imagem por diferentes pontos de vista, fazendo com que os pixels considerados como bordas em mais de uma banda tenham então uma maior importância no peso final $w_{ij}$ das arestas $(i, j) \in E$. Além disso, possíveis bordas que poderiam passar desapercebidas por determinados métodos extratores de *features* são eventualmente detectadas nesse processo, dando, mesmo que de forma sutil, certa importância para determinada informação presente na imagem.

No caso da metodologia adotada nesta pesquisa, os métodos empregados não são apenas compostos por detectores de bordas. Em algumas das análises conduzidas, foram realizadas combinações de métodos de pré-segmentação como o *Gaussian Mixture Model*, o qual emprega o algoritmo EM (*Expectation Maximization Algorithm*) a fim de destacar bordas utilizando a variação da distância local em uma imagem, produzindo um conjunto de *features* bem relevantes para a segmentação final. Além disso, métodos de conversão de modelos de cores também foram analisados e adotados, pois estes ajudam a destacar certas características que o sistema clássico RGB não deixava tão evidente, facilitando assim a utilização das ferramentas de detecção usadas neste trabalho. A seguir é detalhado cada uma das bandas investigadas nesta pesquisa.

## 3.1 Modelo de Mistura Gaussiana

O *Gaussian Mixture Model* (GMM) é um modelo que se baseia na combinação linear de filtros gaussianos. Modelos de misturas são formatos pelas componentes da combinação desses filtros e seus respectivos pesos, sendo essas componentes definidas em termos de distribuições de probabilidade, aqui denotadas por $D_i$, com $i = 1, 2, ...$, e os pesos dados pelas probabilidades $\pi_i$ ligadas a estas distribuições. A densidade resultante deste tipo de modelo é dada pela combinação ponderada das densidades de cada uma das componentes, tendo como fator de ponderação da combinação a probabilidade $\pi_i$ relacionada a $D_i$, de forma que:

$$\sum_{i=1}^{K} \pi_i = 1, \;\; 0 \leq \pi_i \leq 1, \tag{3.2}$$

onde $K$ é o número de componentes do modelo, e o conjunto dos $\pi_i$, com $i = 1, 2, ..., K$, pode ser também denotado como uma probabilidade *a priori*. O agrupamento recorrente de modelos de misturas pode ser categorizado como um algoritmo de otimização iterativo no qual o objetivo principal é a maximização do critério de agrupamento a partir de uma função de verossimilhança (Bishop [11]). Para o caso de modelos de misturas gaussianas, podemos descrever os grupos pela média $\mu_i$ e a matriz de covariância $C_i$, e então escrever a função de densidade de probabilidade da seguinte forma:

$$p(x) = \sum_{i=1}^{K} \pi_i N_i(x|\mu_i, C_i), \tag{3.3}$$

onde $N_i(x|\mu_i, C_i)$ denota a distribuição normal relacionada ao vetor $\mu_i$ e a matriz de covariância $C_i$ como visto no trabalho de Fu e Wang [30]. Embora o modelo esteja bem definido do ponto de vista matemático, na prática, os valores dos parâmetros $\pi_i$, $\mu_i$ e $C_i$ não são conhecidos, e para problemas de agrupamento (*clustering*) estes valores precisam ser estimados de alguma maneira. Como o objetivo é maximizar a função de verossimilhança, neste trabalho, usaremos o algoritmo EM para determinar os valores desconhecidos.

### 3.1.1 Algoritmo EM

Quando nos deparamos com uma situação em que precisamos estimar parâmetros desconhecidos (o que é comum em diversos tipos de problemas de agrupamento), precisamos usar algum método para estimar tais valores. Assim, uma abordagem eficiente na realização dessa tarefa é o Algoritmo *Expectation Maximization* (EM). Tal algoritmo foi introduzido por Dempster e colaboradores [25] e vem sendo estudado e aprimorado desde então, tendo como principal papel o cálculo iterativo de estimativas de máxima verossimilhança, a *Maximum Likelihood Estimate* (MLE) (McLachlan e Krishnan [46]).

O algoritmo EM funciona em dois passos (McLachlan e Peel [45]). O primeiro passo é denotado de etapa E, *Expectation*, em que para esta etapa é definido um conjunto de dados completo que baseia-se nos dados observados para calcular o valor esperado da função de máxima verossimilhança. Em seguida, tem-se a etapa M, *Maximization*, que usa os dados obtidos da função de verossimilhança para definir os novos valores para os parâmetros desconhecidos anteriormente. As etapas E e M se repetem até a convergência do método iterativo.

Note que, se conhecêssemos todos os dados $x_k$ de um Modelo de Mistura Gaussiana, de $M$ componentes, então seria mais simples de se calcular os parâmetros desse modelo, representados por $(\pi_1, \pi_2, ..., \pi_M, \theta_1, \theta_2, ..., \theta_M)$, onde $\theta_i = (\mu_i, C_i)$ são as médias e as matrizes de covariância associadas. Ou ainda, se os parâmetros $(\pi_1, \pi_2, ..., \pi_M, \theta_1, \theta_2, ..., \theta_M)$ fossem conhecidos, então também seria possível gerar os valores dos dados $x_k$. Entretanto, na prática, nenhum destes valores são conhecidos. Portanto, uma saída é estimar tais parâmetros utilizando o Algoritmo EM que, como mencionado anteriormente, estima a máxima verossimilhança mesmo a partir de dados incompletos.

Sendo assim, a etapa E de um modelo de Mistura Gaussiana com $M$ componentes Gaussianas e com $N$ dados $x_k$ pode ser executada da seguinte forma:

$$Q(\theta|\theta^{(t)}) = \sum_{l=1}^{M}\sum_{i=1}^{N}\gamma(Z_{il})log\pi_l + \sum_{l=1}^{M}\sum_{i=1}^{N}\gamma(Z_{il})logN(x_i|\theta_l), \tag{3.4}$$

onde $N(x_i|\theta_l)$ representa a probabilidade da distribuição para a l-ésima componente Gaussiana de $x_i$ e $\gamma(Z_{il})$ é a probabilidade a posteriori da l-ésima componente Gaussiana, que segundo Bishop [11], é representada por:

$$\gamma(Z_{il}) = \frac{\pi_l N(x_i|\theta_l)}{\sum_{j=1}^{M}\pi_j N(x_i|\theta_j)}\,. \tag{3.5}$$

Após a realização dos passos da etapa E, os dados obtidos são então usados na reestimativa dos parâmetros da próxima iteração pela etapa M, que pode ser resumidamente descrita pelos seguintes passos:

$$\pi_l^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N}\gamma(Z_{il}) \tag{3.6}$$

$$\mu_l^{(t+1)} = \frac{1}{N\pi_l^{(t+1)}}\sum_{i=1}^{N}x_i\gamma(Z_{il}) \tag{3.7}$$

$$C_l^{(t+1)} = \frac{1}{N\pi_l^{(t+1)}}\sum_{i=1}^{N}(x_i - \mu_l^{(t+1)})^2\gamma(Z_{il})\,. \tag{3.8}$$

Quando a etapa M é concluída, os parâmetros re-estimados são usados na próxima iteração da etapa E, assim as etapas *Expectation* e *Maximization* se alternam até a convergência. Exemplos visuais de segmentação para diferentes parâmetros $k$, de números de agrupamentos desejados, podem ser vistos na Figura 3.2.

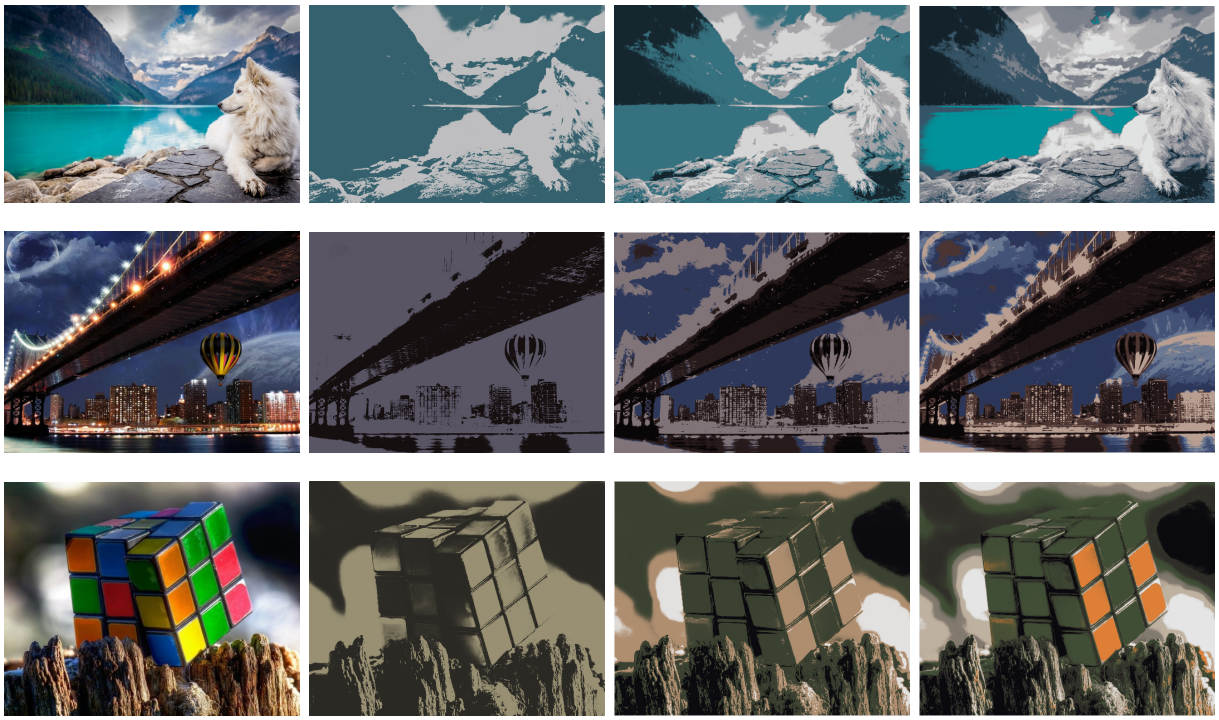


Figura 3.2: Resultados do processamento de imagens via GMM através do Algoritmo EM para diferentes parâmetros $k$. Da esquerda para a direita: imagens de entrada, e os resultados obtidos tomando-se $k = 2$, $k = 4$ e $k = 7$.

## 3.2 Modelos de Cores e Suas Conversões

Quando se trata da representação de uma imagem digital colorida, existem diversos sistemas de definição de cores, em que cada modelo (também conhecidos como espaço de cores) define as cores utilizando componentes de cores específicos. Enquanto o olho humano está limitado a interpretar dezenas de tonalidades de cinza, os sistemas de cores são capazes de diferenciar um grande número de tonalidades e intensidades de cores. Do ponto de vista matemático, para que a representação vetorial de uma cor seja factível, é necessário um espaço de cores capaz de representar o maior número possível de tonalidades diferentes, e assim possuir uma base capaz de gerar todo o espaço de combinações.

O objetivo de um dado modelo de cores é facilitar a especificação de cores em algum padrão. Essa especificação é feita em um sistema de coordenadas, geralmente tridimensional, no qual cada cor dentro deste sistema é representada por um único ponto, sendo estes modelos projetados para *hardwares* específicos ou, ainda, para serem passíveis de percepção humana.

Dentre os diversos modelos de cores orientados para *hardwares*, temos aqueles que são usados com maior frequência, como por exemplo, o RGB (*Red, Green, Blue*), muito comum em monitores e câmeras de vídeo, o CMYK (*Cyan, Magenta, Yellow, Black*) para impressões coloridas, e ainda outros sistemas que englobam também modelos orientados

para percepção humana como o HSI, HSV, YIQ (Gonzalez e Woods [32]). Neste trabalho, foram utilizadas *features* baseadas nos seguintes modelos de cores: o sistema (RGB), o modelo usado para conversão (XYZ), e o modelo resultante do (L*a*b*).

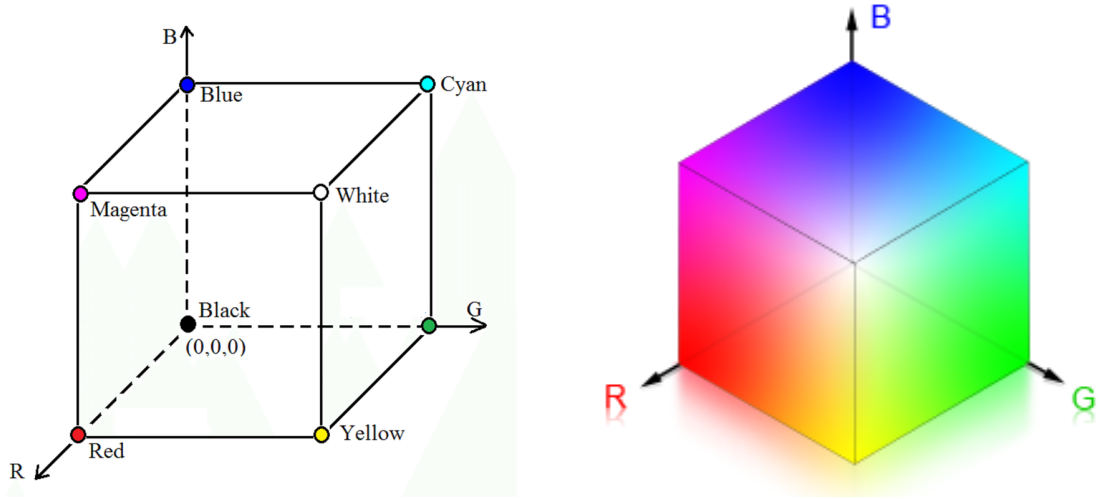
### 3.2.1 O Modelo RGB

O RGB se categoriza como um modelo aditivo de cores, onde as cores nos seus componentes primários, vermelho, verde e azul se unem para formar todas as outras cores. Em geral, este se define a partir de três componentes de cores primárias, por ser o número de diferentes tipos de fotorreceptores que o olho humano possui. Note que é possível um modelo ter mais do que três cores primárias, no entanto, isso não teria uma representação bem definida no sentido da teoria matemática empregada de espaço vetorial. A representação de uma cor $C$ no modelo RGB para cada pixel da imagem pode ser obtida por:

$$C = rR + gG + bB\,, \tag{3.9}$$

na qual $R$ representa a luz vermelha, com comprimento de onda de $700nm$, $G$ a luz verde, com comprimento de onda de $546nm$, e $B$ a luz azul, com comprimento de onda de $435.8nm$. Os valores $r$, $g$, $b$ são os coeficientes da combinação, definindo assim a base de RGB onde uma dada cor é definida pela terna $(r, g, b)$ como um ponto deste espaço.

