

**Segmentação Não Supervisionada de Imagens
via Transformada Imagem-Floresta Orientada
sujeita a Restrições de Alto Nível**

Hans Harley Ccacyahuillca Bejar

TESE APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
DOUTOR EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof. Dr. Paulo André Vechiatto de Miranda.

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CAPES

São Paulo, 23 de agosto de 2023

Segmentação Não Supervisionada de Imagens via Transformada Imagem-Floresta Orientada sujeita a Restrições de Alto Nível

Esta versão da tese contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 19/07/2023. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Paulo André Vechiatto de Miranda (orientador) - IME-USP
- Prof. Dr. Ronaldo Fumio Hashimoto - IME-USP
- Prof. Dr. Silvio Jamil Ferzoli Guimarães - PUC-MG
- Prof. Dr. Alexandre Xavier Falcão - UNICAMP
- Prof. Dr. Fábio Augusto Menocci Cappabianco - UNIFESP

Agradecimentos

a todos que contribuíram diretamente com o desenvolvimento deste trabalho: o professor Dr. Paulo André Vechiatto de Miranda pela orientação ao longo do doutorado, ao Instituto de Matemática e Estatística (IME) e a Universidade de São Paulo (USP) pela estrutura, recursos computacionais e material bibliográfico.

a minha família pelo apoio nesse tempo de ausência, minha mãe Maria Luisa, minhas irmãs Sara, Jenny, Nery e meus sobrinhos Vania, Yvaine e Gael que são a minha motivação.

aos meus amigos de longa data, meus colegas do IME, da USP e amigos que conheci em São Paulo, obrigado pelo apoio e pelos momentos compartilhados.

aos colegas do laboratório eScience pelas conversas e momentos compartilhados e aos docentes da USP pelo aprendizado.

às agências de fomento CNPq (313554/2018-8, 486988/2013-9, 307062/2016-3, 421521/2016-3, FINEP1266/13), FAPESP (2014/12236-1, 2016/21591-5), FAPEMIG (PPM-06-18), NAP eScience e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Código Financeiro 001.

Resumo

CCACYAHUILLCA-BEJAR, HANS H. **Segmentação Não Supervisionada de Imagens via Transformada Imagem-Floresta Orientada sujeita a Restrições de Alto Nível.** 2023. 146 f. Tese - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2023.

Segmentar uma imagem consiste em particioná-la em regiões que a compõem, tal como para isolar os pixels de um objeto de interesse em uma dada aplicação. A segmentação de imagem é um dos problemas mais desafiadores em processamento de imagens e visão computacional, tendo que lidar com problemas como ruído, volume parcial, inhomogeneidade, variação de iluminação, baixo contraste, variações de resolução da câmera, heterogeneidade do plano de fundo, objetos complexos com diferentes formas e tamanhos. No contexto de segmentação não supervisionada, a dificuldade é agravada pela ausência de pixels rotulados que poderiam nos fornecer pistas para a correta identificação dos objetos de interesse e, portanto, tornando a tarefa menos sujeita a ambiguidades.

A *Oriented Image Foresting Transform* (OIFT) tem sido empregada com sucesso no contexto de segmentação interativa de imagens, permitindo a incorporação de várias restrições de alto nível, tais como polaridade de borda, restrições de conectividade, restrições de forma e relações hierárquicas de inclusão/exclusão, a fim de customizar a segmentação para um dado objeto alvo ou grupo de objetos de interesse. Neste trabalho, estendemos a OIFT para a segmentação não supervisionada de imagens sujeita a restrições de alto nível com base em cortes ótimos em grafos direcionados. O novo método proposto, denominado *Unsupervised OIFT* (UOIFT), considera relações assimétricas para calcular hierarquias de partições. Dentre as restrições de alto nível atualmente suportadas pela UOIFT, temos: (a) Polaridade de borda, que favorece configurações esperadas para as transições de borda do objeto desejado (e.g., transições de claro para escuro/escuro para claro ou entre duas cores esperadas). (b) Penalização de contornos com elevada curvatura, sendo possível distinguir partes côncavas e convexas da forma, assim permitindo a filtragem de formas indesejadas. (c) Favorecimento de objetos com maior área/volume, via reorganização da hierarquia, conservando as demais restrições.