Se pensarmos no modelo em um sistema de coordenadas cartesianas tal como na Figura 3.3, considerando um cubo unitário onde as intensidades das três cores primárias $r$, $g$, $b$ representam os eixos $x$, $y$, $z$, respectivamente, com $r, g$ e $b$ pertencentes ao intervalo fechado $[0, 1]$, temos então que a cor vermelha está na coordenada $(1, 0, 0)$, a cor verde em $(0, 1, 0)$, e a cor azul em $(0, 0, 1)$.




(a) Cores nos vertices (b) Representação completa

Figura 3.3: Representações do modelo RGB através do sistema de coordenadas cartesianas com as cores *red, green* e *blue* associadas aos eixos $x$, $y$ e $z$ respectivamente.

Como a cor preta representa a ausência de cores, ela está na origem. De forma contrária, a cor branca representa a união de todas as cores e se localiza em $(1, 1, 1)$, e finalmente, as tonalidades de cinza são representas no sistema RGB nos pontos $(r, g, b)$ tais que $r = g = b$, ou seja, as tonalidades da cor cinza estão sobre a diagonal que liga a cor preta à cor branca.

E possível definir também o amarelo, que é resultante da combinação de verde com o vermelho, o magenta, pela combinação de vermelho com o azul, e o ciano, a partir da combinação de verde com o azul. As cores amarelo, magenta e ciano estão localizadas respectivamente nos vértices $(1, 1, 0)$, $(1, 0, 1)$ e $(0, 1, 1)$. Assim, o subespaço de todas as cores do RGB são os pontos sobre ou dentro deste cubo.

Os exemplos mais comuns do uso desse modelo são *hardwares* como televisores, monitores, maquinas fotográficas digitais, *datashows*, entre outros, e as imagens digitais no modelo RGB são constituídas de três planos independentes, que representam cada um deles uma das três cores primárias do modelo que se combinam sobre a tela do monitor colorido para formar uma imagem de cores compostas como na Figura 3.4.

Figura 3.4: Composição das cores $r$, $g$ e $b$ em uma imagem colorida.

### 3.2.2 O Modelo CIEXYZ

Assim como o RGB, o modelo de cores XYZ também é um modelo aditivo de cores primárias da CIE (Comissão Internacional de Iluminação), do francês *Commission internationale de l'éclairage*. Este modelo descreve uma cor através de três cores primárias virtuais, $X$, $Y$ e $Z$, que são definidas matematicamente e servem como uma base para a maioria dos modelos de cores, sendo comum a utilização do XYZ em conversões entre sistemas de cores.

O modelo XYZ foi criado no ano de 1931 pela CIE devido à inexistência de um conjunto finito de cores primárias que permitisse representar todas as cores visíveis em um gráfico bidimensional através de coordenadas de cromaticidade $x$ e $y$. Nesse modelo, a intensidade da cor define a sua tonalidade: por exemplo, a cor branca e um tom de cinza, a princípio, tem a mesma combinação de cores primárias, no entanto, o branco é muito mais intenso que o cinza.

As cores primárias $X$, $Y$ e $Z$ desse modelo podem ser expressas matematicamente da seguinte maneira:

$$x = \frac{X}{X+Y+Z} \tag{3.10}$$

$$y = \frac{Y}{X+Y+Z} \tag{3.11}$$

$$z = \frac{Z}{X+Y+Z} \tag{3.12}$$

onde $x$ e $y$ representam a matiz e a saturação que, quando combinados, formam a cromaticidade, e $z$ representa a luminância ou intensidade luminosa. Como pode ser observado nas equações acima, temos que $x + y + z = 1$.

Conforme salientado anteriormente, um dos grandes pontos positivos desse modelo de cores, inclusive uma das motivações de sua criação, é que pode-se separar a variável de intensidade luminosa $z$ de forma a representar o espaço de cores em um gráfico com as dimensões de cromaticidade $x$ e $y$, como na Figura 3.5.

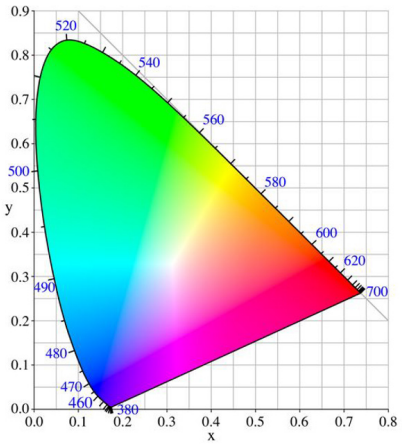
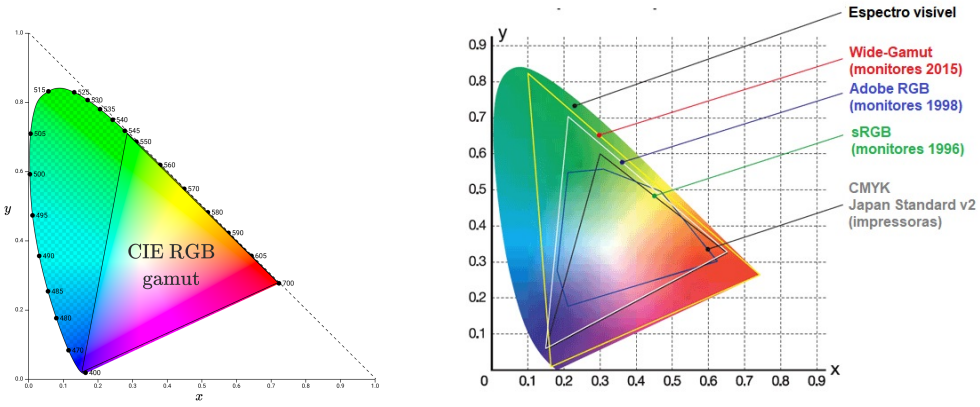



Figura 3.5: Espaço de cores CIEXYZ representado em duas dimensões.

A gama de cores, ou *Color gamut* (Smith [56]), (Gozalez e Woods [32]), que determina um intervalo de cores de forma mais representativa a partir do intervalo de cores visível ao olho humano, o chamado espectro visível, de diferentes dispositivos, pode ser representada sobre o gráfico do modelo XYZ. Por exemplo, um monitor com padrões sRGB e Adobe RGB tem diferentes gamas de cores, outro modelo de cores o CMYK comum em dispositivos como impressoras tem um gama de cores geralmente ainda mais limitado do que monitores, entretanto, todos eles podem ser representados dentro do espectro visível (Figura 3.6). O mesmo é válido para o modelo CIERGB.

### 3.2.3 O Modelo CIE L*a*b*

A percepção de cores pelo olho humano é gigantesca, porém, duas pessoas podem observar uma mesma imagem e enxergar cores diferentes mesmo que nenhuma delas apresente sintomas de daltonismo ou outras complicações referentes à visão. Isso deve-se ao fato da percepção das cores por cada indivíduo não ser a mesma. Muitas imagens que exemplificam esse evento ficaram famosas nas redes sociais nos últimos anos, nas quais,




(a) CIERGB *Gamut* (b) Demais *Gamuts*

Figura 3.6: Visualização de diversos gamas de cores sobre a representação bidimensional do modelo XYZ.

parte dos usuários enxergavam uma certa combinação de cores para um vestido ou um tênis, e outros enxergavam cores diferentes (vide Figura 3.7).

Figura 3.7: Ilusões de ótica populares na *internet*.

Segundo o site de tecnologia *wired.com*, na matéria "*The Science of Why No One Agrees on the Color of This Dress*", isso acontece pela forma como os olhos e o cérebro evoluíram para interpretar cores sob luz solar. Assim, pela percepção de um indivíduo, o mesmo objeto tem tonalidades diferentes em diferentes horários do dia de acordo com a quantidade de luz solar que está sendo aplicada a ele. Isso significa que o cérebro tenta "descontar" a intensidade da luz solar ou qualquer outra fonte de luz para definir uma cor que a considere como sendo "verdadeira".

Com a intenção de facilitar a avaliação e expressar de forma mais precisa a cor de um objeto para diferentes pessoas, foram criados sistemas de cores como o CIE L*a*b*. Tal modelo de cores baseia-se na versão original de Hunter (1948), ambos os modelos Hunter Lab e CIE L*a*b* são derivados do modelo CIEXYZ de 1931. A maior diferença entre eles é que as coordenadas do modelo CIE L*a*b* são criadas a partir de uma transformação de raiz cúbica das informações de cor do espaço CIEXYZ (Equação (3.18)), enquanto o Hunter Lab utiliza uma transformação por raiz quadrada. Apesar de haver uma certa discordância entre o público entre as cores cinza e azul com rosa e branco para o tênis e azul e preto com branco e dourado para o vestido na Figura 3.7, se utilizarmos a ferramenta

conta-gotas para expandir as cores a partir de um software de edição de imagens, tais cores ficaram bem mais definidas como ilustrado na Figura 3.8.

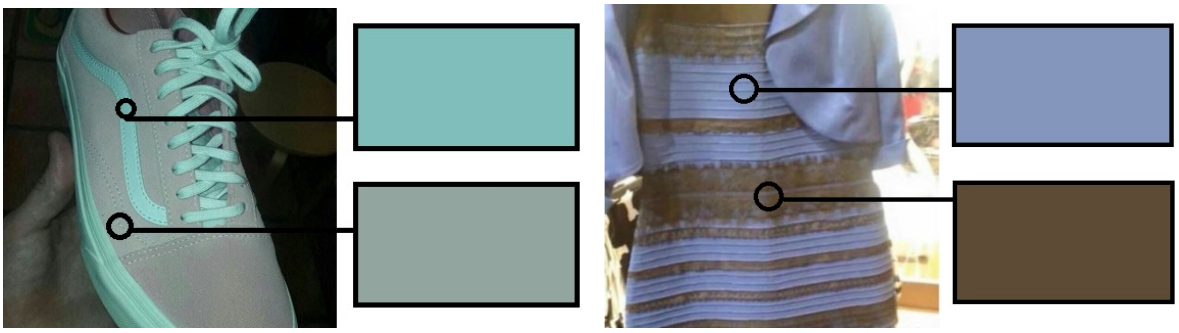
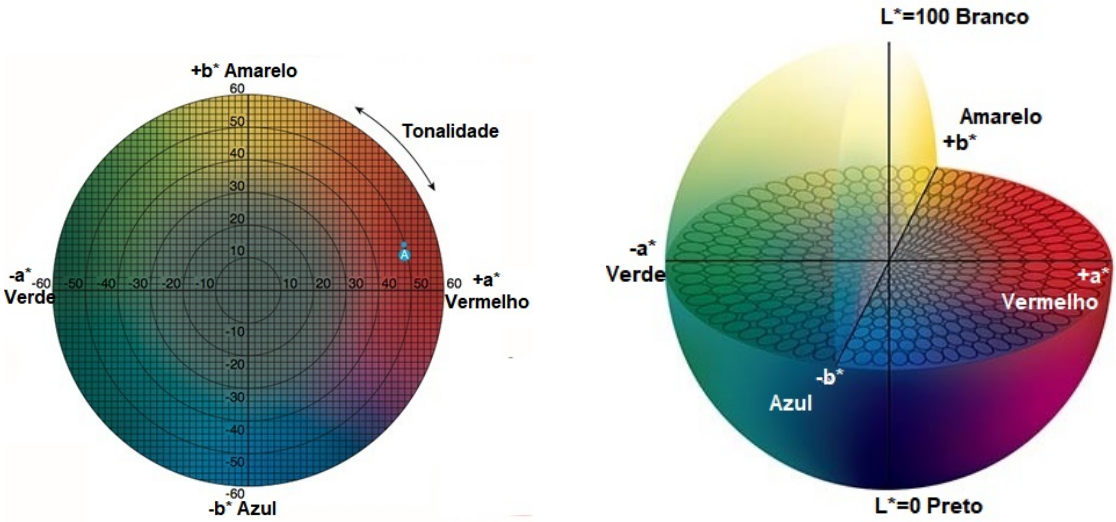


Figura 3.8: Medição e expansão das cores presentes nas figuras acima dentro da área circulada com a utilização da ferramenta conta gotas.

O CIE L*a*b* é, nos dias de hoje, o mais popular dentre os modelos de cores uniformes utilizados ao avaliar cores. Essa grande popularização deve-se ao fato do modelo correlacionar coerentemente os padrões de cores com a percepção visual do olho humano. Quando ordenadas, cores podem ser representadas em termos de tonalidade, luminosidade e saturação. Sendo assim, se definidas escalas para estes atributos, pode-se então determinar cores de modo bastante preciso.

Este modelo foi criado a partir da teoria de cores opostas, que diz que duas coordenadas de uma cor não podem assumir ao mesmo tempo tons de verde e vermelho, ou tons de amarelo e azul. Nesse modelo, a luminosidade é representada por $L^*$ onde $L^* = 0$ representa o preto e $L^* = 100$ o branco, os tons de verde a vermelho são representados sobre um eixo $a^*$, e os tons de azul a amarelo sobre um eixo $b^*$, como pode ser visto na Figura 3.9.




(a) $L^*a^*b^*2D$ (b) $L^*a^*b^*3D$

Figura 3.9: Representação do modelo $L^*a^*b^*$ em um circulo (2D) e uma esfera (3D).

Desta forma, as coordenadas de cromaticidade da cor são representadas pela combinação de a* com b*, e a intensidade ou luminosidade da mesma é dada por L*. Com relação ao dimensionamento e os extremos dos eixos a* e b*, isso dependerá especificamente da implementação da cor no modelo L*a*b*. Na representação em duas dimensões

da Figura 3.9, é possível perceber que os limites são $-60$ e 60 para ambos os eixos, no entanto, esse número pode variar de acordo com a utilização do modelo.

A maior vantagem do modelo CIELAB sobre o modelo RGB é que a cor LAB foi projetada para se associar à visão do olho humano, pois almeja a uniformidade perceptual, além de auxiliar o componente de luminosidade L* que é muito semelhante à percepção humana. Entretanto, este não considera o efeito Helmholtz-Kohlrausch (Donofrio [27]). Desta forma, o modelo é capaz de realizar correções de equilíbrio de cor de forma mais precisa apenas mudando as curvas de saída dos eixos a* e b*, ou regulando a componente de luminosidade L*, diferindo dos modelos como o RGB, no qual esse tipo de transformação só pode ser realizada com a ajuda de modos de mesclagem próprios de aplicativos de edição.

Felizmente para os propósitos deste trabalho, com a chegada dos suportes de 16 *bits* por canal e de ponto flutuante, realizar conversões entre modelos de cores como RGB e CIELAB se tornou muito mais eficiente do que foi na década de 1990, quando as máquinas e *softwares* eram restritos a armazenar e manipular *bitmaps* de 8 *bits* por canal, o que tornava esse tipo de conversão passível de falhas. Por outro lado, hoje perdas por causa da quantização nesse tipo de operação acabam se tornando desprezíveis.

### 3.2.4 Conversão RGB para CIE L*a*b*

A conversão entre esses dois modelos de cores, passa pelo modelo XYZ e é dividida em duas etapas. Na primeira etapa, deve ser realizada a conversão do RGB para o XYZ da seguinte maneira:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.13}$$

na qual $M$ é uma matriz de coeficientes $3 \times 3$ que varia de acordo com o modelo RGB específico que está sendo convertido. No caso do modelo padrão, temos:

$$M = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \tag{3.14}$$

E assim está concluído o primeiro passo. O segundo passo apresenta uma complexidade maior, porém, não traz grandes desafios. Em resumo: agora deve ser conduzida a conversão do modelo XYZ para o modelo L*a*b* e, neste caso, os valores de L*, a* e b* podem ser definidos por funções, da seguinte forma:

$$L^* = 116 \cdot f\left(\frac{Y}{Y_n}\right) - 16 \tag{3.15}$$

$$a^* = 500 \cdot \left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \tag{3.16}$$

$$b^* = 200 \cdot \left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \tag{3.17}$$

onde a função $f$ é definida como segue:

$$f(t) = \begin{cases} \sqrt[3]{t}, & \text{se } t \geq \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29}, & \text{caso contrário} \end{cases}, \tag{3.18}$$

em que $\delta = 6/29$ e os valores de $X_n$, $Y_n$ e $Z_n$ referem-se ao ponto de cor branca do modelo, que também muda de acordo com variações do modelo XYZ, mas para alguns exemplos clássicos como o iluminante $D65$ temos que:

$$X_n = 95,047; \quad Y_n = 100; \quad Z_n = 108,883. \tag{3.19}$$

e para o iluminante $D50$ segue que:

$$X_n = 96,6769; \quad Y_n = 100; \quad Z_n = 82,5188. \tag{3.20}$$

Assim, em nossa abordagem, a imagem, inicialmente definida na escala de cores RGB, é convertida para o sistema L*a*b* de forma que as bandas coloridas resultantes desse processamento são computadas na determinação dos pesos do grafo de afinidade da imagem.

## 3.3 Extração de Contornos da Imagem

### 3.3.1 Variação Local de uma Imagem

Quando trabalhamos com uma imagem suavizada, uma ferramenta de grande auxílio para destacar de forma mais clara os objetos presentes na imagem são as funções de variação local, mais especificamente funções que medem a variação, ou a distância entre pixels em uma certa vizinhança.

Seja $I$ a imagem-alvo e $P_i$ um pixel de $I$. Considere, também, $N_r(P_i)$ como sendo a vizinhança $r \times r$ em torno do pixel $P_i$, sendo $r$ um número ímpar, e seja também $x_i$ com $i = 1, ..., n$ $(n \leq r^2)$ os elementos dessa vizinhança. Então, a função de distância local da imagem $I$, também conhecida por *rangefilt* no *software* MATLAB, atribui ao pixel da matriz de saída da função o valor resultante da seguinte relação:

$$P_i = \max_{1 \leq i \leq n}(x_i) - \min_{1 \leq i \leq n}(x_i). \tag{3.21}$$

Ou seja, para cada elemento da matriz $I$, é avaliada a máxima distância entre dois pixels da sua vizinhança, e este valor é atribuído ao elemento que está sendo analisado. A Figura 3.10 ilustra processo descrito.

Como pode ser observado na Figura 3.11, aplicar uma função que mede a maior distância entre dois pixels na região local de uma imagem pode ajudar a destacar os contornos e a superfície dos objetos presentes na mesma.

A utilização desse tipo de tratamento é possível tanto em imagens de duas dimensões, em tons de cinza, quanto em imagens de três dimensões, coloridas. Quando usada em uma

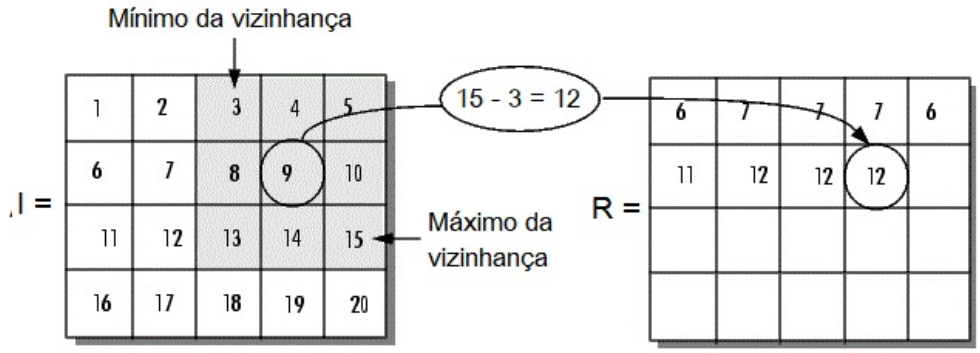



Figura 3.10: Representação do processo da função de variação de pixels.

(a) Imagem Original (b) *Rangefilt* $r = 3$ (c) *Rangefilt* $r = 5$

Figura 3.11: Aplicação da função de variação local para uma vizinhança $r \times r$.

imagem de duas dimensões, o processo da função de distância local vai ser exatamente tal como descrito acima e ilustrado pela Figura 3.11.

Além disso, tal processo ocorrerá de forma muito semelhante quando aplicado em imagens de três dimensões, porém, neste caso, uma abordagem que produz resultados satisfatórios é aplicar a função em cada uma das bandas da imagem. Por exemplo, se a imagem em questão for colorida, ela será divida em matrizes bidimensionais que, compostas, formam a imagem original. Assim, cada uma dessas matrizes passará pelo processo de análise de distância entre elementos de uma vizinhança de forma que a solução final será dada pela composição de cada banda avaliada como na Figura 3.12.

### 3.3.2 Detector de Bordas Canny

Esta técnica de detecção de bordas, desenvolvida pelo pesquisador Australiano John F. Canny [16], foi projetada com a intenção de melhorar algumas falhas em técnicas anteriores. Basicamente, os principais objetivos almejados por Canny eram:

- Diminuir a taxa de erro na detecção de bordas;
- Os pontos considerados como borda deveriam estar bem localizados;
- Reduzir ruídos causados quando mais de um ponto em uma região era considerado uma borda.

Encontrar uma solução que satisfaça todos os ítens supracitados é extremamente difícil, porém, Canny trabalhou a partir de pesquisas anteriores de detecção de bordas e concluiu

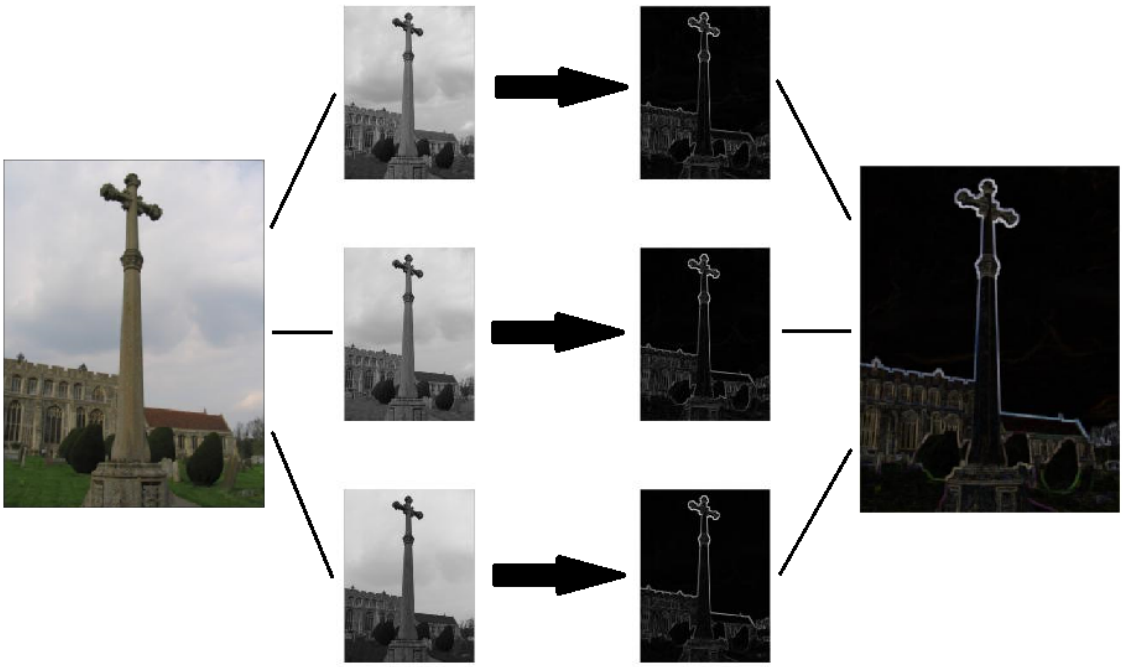


Figura 3.12: Aplicação da função de variação local para imagens coloridas.

que uma boa aproximação seria dada pela derivada de uma Gaussiana (Gonzalez e Woods [32]) da seguinte forma:

$$\frac{d}{dx}e^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}} . \tag{3.22}$$

Se considerarmos um problema em duas dimensões, então temos que a Gaussiana a ser derivada e posteriormente utilizada nos cálculos pode ser representada por:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} . \tag{3.23}$$

A partir dessas observações, foi desenvolvido um algoritmo *multi-stage*, o qual tem seus passos expressos de formas variadas por diferentes autores, porém descrevem basicamente o mesmo processo. Por se tratar de um algoritmo com vários passos, vamos separar cada um deles de forma a descrever melhor o desenvolvimento do extrator de bordas Canny.

### Passo 1

A primeira etapa do algoritmo é a realização da suavização via convolução Gaussiana na imagem que se quer extrair as bordas. Este processo é também conhecido por *Gaussian Filter* (Gonzalez e Woods [32]), ou *Gaussian Blur* (Gedraite e Hadad [31]), pois métodos de detecção de bordas são suscetíveis a ruídos quando examinados os pixels nos limites das bordas e, portanto, a suavização tende a reduzir este tipo de problema.

Alguns autores também trabalham com processos de suavização específicos para o detector de bordas Canny como no artigo de Bourennane e colaboradores [12].

(a) Imagem Original (b) Imagem Suavizada

Figura 3.13: Processo de suavização por convolução Gaussiana de uma imagem para o extrator de bordas Canny

**Passo 2**

Com a imagem já suavizada, calcula-se o gradiente da imagem através de um filtro que retorna a primeira derivada de $G(x, y)$ quando aplicado na horizontal ($G_x$) e na vertical ($G_y$). Cada um desses filtros gera uma imagem $G_x$ e $G_y$, e o gradiente de bordas $G$ pode ser expresso então da seguinte maneira:

$$G = \sqrt{G_x^2 + G_y^2} \tag{3.24}$$

O gradiente é sempre perpendicular à borda.