Os resultados são demonstrados utilizando um grafo de regiões adjacentes de *superpixels* em imagens médicas e naturais, exigindo um número menor de partições para isolar com precisão os objetos de interesse nas imagens em comparação com outros métodos da literatura.

Palavras-chave: Segmentação baseada em grafos, segmentação não supervisionada, Transformada Imagem-Floresta Orientada, polaridade de borda, segmentação hierárquica, curvatura, restrições de alto nível.

Abstract

CCACYAHUILLCA-BEJAR, HANS H. **Unsupervised Image Segmentation by Oriented Image Foresting Transform subject to High-Level Constraints**. 2023. 146 p. Thesis - Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2023.

Segmenting an image consists of partitioning it into its composing regions, such as to isolate the pixels of an object of interest in a given application. Image segmentation is one of the most challenging problems in image processing and computer vision, having to deal with problems such as noise, partial volume, inhomogeneity, lighting variation, low contrast, variations of camera resolution, background heterogeneity and complex objects with different shapes and sizes. In the context of unsupervised segmentation, the difficulty is aggravated by the absence of labeled pixels that could provide us with clues for the correct identification of the objects of interest, thus making the task less subject to ambiguity.

The Oriented Image Foresting Transform (OIFT) has been successfully employed in the context of interactive image segmentation, allowing the incorporation of several high-level constraints, such as boundary polarity, connectivity constraints, shape constraints, and hierarchical relations of inclusion/exclusion, in order to customize the segmentation for a given target object or group of objects of interest. In this work, we extend OIFT to perform unsupervised image segmentation subject to high-level constraints based on optimal cuts in directed graphs. The newly proposed method, called Unsupervised OIFT (UOIFT), considers asymmetric relations to calculate hierarchical partitions. Among the high-level constraints currently supported by UOIFT, we have: (a) Boundary polarity, which favors expected settings for the edge transitions of the desired object (e.g., transitions from light to dark/dark to light, or between two expected colors). (b) Penalty for contours with high curvature, making it possible to distinguish concave and convex parts of the shape, thus allowing the filtering of unwanted shapes. (c) Favoring of objects with larger area/volume, via the reorganization of the hierarchy, preserving the other restrictions.

The results are demonstrated using a region adjacency graph of superpixels in medical and natural images, requiring a smaller number of partitions to accurately isolate the objects of interest in the images compared to other methods in the literature.

Keywords: Graph-based segmentation, unsupervised segmentation, Oriented Image Foresting Transform, boundary polarity, hierarchical segmentation, curvature, high-level constraints.

Sumário

Lista de Abreviaturas	xiii
Lista de Símbolos	xvii
Lista de Figuras	xix
Lista de Tabelas	xxiii
1 Introdução	1
1.1 Motivação e Objetivos	4
1.2 Hipóteses de Pesquisa	5
1.3 Revisão bibliográfica: IFTs múltiplas	6
1.4 Contribuições	8
1.5 Organização do trabalho	9
2 Conceitos	11
2.1 Imagem digital	11
2.1.1 A) Imagem em tons de cinza	11
2.1.2 B) Imagem multidimensional	11
2.1.3 C) Imagem multibanda	11
2.2 Digrafos	12
2.3 Grafos a partir de imagens	13
2.4 Caminhos no grafo	14
2.5 Componentes fortemente conexos	15
2.6 Componente conexo direcionado	15
2.7 Segmentação de imagem	16
2.7.1 Partição	16
2.7.2 Superpixel	16
2.7.3 Hierarquia	17
2.7.4 Correspondência entre hierarquias e mapas de saliência	19
2.8 Medidas de avaliação estatística	21
2.8.1 Matriz de confusão	21
2.8.2 Coeficiente de similaridade de <i>Dice</i>	22
2.8.3 Coeficiente de Jaccard	23
2.8.4 Coeficiente Geral Balanceado	24
2.8.5 Avaliação de segmentação interativa por usuário robô	24