**Passo 3**

Após o passo anterior, a imagem resultante terá bordas grossas, mas o objetivo é que essas bordas sejam finas e bem definidas, portanto deve-se realizar uma supressão por máximo local, onde cada ponto de borda será avaliado com outros dois do gradiente e apenas os pontos avaliados que representarem o máximo dos três serão mantidos para o próximo passo, isto é, os demais serão anulados. A Figura 3.14 ilustra o procedimento.

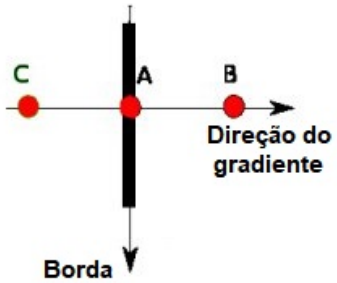



Figura 3.14: Supressão de pontos não máximos.

O ponto A é um ponto que foi considerado de borda da imagem na vertical, logo tem seu gradiente na horizontal. Os pontos B e C são pontos na direção do gradiente, então é verificado se A é o máximo de $(A, B, C)$, então A continua como candidato a ponto de borda, caso contrário, é descartado antes do início do próximo passo.

**Passo 4**

Após os passos anteriores, alguns pontos considerados pertencentes a borda da imagem podem ainda serem ruídos. Assim, neste passo, todos os pontos remanescentes dos testes anteriores são definidos como elementos pertencentes a borda, ou não. Para tal, é preciso definir pelo menos dois valores de *threshold*, um valor mínimo $a$, e um valor máximo $b$, com $a, b$ pertencentes ao intervalo aberto $(0, 1)$. Uma boa escolha para os valores de $a$ e $b$ é importante. Desta forma, para melhor resultado do algoritmo original, existem aprimoramentos como em Wenjun e colaboradores [62], cujos autores implementaram o detector de bordas Canny juntamente com o algoritmo que determina os melhores valores de *threshold* via método de Otsu [52].

Os pixels que possuem valor maior do que $b$ são chamados de "bordas forte" e são mais prováveis de realmente pertencer à borda da imagem, logo são automaticamente promovidos a ponto de borda pelo algoritmo. O oposto é válido para os pixels que possuem valor menor do que $a$, que são removidos da solução final. Os pontos que estão entre $a$ e $b$ são conhecidos como "bordas fracas", e devem ser avaliados antes de serem configurados com pertencentes ou não à borda da imagem.

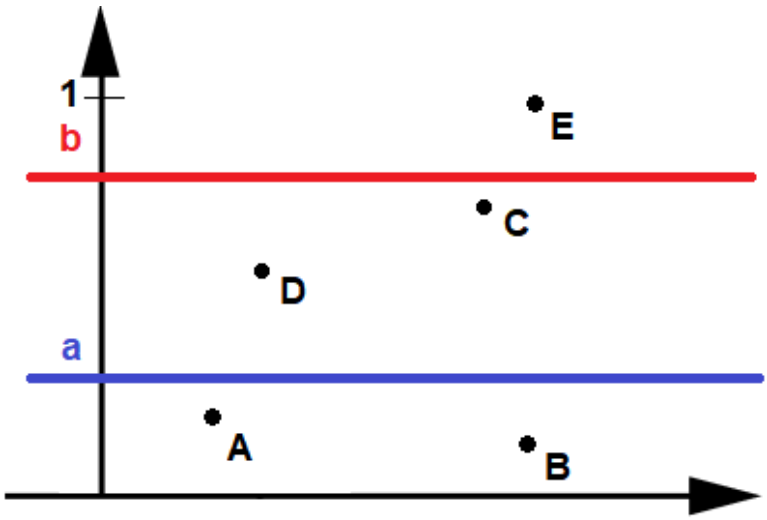



Figura 3.15: Esquema de remoção de pontos via *threshold*.

Na situação descrita na Figura 3.15, pode-se concluir que $A$ e $B$ não pertencem à solução, $E$ é um ponto de borda, e os pontos $C$ e $D$ vão ser avaliados para definir se fazem ou não parte da solução final do detector Canny. Para definir se um ponto é considerado uma borda fraca ou não é um ponto de borda da imagem, deve-se analisar se ele está conectado a um ponto de borda forte, se sim, então esse ponto é considerado parte da solução, caso contrário, se não há conexões com nenhum ponto deste tipo, ele então será descartado.

(a) Imagem Original (b) Bordas após Passo 3 (c) Solução

Figura 3.16: Bordas extraídas de uma imagem através do detector de bordas Canny.

A solução então é uma imagem binária onde apenas os pontos que foram definidos como bordas da imagem são exibidos e os demais pontos são considerados como 0. Na Figura 3.16 foram usados como valores de *threshold* $a = 0.075$ e $b = 0.175$.

Capítulo

# 4

# Resultados e Discussão

Antes de apresentarmos os resultados obtidos nessa pesquisa, iniciaremos nossa discussão a partir da descrição da base de dados utilizada nos testes assim como as métricas de qualidade aplicadas para avaliar o desempenho das segmentações. Tal como comumente abordado em outros trabalhos da literatura, os resultados serão divididos em dois tipos de análises distintas:

1. **Resultados Quantitativos:** onde serão compilados os resultados das métricas em forma de tabelas e gráficos, visando assim estabelecer comparações entre os diferentes *scores* obtidos em nossas análises.
2. **Resultados Qualitativos:** onde serão exibidos os resultados visuais, isto é, as imagens segmentadas com a utilização das novas combinações de *features* em comparação com resultados de métodos anteriores.

A análise dos resultados tal como proposto acima é capaz de demonstrar a efetividade da pesquisa com relação à melhora na precisão e acurácia do processo de segmentação.

## 4.1 Métricas de Avaliação e Base de Dados

### 4.1.1 Métricas de Qualidade

Para a obtenção dos resultados quantitativos, foram utilizadas cinco métricas as quais visam mensurar a qualidade final das segmentações. Assim, neste estudo, foram aplicadas as seguintes métricas de avaliação de qualidade:

- ***Rand Index*** **(RI).**
- ***Global Consistency Error*** **(GCE).**
- ***Variation of Information*** **(VI).**
- ***Boundary Displacement Error*** **(BDE).**
- ***Dice Similarity Coefficient*** **(DICE).**

O cálculo dessas métricas retornam valores escalares que representam uma avaliação da precisão com relação aos resultados da segmentação, porém, cada uma explora na imagem algum tipo de informação específica ou padrão de particionamento. Mais especificamente, cada métrica leva em consideração alguma característica particular da imagem como região de segmentação ou as delimitações da fronteira do objeto-alvo. A depender do tipo de métrica empregada, seus valores indicam melhores segmentações caso se obtenha um valor mínimo na mensuração ou, ainda, um valor que seja máximo. Essa informação é indicada por setas nos gráficos e tabelas de resultados.

Ao aplicar tais métricas, o programa compara o resultado da segmentação obtida pelo modelo com as respectivas imagens previamente segmentadas por uma ou um conjunto de pessoas. Assim, neste contexto, tais imagens são denominadas *Ground-Truth Images* (GT). De fato, essas imagens são consideradas como sendo o resultado "esperado" a ser atingido pelos segmentadores aplicados.

A fim de entendermos melhor como cada métrica impacta na imagem segmentada de saída, será realizada uma breve descrição do processo que cada métrica leva em consideração para comparar uma segmentação com seu respectivo *ground truth*.

### Rand Index (RI)

Essa métrica tem como finalidade medir a proximidade entre a segmentação obtida, denotada aqui por $S$, e o respectivo $GT$ da imagem de entrada. Tal comparação é feita levando em consideração o número de pares de pixels associados em $S$ e $GT$, que possuem o mesmo rótulo. De fato, a métrica (RI) era utilizada, a princípio, apenas para avaliar resultados de técnicas de *clustering*, sendo modificada por Unnikrishnan e colaboradores [60], e Yang e colaboradores [64] para aplicações que viessem a contemplar outras abordagens na área de segmentação.

Basicamente se, neste caso, denotarmos um par de pixels por $S_i \in S$ e $G_i \in GT$, ambos de mesmo índice, ou seja, na mesma posição com relação à imagem base, o valor do *Rand Index* (Rand [54]) pode ser então computado da seguinte maneira:

$$RI = \frac{\text{Pares com mesmo rótulo}}{\text{Total de pares}}. \tag{4.1}$$

Note que, tem-se que $0 \leq RI \leq 1$, e também que $100 \cdot RI$ representa a porcentagem do total de pixels da imagem que foram segmentados conforme esperado, logo quanto maior o valor de $RI$, melhor será o resultado da segmentação segundo tal métrica.

### Global Consistency Error (GCE)

Essa métrica de qualidade avalia o quanto uma segmentação é um refinamento de outra, e faz com que todos os refinamentos correspondam a um mesmo padrão. Assim, uma característica predominante da GCE é que tal métrica não leva em consideração as escalas das imagens que estão sendo comparadas, mas sim apenas suas semelhanças em um contexto mais geral. Além disso, a GCE permite a análise do refinamento de ambos os pontos de vista, isto é, de uma imagem $A$ para a imagem $B$, e vice-versa, garantindo, desta forma, consistência mesmo quando aplicada em imagens de diferentes escalas (Martin e colaboradores [44]). Portanto, quanto menor for o escore resultante da GCE, melhor será o resultado da segmentação em comparação com seu *ground truth*.

Antes de calcular o valor escalar da GCE, é preciso definir o erro de refinamento $E$ (Iancu e colaboradores [38]), o qual é dado por:

$$E(S, GT, p_i) = \frac{|R(S, p_i) \backslash R(GT, p_i)|}{|R(S, p_i)|} \tag{4.2}$$

onde $R(S, p_i)$ representa o conjunto de pixels correspondentes à região na segmentação $S$ que contém o pixel $p_i$; analogamente para $R(GT, p_i)$. Sendo então $n$ o número de pixels das imagens, o valor da GCE é definido por:

$$GCE(S, GT) = \frac{1}{n} min \left\{ \sum_i E(S, GT, p_i), \sum_i E(GT, S, p_i) \right\} . \tag{4.3}$$

### Variation of Information (VI)

A métrica *Variation of Information*, como o próprio nome do inglês sugere, computa a variação da informação. Na prática, ela procura medir a distância com relação à entropia do resultado da segmentação e do GT, ou seja, esta métrica quantifica a quantidade de informação desnecessária entre as duas imagens, logo o que esperamos como resultado do uso dessa métrica é que a mesma seja a menor possível. Além disso, pode-se demonstrar que a VI é uma métrica fundamentada em conceitos clássicos da Álgebra Linear (Meila [47], Mignotte [48]).

Sendo assim, o valor escalar da VI (Benes e Zivotá [9]) é dado por:

$$VI = H(S) + H(GT) - 2M(S, GT) , \tag{4.4}$$

onde $H(S)$ e $H(GT)$ são os valores de entropia relacionados à segmentação e ao *ground-truth*, e $M(S, GT)$ representa as informações que $S$ e $GT$ compartilham, chamada de *Mutual Information*.

### Boundary Displacement Error (BDE)

A ideia acerca da (BDE) é relativamente simples: ela leva em consideração em sua análise os pixels pertencentes à fronteira do objeto e, com relação as estes, calcula o erro de deslocamento das posições que eles assumem no resultado da segmentação e no (GT) (Unnikrishnan e colaboradores [59]).

Quanto maior o valor obtido pela aplicação da (BDE), maior será o deslocamento dos pixels de fronteira com relação à segmentação esperada, (GT), logo valores menores indicam melhores resultados de segmentação.

### Dice Similarity Coefficient (DICE)

Essa métrica de similaridade baseia-se na distância Sørensen [57] que, quando comparada à distância Euclidiana, conta com a diferença de reduzir o peso de valores extremos dentro de um determinado conjunto de dados, tornando o cálculo mais sensível quando se trata de conjuntos mais heterogêneos.

O coeficiente DICE analisa a imagem de saída resultante da segmentação com o GT, e retorna um valor escalar entre 0 e 1. Se o grau de similaridade for 1, isso representa que as duas segmentações comparadas são idênticas, logo espera-se obter sempre valores que sejam próximos de 1.

Matematicamente, o coeficiente DICE (Benes e Zitová [9]) é obtido através da seguinte equação:

$$DC = \frac{2|S_O \cap GT_O|}{|S_O| + |GT_O|}, \tag{4.5}$$

na qual $S_O$ e $GT_O$ representam os elementos da imagem segmentada e do *ground-truth* rotulados como "objeto", respectivamente. E ainda, na equação (4.5) $|S_O|$ e $|GT_O|$ representam o número de elementos presentes nos conjuntos $S_O$ e $GT_O$.

### 4.1.2 Base de Dados de Imagens/Seeds

No intuito de estabelecer comparações com outros métodos de segmentação interativo, a análise realizada neste trabalho toma como referencial uma base de imagens clássica da literatura de segmentação, a qual conta com uma grande variedade de imagens coloridas contendo diversas características a serem exploradas como cenários, pessoas, animais, paisagens, etc. O objetivo é aplicar o método de segmentação das Coordenadas de Laplace a fim de explorar diferentes configurações de *features*, possibilitando assim estabelecer um conjunto de bandas que atinja um maior grau de acurácia para o segmentador analisado.

Mais precisamente, o *dataset* utilizado para os testes neste trabalho foi o clássico *GrabCut* [55], da Microsoft. Essa base é constituída por um conjunto de 50 imagens públicas e seus respectivos *Ground-Truth*. Uma vez que o conjunto padrão de *seeds* do *GrabCut* contempla apenas marcações densas para as imagens, optamos por utilizar como *dataset* complementar de *seeds* aquele apresentado por Andrade e Carrera [2]. Para uma ilustração do *dataset* de imagens do *GrabCut*, vide Figura 4.1, bem como a Figura 4.2 para uma ilustração de alguns dos mapas de *seeds* descritos por Andrade e Carrera [2].

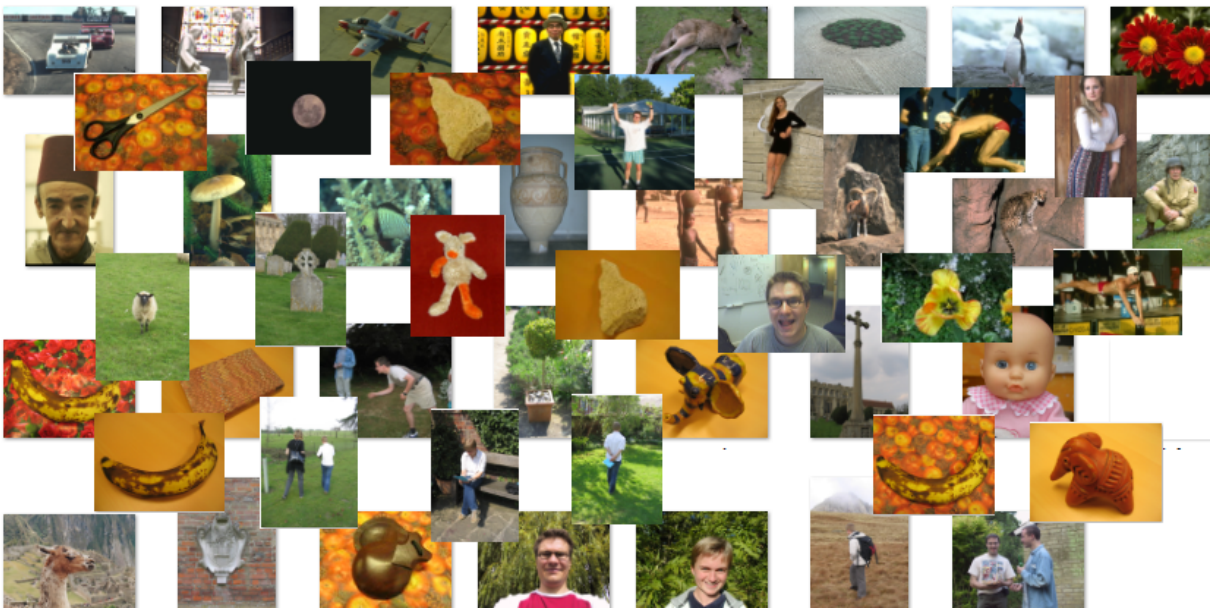


Figura 4.1: Base de imagens do *GrabCut dataset*.

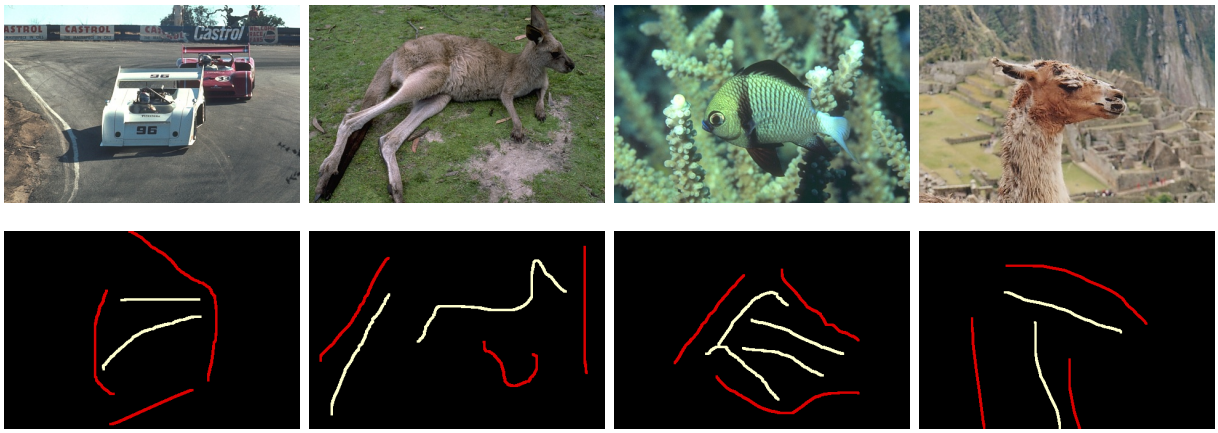


Figura 4.2: Exemplos de imagens do *GrabCut dataset* exibido na Fig. 4.1 e suas respectivas *seeds* sugeridas.

## 4.2 Experimentos e Comparações com Diferentes Conjuntos de *Features*

### 4.2.1 Resultados Quantitativos

Nesta seção, são exibidos os resultados dos testes com a base descrita anteriormente a partir da aplicação das cinco métricas mencionadas na Seção 4.1.1, a saber: *Rand Index* (**RI**), *Global Consistency Error* (**GCE**), *Variation of Information* (**VI**), *Boundary Displacement Error* (**BDE**) e *Dice Similarity Coefficient* (**DICE**). Assim, para uma melhor compreensão dos valores das métricas, aquelas que apresentam o melhor resultado no sentido de "quanto maior, melhor" estão acompanhadas do símbolo (↑) nas tabelas, enquanto que aquelas que estabelecem "quanto menor, melhor" estão seguidas do símbolo (↓). Todos os cálculos das métricas levam em conta as sumarizações de todas as 50 imagens coloridas (*Grabcut dataset* - Seção 4.1.2), as quais são submetidas aos testes de qualidade a partir do uso de diferentes conjuntos de *features*.

#### Teste com Conjunto de *Features* #1

Em nosso primeiro teste experimental, foram utilizadas imagens suavizadas por intermédio da operação de convolução. Assim, vamos então definir essas imagens como $I^{(s)}$. Em seguida, foi realizada a conversão de $I^{(s)}$ para o sistema de cores $L^*a^*b^*$, resultando nas bandas $I^{(s)}_{Lab}$. Desta forma, foram então empregadas como *features* de entrada para o cálculo dos pesos da matriz $W$, na equação (3.1), as 3 bandas de $I^{(s)}$, juntamente com as 3 bandas de $I^{(s)}_{Lab}$, as quais foram todas, na sequência, submetidas à função de distância local (*rangefilt* no MATLAB) com diferentes parâmetros de vizinhança $r \times r$. Os resultados obtidos encontram-se na Tabela 4.1.

| **Métricas** | | $r = 3$ | $r = 5$ |
|---|---|---|---|
| RI | (↑) | 0.9048 | 0.9257 |
| GCE | (↓) | 0.0765 | 0.0636 |
| VI | (↓) | 0.4488 | 0.3801 |
| BDE | (↓) | 8.5205 | 6.6576 |
| DICE | (↑) | 0.9222 | 0.9394 |
| Tempo ($s$) | | 166 | 170 |

Tabela 4.1: Resultados obtidos pelo conjunto de *features* #1.

Analisando os *scores* listados na Tabela 4.1, é possível observar que, para esse conjunto específico de *features*, as métricas em geral apresentaram resultados mais acurados quando empregado o parâmetro $r = 5$ na função *rangefilt*. Com base nos resultados obtidos nessa primeira bateria de testes, nos testes seguintes, foram empregados a distância local de uma vizinhança $5 \times 5$ para os pixels, ou seja, foi adotado o parâmetro $r = 5$ no método de distância local *rangefilt*. É importante ressaltar que o tempo que aparece na ultima linha da Tabela 4.1, assim como nas tabelas que seguem neste capítulo, é referente ao conjunto de 50 imagens descrito na seção 4.1.2.

### Teste com Conjunto de *Features* #2

Em nosso próximo teste de análise de *features*, foram descartadas as bandas de entrada descendentes da matriz $I^{(s)}$, mas preservadas as da matriz $I^{(s)}_{Lab}$, que desta vez foram combinadas com as *features* resultantes da imagem suavizada pré-segmentada pelo modelo de Mistura Gaussiana, GMM, e o Algoritmo EM a fim de explorar diferentes parâmetros $k$ do GMM, o qual foi denotado aqui por $G_{(k)}$.

Desta forma, note que as *features* de entrada para a matriz $W$ agora são as 3 bandas da imagem $I^{(s)}_{Lab}$ combinadas com as 3 bandas de $G_{(k)}$, que também foram submetidas à função *rangefilt* para que houvesse destacamento de bordas com parâmetro $r = 5$, estabelecido em todos os casos testados. Os resultados obtidos foram reportados na Tabela 4.2.

| **Métricas** | | $k = 2$ | $k = 3$ | $k = 5$ | $k = 8$ | $k = 15$ |
|---|---|---|---|---|---|---|
| RI | (↑) | 0.9248 | 0.9100 | 0.9163 | 0.9181 | 0.9227 |
| GCE | (↓) | 0.0655 | 0.0725 | 0.0698 | 0.0689 | 0.0662 |
| VI | (↓) | 0.3920 | 0.4324 | 0.4147 | 0.4071 | 0.3937 |
| BDE | (↓) | 6.4274 | 7.8427 | 7.7085 | 7.5599 | 7.0764 |
| DICE | (↑) | 0.9390 | 0.9259 | 0.9311 | 0.9325 | 0.9368 |
| Tempo ($s$) | | 201 | 231 | 321 | 465 | 1053 |

Tabela 4.2: Resultados utilizando o Algoritmo EM como pré-segmentação de imagens via Modelos de Mistura.

Conforme pode ser visualizado na Tabela 4.2, os resultados utilizando as *features* relacionadas ao algoritmo EM apresentaram melhor desempenho para essa combinação com $I^{(s)}_{Lab}$ quando $k = 2$. É possível também notar que, após um resultado satisfatório com $k = 2$, as métricas caem de rendimento para o próximo $k$, porém, vão se re-estabelecendo conforme o parâmetro $k$ é aumentado, quase comportando-se como uma função logarítmica. No entanto, é importante salientar que o tempo de processamento também aumenta, porém quase que exponencialmente. Em suma, a partir da adoção do parâmetro $k = 2$,

tem-se o melhor *trade-off* para a utilização dessa combinação de *features* visando não só a melhor qualidade na segmentação, como também um custo computacional viável.

Na combinação das bandas de $I^{(s)}_{Lab}$ com $G_{(2)}$, embora as métricas não sejam melhores "quantitativamente falando" do que os resultados obtidos na Tabela 4.1, as *features* geradas na utilização do EM promovem resultados mais acurados com as que foram introduzidas nos testes seguintes, trazendo assim resultados de maior precisão na segmentação do que aqueles originalmente oferecidos pelas bandas de $I$ na Tabela 4.1.

### Teste com Conjunto de *Features* #3 até #6

Os testes a seguir contam com novas técnicas ainda não utilizadas até então, além do realce de contornos com a utilização da função *rangefilt* e da conversão de matrizes para o tipo $L^*a^*b^*$. Além disso, foram também introduzidos os métodos de detecção de bordas como o *Canny*, tal como visto na Seção 3.3.2, com vetores de parâmetros *threshold* e *sigma* fixos. A ideia é então explorar diferentes tipos de combinações de *features* de forma a obter a melhor delas na tarefa de segmentação a partir do uso do segmentador das Coordenadas de Laplace.

Os próximos quatro testes foram realizados com diferentes composições de *features*, utilizando-se todas as técnicas discutidas no Capítulo 3. Tais composições de *features* foram representadas por: Composição $(a)$, Composição $(b)$, Composição $(c)$ e Composição $(d)$, e foram construídas utilizando de 4 a 7 das *features* descritas a seguir:

- ***Feature* 1:** Primeira banda da imagem suavizada $I^{(s)}$ convertida para o formato $L^*a^*b^*$, $I^{(s)}_{Lab}$.
- ***Feature* 2:** Segunda banda de $I^{(s)}_{Lab}$.
- ***Feature* 3:** Terceira banda de $I^{(s)}_{Lab}$.
- ***Feature* 4:** Resultado da combinação das três bandas da imagem resultante do algoritmo EM para $k = 2$, $G_{(2)}$, em uma única *feature* que representa a média de todas as bandas que compõem $G_{(2)}$.
- ***Feature* 5:** Resultado da aplicação do detector de bordas Canny na imagem em formato RGB. Tal procedimento gera 3 bandas com bordas destacadas que foram combinadas em uma única *feature*, representando a média das 3.
- ***Feature* 6:** Processo análogo ao da ***feature* 5**, porém, para a imagem no formato $L^*a^*b^*$.
- ***Feature* 7:** Resultado da combinação das ***features* 5 e 6** para gerar uma nova *feature* com bordas fortes, mas também rica em detalhes.
- ***Feature* 8:** Resultado da aplicação da função de distância local com parâmetro $r = 3$ na ***feature* 7**, com intenção de destacar suas bordas.

| ***Features*** $\rightarrow$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Composição $(a)$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | | | | |
| Composição $(b)$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | | |
| Composição $(c)$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | |
| Composição $(d)$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | $\otimes$ | | $\otimes$ |

Tabela 4.3: Composições de *features*.

Como observado anteriormente, cada *feature* se trata de uma imagem que carrega características relevantes sobre as bordas da imagem. A Figura 4.3 mostra o resultado de cada uma das *features* utilizadas na composição que resultou as melhores mensurações para as métricas de qualidade.