3	Métodos Relacionados	27
3.1	Transformada Imagem-Floresta como Algoritmo Base	27
3.1.1	Função de conectividade	28
3.1.2	Função Monotonicamente Incremental	29
3.1.3	Caminho ótimo	29
3.1.4	Floresta de espalhamento como mapa de predecessores	30
3.1.5	Algoritmo geral da IFT	30
3.2	Live Wire (LW)	31
3.3	Transformada Imagem-Floresta Orientada (OIFT)	32
3.4	Métodos de Estimativa de Curvatura em Imagens Digitais	34
3.4.1	Estimativa de curvatura pelo cálculo de diferenças na direção da borda	34
3.4.2	Medida de canto baseada em região	35
3.4.3	Método de detecção de cantos por braços assimétricos	37
3.4.4	Detecção de cantos por áreas de influência	38
3.4.5	Suavização de curva via minimização da energia Elástica	39
3.4.6	k-curvatura	40
4	Segmentação pela Transformada Imagem-Floresta Orientada com Características Locais de Curvatura	43
4.1	Transformada Imagem-Floresta Orientada com Características Locais de Curvatura (OIFT-LCF)	44
4.2	Resultados Experimentais	47
4.3	Considerações parciais	49
5	Segmentação Não Supervisionada de Imagens por Cortes Ótimos em Grafos	51
5.1	Transforma Imagem-Floresta Orientada Não Supervisionada (UOIFT)	54
5.1.1	Definições	55
5.1.2	Próximo objeto de energia máxima com a função de conectividade f_ω	56
5.2	Algoritmo da UOIFT	56
5.3	UOIFT para Imagens Coloridas	57
5.3.1	Explicação do algoritmo UOIFT com imagens sintéticas	59
5.3.2	Relação da UOIFT com uma Árvore de Espalhamento Mínimo (MST)	63
5.4	Tratamento de Empate no Algoritmo UOIFT	63
5.4.1	Problema de Empate	63
5.4.2	Política de Desempate	66
5.5	Manipulação de Hierarquia pelo valor de Extinção de Volume	68
5.6	Resultados Experimentais	75
5.6.1	Análise Qualitativa	75
5.6.2	Análise Quantitativa	80
5.6.3	Experimentos em Imagens Coloridas	82
5.7	Considerações parciais	83

6 Restrições de Curvatura na Transformada Imagem-Floresta Orientada Não Supervisionada	85
6.1 Características Locais de Curvatura entre <i>Superpixels</i>	86
6.2 Algoritmo da UOIFT com Características Locais de Curvatura (UOIFT+LCF) . . .	87
6.3 Algoritmo da UOIFT com Características Locais de Curvatura Côncava (UOIFT+LCCF)	93
7 Conclusões	105
7.1 Considerações finais	105
7.2 Trabalhos futuros	107
Referências Bibliográficas	109
Índice Remissivo	120

Lista de Abreviaturas

COIFT	Transformada Imagem-Floresta Orientada com restrição de Conexidade (<i>Connected Oriented Image Foresting Transform</i>)
DSC	Coefficiente de similaridade de Dice (<i>Dice Similarity Coefficient</i>)
EF+WS	Transformada de Watershed após Filtro de Extinção de volume (<i>Watershed with Extinction Filter</i>)
FC	Conexidade <i>Fuzzy</i> (<i>Fuzzy Connectedness</i>)
FH	Método de segmentação não supervisionado proposto por Felzenszwalb e Huttenlocher (<i>Felzenszwalb and Huttenlocher</i>)
GBC	Coefficiente Geral Balanceado (<i>General Balanced Coefficient</i>)
GC	Corte em grafo (<i>Graph Cut</i>)
GGC	Corte em Grafo Generalizado (<i>Generalized Graph Cut</i>)
GJC	Coefficiente Geral de Jaccard (<i>General Jaccard Coefficient</i>)
GSC	Convexidade Geodésica em Estrela (<i>Geodesic Star Convexity</i>)
HFH	Hierarquização da segmentação de Felzenszwalb e Huttenlocher (<i>Hierarchical segmentation of Felzenszwalb and Huttenlocher</i>)
IFT	Transformada Imagem-Floresta (<i>Image Foresting Transform</i>)
LW	O método de perseguição de contornos (<i>Live Wire</i>)
MI	Função de custo Monotonicamente Incremental (<i>Monotonic-Incremental cost function</i>)
MST	Árvore de Espalhamento Mínimo (<i>Minimum Spanning Tree</i>)
NCut	Corte Normalizado (<i>Normalized Cut</i>)
NMI	Função de custo Não Monotonicamente Incremental (<i>Non-Monotonic-Incremental cost function</i>)
OIFT	Transformada Imagem-Floresta Orientada (<i>Oriented Image Foresting Transform</i>)
OIFT-LCF	Transformada Imagem-Floresta Orientada com Características Locais de Curvatura (<i>Oriented Image Foresting Transform with Localized Curvilinear Features</i>)
ORFC	Conexidade <i>Fuzzy</i> Relativa Orientada (<i>Oriented Relative Fuzzy Connectedness</i>)