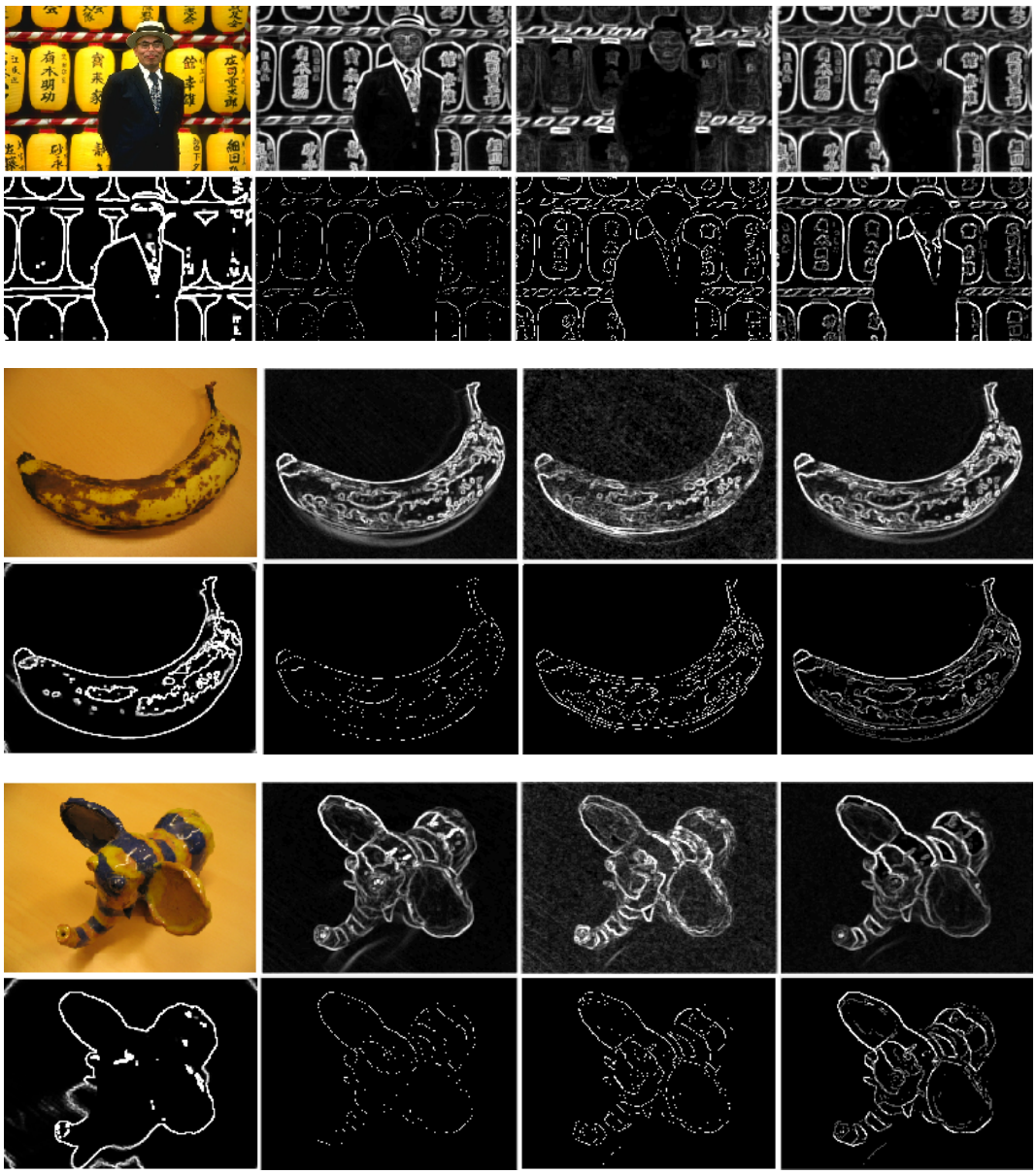


Figura 4.3: Imagens submetidas ao processo de segmentação, seguidas pelas 7 *features* descritas na composição $(d)$.

Os resultados da aplicação de cada composição conforme descritas anteriormente são apresentados na Tabela 4.4.

| **Métricas** | | $(a)$ | $(b)$ | $(c)$ | $(d)$ |
|---|---|---|---|---|---|
| RI | $(\uparrow)$ | 0.9216 | 0.9352 | 0.9357 | 0.9386 |
| GCE | $(\downarrow)$ | 0.0663 | 0.0575 | 0.0570 | 0.0550 |
| VI | $(\downarrow)$ | 0.3979 | 0.3492 | 0.3479 | 0.3356 |
| BDE | $(\downarrow)$ | 6.5259 | 5.6398 | 5.6731 | 5.2602 |
| DICE | $(\uparrow)$ | 0.9364 | 0.9479 | 0.9480 | 0.9508 |

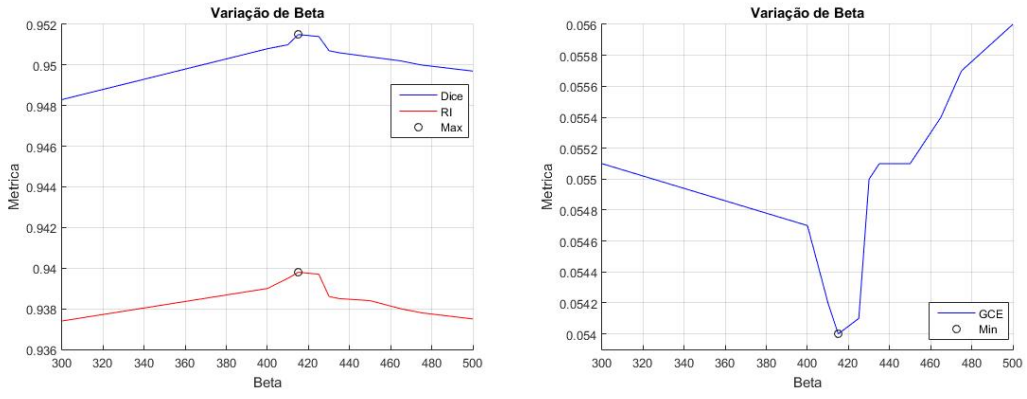
Tabela 4.4: Resultados introduzindo novas composições de *features*.

Com base nos valores reportados na Tabela 4.4, quase todas as composições apresentaram resultados bastante acurados e competitivos em comparação com os *scores* obtidos pelo método com a composição de bandas original. Ainda que na Coluna $(a)$ da Tabela 4.4, os resultados não foram melhores que os obtidos na Tabela 4.1, nos demais testes, reportados a partir da Coluna $(b)$ com a introdução do detector de bordas *Canny* (Seção 3.3.2), é possível observar que houve uma melhora significativa nos *scores* das métricas utilizadas.

Já em termos comparativos envolvendo apenas as quatro combinações de *features* adotadas, a última delas, Composição $(d)$, foi a que apresentou os melhores resultados até então com relação às métricas estudadas, embora com um tempo computacional mais elevado. Por exemplo, a DICE ficou na faixa de 0.95, o que leva a um ganho de qualidade considerável nas segmentações obtidas pelo método a partir dessa configuração de bandas.

É válido ainda ressaltar que um grande número de testes envolvendo as composições descritas na Tabela 4.3 também foram realizados durante a pesquisa, entretanto nenhuma configuração adicional superou a configuração apresentada na Composição $(d)$.

E finalmente, após determinar a melhor combinação de *features*, foi conduzido um estudo na variação do parâmetro $\beta$ na equação (2.11), ou na sua versão adaptada para diversas *features* (equação (3.1)), com o objetivo de melhorar ainda mais a qualidade da segmentação. O valor adotado nos testes iniciais foi $\beta = 475$, porém, quando estudados no intervalo $300 \leq \beta \leq 500$, as métricas de qualidade se comportaram como na Figura (4.4).




(a) Dice e Rand Index (b) Global Consistency Error

Figura 4.4: Comportamento das métricas de qualidade com a variação de $\beta$.

Desta forma, os melhores resultados obtidos neste trabalho foram alcançados utilizando a composição referente à coluna $(d)$, combinada com a utilização do parâmetro $\beta = 415$, resultando na evolução reportada na última coluna da Tabela 4.5.

| **Métricas** | | Inicial | $(a)$ | $(b)$ | $(c)$ | $(d)$ | Atual |
|---|---|---|---|---|---|---|---|
| RI | $(\uparrow)$ | 0.9257 | 0.9216 | 0.9352 | 0.9357 | 0.9386 | **0.9397** |
| GCE | $(\downarrow)$ | 0.0636 | 0.0663 | 0.0575 | 0.0570 | 0.0550 | **0.0540** |
| VI | $(\downarrow)$ | 0.3801 | 0.3979 | 0.3492 | 0.3479 | 0.3356 | **0.3305** |
| BDE | $(\downarrow)$ | 6.6576 | 6.5259 | 5.6398 | 5.6731 | 5.2602 | **5.1862** |
| DICE | $(\uparrow)$ | 0.9394 | 0.9364 | 0.9479 | 0.9480 | 0.9508 | **0.9515** |
| Tempo $(s)$ | | **170** | 190 | 280 | 288 | 293 | 298 |

Tabela 4.5: Evolução dos *scores* no desenvolvimento deste trabalho.

Para melhor entender o aumento no tempo de processamento, foi levantado para cada conjunto de *features*, o tempo gasto para suas devidas produções, levando em consideração uma imagem de dimensão $640 \times 533$ pixels, como mostrado na Tabala 4.6.

Na primeira coluna da tabela, cada linha está separada de acordo com a ordem em que as *features* foram criadas no processo de segmentação, sendo as linhas compostas por mais de uma *feature* relativas aos conjuntos que foram criados ao mesmo tempo.

| Conjuntos | | Tempo gasto $(s)$ |
|---|---|---|
| ***features*** | **1, 2 e 3** | 0.1046 |
| ***feature*** | **4** | 0.4648 |
| ***features*** | **5 e 6** | 3.9519 |
| ***feature*** | **7** | 0.0001 |
| ***feature*** | **8** | 0.0201 |

Tabela 4.6: Tempo de produção de cada conjunto de *features*.

Com relação aos conjuntos ao lado esquerdo da tabela, é interessante destacar que as *features* 1, 2 e 3, são criadas no mesmo processo, pois são 3 bandas de uma imagem no formato $L^*a^*b^*$, enquanto as *features* 7 e 8, apesar de criadas de forma individual, dependem da criação das *features* 5 e 6 (*Canny*), pois estas são resultados de manipulações com as *features* já criadas pelo detector de bordas *Canny*.

Como forma de simplificar a interpretação dos resultados desta pesquisa, foi feito uma comparação da evolução reportada na Tabela 4.5 por meio de gráficos (Figura 4.5 e Figura 4.6), separados em métricas crescentes, isto é, aquelas que os melhores resultados são representados pelos maiores valores, como é o caso da **RI** e da **DICE**, e também, métricas decrescentes, nas quais os melhores resultados são representados pelos menores valores, tais como **GCE**, **VI** e **BDE**.

Para finalizar a seção de Resultados Quantitativos, na Tabela 4.7, podemos visualizar uma comparação de desempenho entre o método original das Coordenadas de Laplace e alguns métodos clássicos da literatura, bem com ainda a versão aprimorada do referido método conforme estabelecido com o uso de novas *features* introduzidas neste trabalho, referenciado aqui como (LC+NF).

Nas Tabelas 4.7 e 4.8 estão listados os seguintes métodos interativos de segmentação de imagens: *Randon Walker* (RW) (Grady [33]), e suas variantes *Normalized Random Walker* (NRW) (Bampis e Maragos [6]), *Normalized Lazy Random Walker* (NLRW) (Bampis e colaboradores [7]), uma representação do método *Shortest Path* intitulada *Image Foresting Transform* (IFT) na sua versão que utiliza precisão de Sub-pixel (IFTS) (Malmberg e colaboradores [43]), *Power Watersheds* (PWS) (Couprie e colaboradores [21]), bem como uma variante do *Graph Cut* conhecida por *One Cut* (ONE) (Tang e colaboradores [58]).

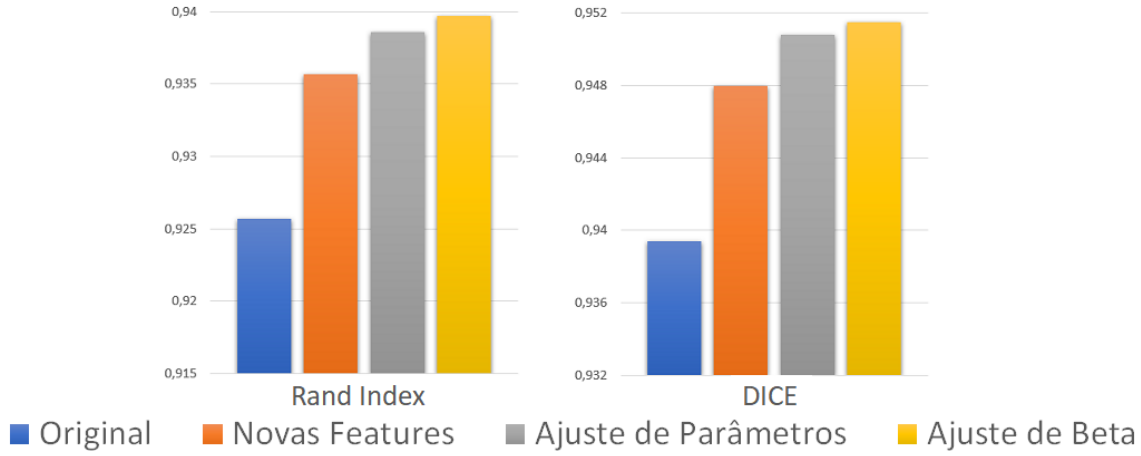



Figura 4.5: Resultados quantitativos para métricas crescentes.

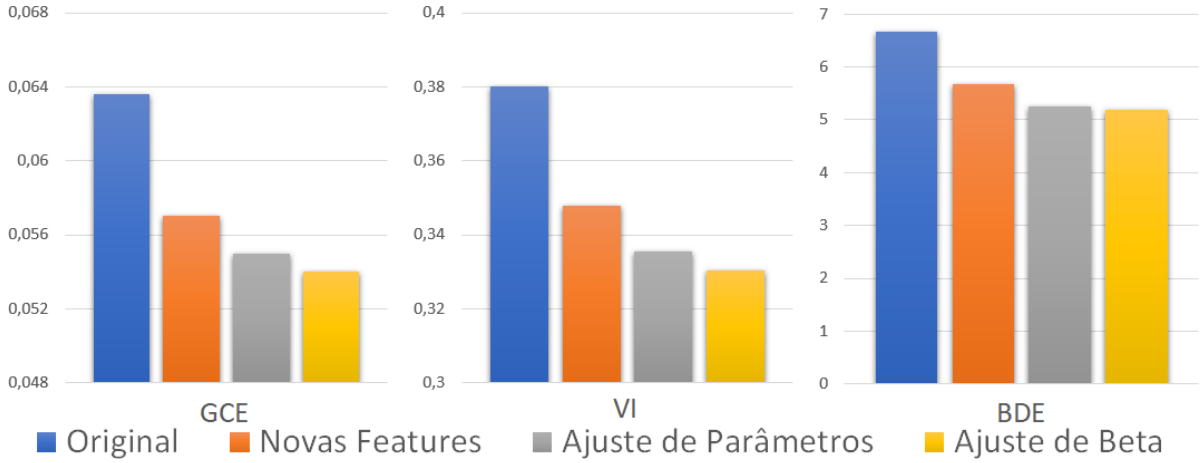



Figura 4.6: Resultados quantitativos para métricas decrescentes.

Os métodos do estado-da-arte foram submetidos a testes com o mesmo conjunto de *seeds* empregado neste trabalho e para 4 métricas de qualidade também descritas neste capítulo, sendo elas: *Rand Index* (RI), *Variation of Information* (VI), *Boundary Displacement Error* (BDE) e *Dice Similarity Coefficient* (DICE).

| **Métricas** | RI (↑) | VI (↓) | BDE (↓) | DICE (↑) |
|---|---|---|---|---|
| IFTS | 0.8817 | 0.5269 | 10.3872 | 0.8980 |
| RW | 0.8980 | 0.4682 | 9.1247 | 0.9138 |
| NRW | 0.9205 | 0.3941 | 6.9122 | 0.9327 |
| NLRW | 0.9208 | 0.3936 | 6.8990 | 0.9329 |
| PWS | 0.9181 | 0.4031 | 7.3470 | 0.9308 |
| ONE | 0.9286 | **0.3126** | 8.3735 | 0.9189 |
| LC+NF | **0.9397** | 0.3305 | **5.1862** | **0.9515** |

Tabela 4.7: Comparação dos segmentadores com o novo conjunto de *features*.

De acordo com os resultados exibidos na Tabela 4.7, é possível ter uma ideia de que a combinação de *features* adotada, juntamente com um estudo do parâmetro de ajuste $\beta$ trouxeram bons resultados quando comparado a métodos clássicos da literatura.

Finalmente, foram realizados testes utilizando as *seeds* originais do *GrabCut* (Seção 4.1.2), isto é, o conjunto de *seeds* densas original da referida base de imagens. A intenção foi também comparar o desempenho da nova configuração de *features* quando combinada ao método de segmentação das Coordenadas de Laplace para segmentar imagens cujos mapas de *seeds* sejam mais densos. Assim como nos testes anteriores, este também foi realizado com as *features* da Composição (*d*), no entanto, foi descartada a ***Feature* 4**. O resultado das métricas computadas para o novo (LC+NF*) pode ser verificado na Tabela 4.8.

| **Métricas** | RI (↑) | VI (↓) | BDE (↓) | DICE (↑) |
|---|---|---|---|---|
| IFTS | 0.9683 | 0.2060 | 3.3637 | 0.9704 |
| RW | 0.9699 | 0.1933 | 3.4522 | 0.9715 |
| NRW | 0.9733 | 0.1793 | **2.8833** | 0.9741 |
| NLRW | 0.9733 | 0.1792 | 2.9225 | 0.9742 |
| PWS | 0.9703 | 0.1931 | 3.1284 | 0.9725 |
| ONE | 0.9651 | 0.2120 | 4.1075 | 0.9645 |
| LC+NF* | **0.9750** | **0.1699** | 3.0676 | **0.9771** |

Tabela 4.8: Comparações dos segmentadores com *seeds* densas (GrabCut *dataset*).

Assim como os resultados do método apresentados na Tabela 4.7, os resultados da Tabela 4.8 também são satisfatórios quando comparados aos métodos clássicos de segmentação interativa, ainda que neste teste a ***Feature* 4**, atrelada ao GMM, não adicione muito aos resultados, as demais bandas foram capazes de gerar bons *scores*, se destacando em 3 de 4 métricas.

### 4.2.2 Resultados Qualitativos

A partir do uso das novas composições de *features*, a segmentação das imagens mesmo com um mapa de *seeds* mais simplificado (esparso) alcançou um bom nível de acurácia com relação aos resultados obtidos pelo método original. A fim de ilustrar essa questão, ambos os resultados podem ser visualizados nas Figuras 4.7 e 4.8, cujas imagens foram amostradas da internet.

(a) Imagem com *Seeds* (b) Resultado Original (c) Novo Resultado

Figura 4.7: Comparação entre o segmentador com o conjunto original de *features* e com a nova composição de bandas, Composição (*d*).

Na Figura 4.9, foram consideradas também outras imagens a fim de comparar visualmente os resultados do modelo com as *features* originais × com as novas bandas investigadas neste trabalho.

(a) *Seeds* (b) Resultado Original (c) Novo Resultado

Figura 4.8: Comparação entre o segmentador com o conjunto original de *features* e com a nova composição de bandas, Composição ($d$).

Já a Figura 4.10 apresenta, além do recorte da segmentação, a máscara obtida pelo método de segmentação das Coordenadas de Laplace quando este assume a configuração de bandas tal como definida pela Composição ($d$). Assim, após ser encontrada a solução do funcional de energia (2.13) através do sistema linear (2.25), que retorna o vetor solução $x$ representando o mapa escalar de valores pertencentes ao intervalo $[0, 1]$, é criada então a máscara que representa a solução da segmentação (mediante ao uso da equação (2.14)). Tal máscara pode ser visualizada na última coluna da Figura 4.10, tal como a imagem, *seeds*, e recorte da segmentação para diferentes imagens modelos. Note que essas imagens não compõem a base *Grabcut*, sendo elas adotadas a fim de ilustrar a capacidade do método na tarefa de segmentação em outras imagens arbitrárias.

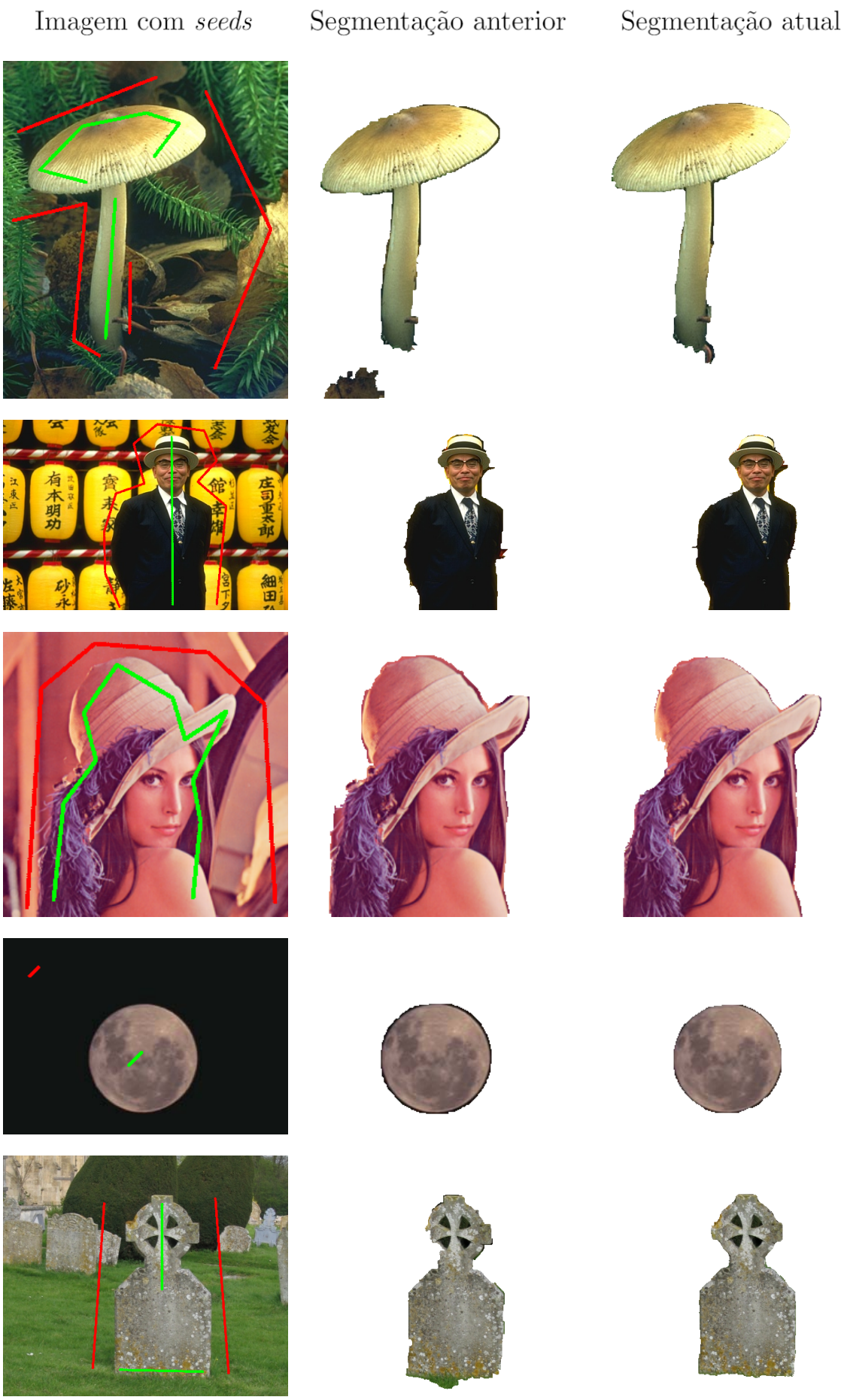



Figura 4.9: Resultados qualitativos: da esquerda para a direita tem-se: imagem de entrada com *seeds*, segmentação original, nova segmentação.

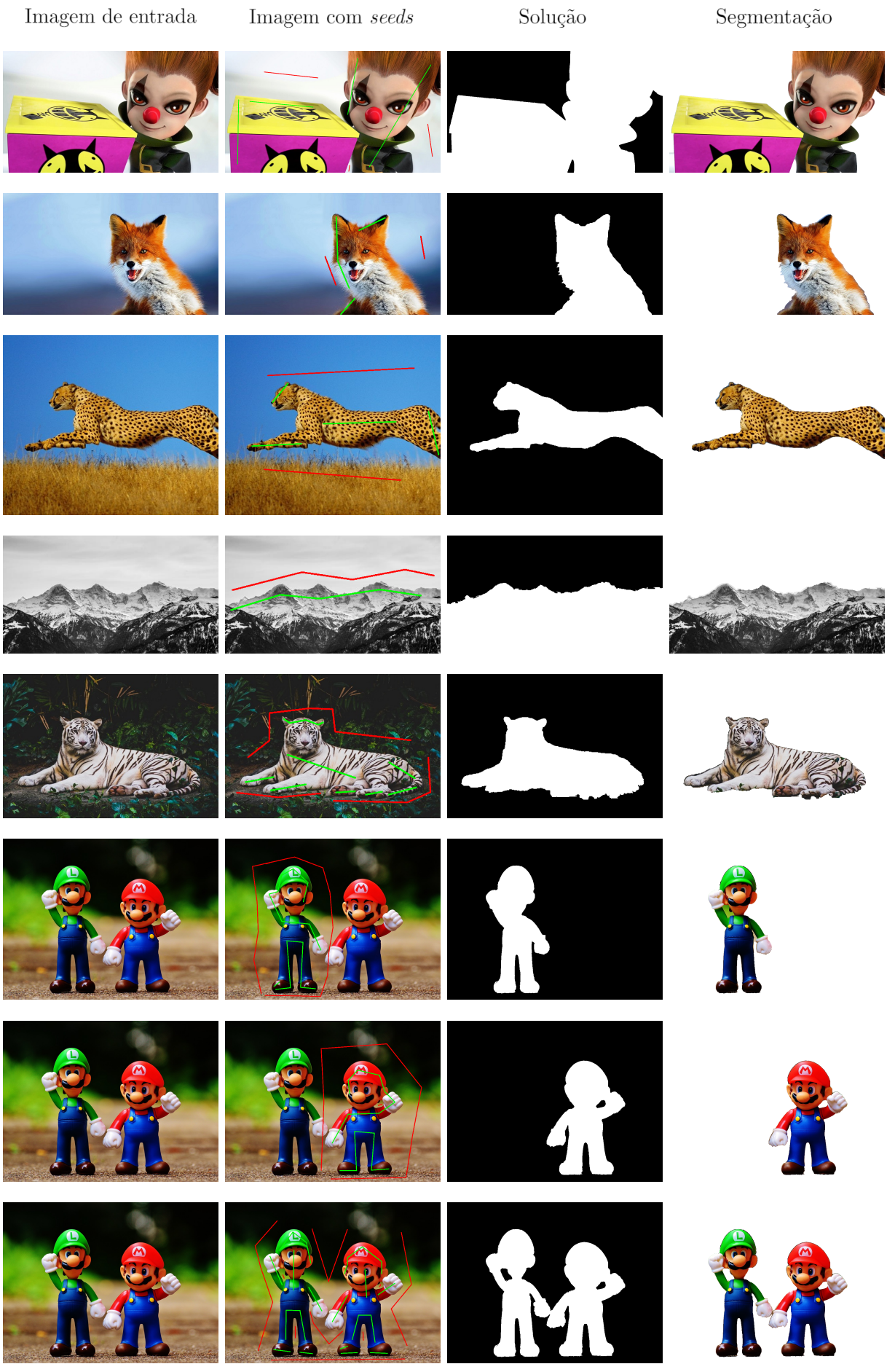



Figura 4.10: Resultados incluindo a solução binária resultante da rotulação como a Eq. (2.14). Da esquerda para direita tem-se: imagem original, imagem com as *seeds*, solução com rótulos $x_F = 1$ (branco) e $x_B = 0$ (preto), resultado da segmentação.

Capítulo

# 5

# Conclusões

A partir da investigação de diferentes métodos de detecção de bordas e sistemas de cores, foi possível concluir que as composições de bandas obtidas por intermédio desses métodos tem muito a contribuir na qualidade da segmentação de uma imagem quando devidamente processadas e introduzidas na matriz $W$ que determina os pesos $w_{ij}$ das arestas do grafo de afinidade do segmentar analisado.

O estudo realizado com foco direcionado às novas configurações de entrada na matriz $W$ quando considerada no modelo de segmentação de imagens das Coordenadas de Laplace trouxe resultados bastante satisfatórios tanto do ponto de vista quantitativo, como também visual. De fato, a ideia de explorar novas técnicas que visam destacar os limites dos objetos em uma imagem, aliada à variação de parâmetros, pode resultar em melhorias ainda mais significativas na tarefa de segmentação.

Dentre os parâmetros explorados, podemos destacar a variável $\beta$ de ajuste dos pesos das arestas do grafo. Tal parâmetro foi estudado após a melhor combinação de *features* ser selecionada e contribuiu ainda mais com o resultado da segmentação. Além disso, foi possível perceber que o valor de $\beta$, que controla a velocidade com que os pesos das arestas tendem a zero, apesar de fixado em um número que apresentou melhores resultados de segmentação, pode ser alterado de acordo com os métodos e/ou composições de *features* que estão sendo utilizados. Visto isso, uma ideia para trabalhos futuros é desenvolvimento de uma ferramenta que interprete um determinado conjunto de imagens, e de forma automática, dentre os parâmetros que foram explorados nesse trabalho, defina os que mais se adéquam a tal conjunto de dados, buscando uma nova melhora na qualidade da segmentação.

Embora a implementação de novos métodos para a melhoria dos resultados acabe aumentando o tempo de processamento da segmentação como um todo, este tempo adicional não torna inviável o uso do recurso como uma ferramenta de segmentação para uso pessoal. Assim, o tempo de processamento para uma imagem específica acaba sendo aceitável, obviamente, a depender das dimensões da mesma.

Ao fim do estudo, foram conduzidos testes com a intenção de analisar o comportamento e o desempenho da atual composição de *features* atrelada ao método das Coordenadas de Laplace, com relação a outro conjunto de *seeds*, bem como ainda a partir de comparações com outros segmentadores do estado-da-arte. Assim, foi possível perceber que o método se comporta bem não só para conjuntos de *seeds* esparsas, mas também apresenta resultados interessantes quando consideradas amostras de *seeds* densas.

# Referências


[1] R. Adams; L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, June 1994.

[2] F. Andrade;E. V. Carrera. Supervised evaluation of seed-based interactive image segmentation algorithms. In *2015 20th Symposium on Signal Processing, Images and Computer Vision (STSIVA)*, pages 1–7, Sep. 2015.

[3] S. Andrews; G. Hamarneh; A. Saad. Fast random walker with priors using precomputation for interactive medical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, pages 9–16, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[4] B. Bai; G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International Journal of Computer Vision*, 82:113–132, 2008.

[5] M. Bai; R. Urtasun. Deep watershed transform for instance segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2858–2866, 2017.

[6] C. G. Bampis; P. Maragos. Unifying the random walker algorithm and the sir model for graph clustering and image segmentation. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 2265–2269, 2015.

[7] C. G. Bampis; P. Maragos; A. C. Bovik. Graph-driven diffusion and random walk schemes for image segmentation. *IEEE Transactions on Image Processing*, 26(1):35–50, 2017.

[8] S. Bao; A. C. S. Chung. Feature sensitive label fusion with random walker for atlas-based image segmentation. *IEEE Transactions on Image Processing*, 26(6):2797–2810, June 2017.

[9] M. Benes; B. Zitova. Performance evaluation of image segmentation algorithms on microscopic image data. *Journal of microscopy*, 257, 09 2014.

[10] G. Bertrand. On topological watersheds. *Journal of Mathematical Imaging and Vision*, 22:217–230, 05 2005.

[11] C Bishop. *Pattern Recognition and Machine Learning*, volume 16. Springer, 2006.

[12] E. Bourennane; P. Gouton; M. Paindavoine; F. Truchetet. Generalization of canny–deriche filter for detection of noisy exponential edge. *Signal Processing*, 82:1317–1328, 2002.

[13] Y. Boykov; M. P. Jolly. Interactive graph cuts for optimal boundary & amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1, 2001.

[14] Y. Boykov; V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, Sep. 2004.

[15] W. Cai; J. Wu; A. C. S. Chung. Shape-based image segmentation using normalized cuts. In *2006 International Conference on Image Processing*, pages 1101–1104, Oct 2006.

[16] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[17] W. Casaca. *Graph Laplacian for spectral clustering and seeded image segmentation*. PhD thesis, Instituto de Ciências Matemáticas e de Computação - ICMC-USP, São Carlos, São Paulo, 2015.

[18] W. Casaca; A. Paiva; E. Gomez-Nieto; P. Joia; L. Nonato. Spectral image segmentation using image decomposition and inner product-based metric. *Journal of Mathematical Imaging and Vision*, 45, 03 2012.

[19] W. Casaca; J. P. Gois; H. C. Batagelo; G. Taubin; L. G. Nonato. Laplacian coordinates: Theory and methods for seeded image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[20] W. Casaca; L. G. Nonato; G. Taubin. Laplacian coordinates for seeded image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 384–391, Washington, DC, USA, 2014. IEEE Computer Society.

[21] C. Couprie; L. Grady; L. Najman; H. Talbot. Power watershed: A unifying graph-based optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1384–1399, July 2011.

[22] J. Cousty; G. Bertrand; L. Najman; M. Couprie. Watershed cuts: Thinnings, shortest path forests, and topological watersheds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:925–939, 2010.

[23] B. Cui; X. Xie; X. Ma; G. Ren; Y. Ma. Superpixel-based extended random walker for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3233–3243, June 2018.

[24] T. A. Davis; W. W. Hager. Dynamic supernodes in sparse cholesky update/downdate and triangular solves. *ACM Trans. Math. Softw.*, 35:27:1–27:23, 2009.

[25] A. P. Dempster; N. M. Laird; D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[26] N. Dias; J. Osowsky; H. R. Gamba; P. Nohama. Controle do cursor do mouse pelo movimento da cabeca usando camera ccd e processamento de imagem. In *IFMBE Proc*, page 442, 2004.

[27] R. Donofrio. Review paper: The helmholtz-kohlrausch effect. *Journal of the Society for Information Display*, 19, 2011.

[28] P. G. Doyle; J. L. Snell. *Random Walks and Electric Networks*. Mathematical Association of America, 1984.

[29] F. França. Indicadores de desempenho e recursos hídricos: proposta de um índice multidimensional para avaliação da implementação de planos de recursos hídricos. Master's thesis, Faculdade de Ciências e Tecnologia (FCT) - Presidente Prudente, 07 2019.

[30] Z. Fu; L. Wang. *Color Image Segmentation Using Gaussian Mixture Model and EM Algorithm*, volume 346, pages 61–66. Springer, 01 2012.

[31] E. Gedraite; M. Hadad. Investigation on the effect of a gaussian blur in image filtering and segmentation, 2011.

[32] R. C. Gonzalez; R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[33] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, November 2006.

[34] L. Grady. Targeted image segmentation using graph methods. In *Image Processing and Analysis with Graphs: Theory and Practice*, Digital Imaging and Computer Vision, pages 111–135. CRC Press, 2012.

[35] L. Grady; A. K. Sinop. Fast approximate random walker segmentation using eigenvector precomputation. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[36] L. Grady; G. Funka-Lea. Multi-label image segmentation for medical applications based on graph-theoretic electrical potentials. In *LNCS*, volume 3117, pages 230–245, 01 2004.

[37] D. Hower; V. Singh; S. C. Johnson. Label set perturbation for mrf based neuroimaging segmentation. *2009 IEEE 12th International Conference on Computer Vision*, pages 849–856, 2009.

[38] A. Iancu; B. Popescu; M. Brezovan; E. Ganea. Quantitative evaluation of color image segmentation algorithms. *IJCSA*, 8:36–53, 2011.

[39] O. Juan; Y. Boykov. Active graph cuts. In *In CVPR*, pages 1023–1029, 2006.

[40] J. Kruskal. On the shortest spanning tree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

[41] A. Y. Lin; A. Novo; S. Har-Noy; N. D. Ricklin; K. Stamatiou. Combining geoeye-1 satellite remote sensing, uav aerial imaging, and geophysical surveys in anomaly detection applied to archaeology. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(4):870–876, Dec 2011.

[42] Y. Liu; F. Cao; J. Zhao; J. Chu. Segmentation of white blood cells image using adaptive location and iteration. *IEEE Journal of Biomedical and Health Informatics*, 21(6):1644–1655, Nov 2017.

[43] F. Malmberg; J. Lindblad, I. Nyström. Sub-pixel segmentation with the image foresting transform. In *Proceedings of the 13th International Workshop on Combinatorial Image Analysis*, pages 201–211. Springer-Verlag, 11 2009.

[44] D. Martin; C. Fowlkes; D. Tal; J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423 vol.2, July 2001.

[45] G. J. McLachlan; D. Peel. *Finite mixture models*. Wiley Series in Probability and Statistics, 2000.

[46] G. J. McLachlan; T. Krishnan. *The EM algorithm and extensions*. Wiley series in probability and statistics. Wiley, 2. ed edition, 2008.

[47] M. Meila. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873 – 895, 2007.

[48] M. Mignotte. A label field fusion model with a variation of information estimator for image segmentation. *Information Fusion*, 20:7–20, 2014.

[49] F. J. C. Monteiro; A. J. C. Campilho. Watershed framework to region-based image segmentation. *2008 19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[50] R. G. Negri; E. A. da Silva; W. Casaca. Inducing contextual classifications with kernel functions into support vector machines. *IEEE Geoscience and Remote Sensing Letters*, 15(6):962–966, June 2018.

[51] T. N. A. Nguyen; J. Cai; J. Zhang; J. Zheng. Robust interactive image segmentation using convex active contours. *IEEE Transactions on Image Processing*, 21(8):3734–3743, Aug 2012.

[52] N. Otsu. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9:62–66, 1979.

[53] R. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

[54] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[55] C. Rother; V. Kolmogorov; A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *ACM TRANS. GRAPH*, pages 309–314, 2004.

[56] A. Smith. Color gamut transform pairs. *ACM Siggraph Computer Graphics*, 12:12–19, 1978.

[57] T. J. Sorensen. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*. Biologiske skrifter. I kommission hos E. Munksgaard, 1948.

[58] M. Tang; L. Gorelick; O. Veksler; Y. Boykov. Grabcut in one cut. In *2013 IEEE International Conference on Computer Vision*, pages 1769–1776, 2013.

[59] R. Unnikrishnan; C. Pantofaru; M. Hebert. A measure for objective evaluation of image segmentation algorithms. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pages 34–34, Sep. 2005.

[60] R. Unnikrishnan; C. Pantofaru; M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):929–944, June 2007.

[61] S. Vicente; V. Kolmogorov; C. Rother. Graph cut based image segmentation with connectivity priors. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[62] Y. Wenjun; Y. Xia; Q. Wang. An improved canny algorithm for edge detection. *Journal of Computational Information Systems*, 75:1516–1523, 2011.

[63] Boykov; G. Funka-Lea. Y. Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision*, 70(2):109–131, nov 2006.

[64] A. Y. Yang; J. Wright; Y. Ma; S. S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110:212–225, 2008.

[65] T. Yuvaraj; N. Krishna; P. Manish; P. Naik; P. Varsha. A review paper on interactive image segmentation. *International Journal of Research and Scientific Innovation (IJRSI)*, 5:285–286, April 2018.