RAG	Grafo de Regiões Adjacentes (<i>Region Adjacency Graph</i>)
RAG+LCCF	Grafo de Regiões Adjacentes com Características Locais de Curvatura Côncavas (<i>Region Adjacency Graph with Local Concave Curvature Features</i>)
RFC	Conexidade <i>Fuzzy</i> Relativa (<i>Relative Fuzzy Connectedness</i>)
RGB	Sistema de cores RGB (<i>Red Green Blue</i>)
RM	Ressonância Magnética (MR - <i>Magnetic Resonance</i>)
RM-T1	Ressonância Magnética com a sequência ponderada T1 (<i>T1 weighted magnetic resonance imaging</i>)
RW	Passeios Aleatórios (<i>Random Walks</i>)
SLIC	Agrupamento Iterativo Linear Simples (<i>Simple Linear Iterative Clustering</i>)
UOIFT	Transformada Imagem-Floresta Orientada Não supervisionada (<i>Un-supervised Oriented Image Foresting Transform</i>)
WS	Bacias Hidrográficas (<i>Watershed</i>)

Lista de Símbolos

ρ	Constante
π	Caminho
π_t	Caminho com destino no pixel t
$\pi_{s \rightsquigarrow t}$	Caminho do pixel s ao pixel t
$\pi_{\mathcal{S} \rightsquigarrow t}$	Caminho com origem restrita a um conjunto \mathcal{S} e destino em t
$H(t)$	Valor inicial para caminhos triviais
$\mathcal{C}(\mathcal{O})$	Conjunto de arcos de corte do objeto binário segmentado \mathcal{O}
\mathcal{L}	Contorno dos <i>superpixels</i>
$\phi_G(P)$	Corte da partição P no grafo G
$Curv_\Sigma$	Somatório das magnitudes de curvatura pelo <i>Power Mean</i>
$d(s, t)$	Operador de distância <i>Euclidiana</i>
ε_∞	Funcional de energia de norma máxima (<i>max - norm</i>)
f	Função de conectividade de caminhos
f_{LS}	Função de alongamento linear (<i>Linear Stretch</i>)
f_{mi}	Função monotonicamente incremental
G	Grafo dirigido (dígrafo)
G^T	Grafo transposto
\mathcal{V}	Conjunto de vértices (nós)
\mathcal{A}	Conjunto de arcos ou arestas
ω	Peso de arco
δ	Peso de arco não direcionado ($\delta(s, t) = \delta(t, s)$)
ω^T	Peso de arco do grafo transposto
G_\leq	Grafo residual
$\ \nabla I\ $	Gradiente de uma imagem
\mathcal{H}	Hierarquia
\hat{I}	Imagem digital
\vec{I}	Mapeamento vetorial da imagem digital
\mathcal{I}	Domínio da imagem
$I(t)$	Intensidade do pixel t
\mathcal{X}	Espaço de todos os objetos binários segmentados
$V_{opt}(t)$	Valor ótimo de um caminho até o pixel t

$\Phi_G(\mathcal{H})$	Mapa de saliência da hierarquia \mathcal{H}
\mathcal{O}	Objeto binário segmentado
Π	Conjunto de caminhos
$[P]_x$	Região da partição P que contém o elemento x
P	Partição
Pr	Mapa de predecessores
Q	Fila de prioridade
$R(t)$	Nó raiz
$a \overset{\kappa}{\rightsquigarrow} b$	Relação binária entre vértices a e b em um grafo não direcionado, tal que existe um caminho $\pi_{a \rightsquigarrow b}$ contendo arestas com pesos abaixo de um limiar κ
\mathcal{S}_b	Conjunto de sementes de fundo da imagem (<i>b - background</i>)
\mathcal{S}_o	Conjunto de sementes de objeto da imagem (<i>o - object</i>)
$G(a)$	Magnitude do gradiente
\hat{x}	Classificação ideal

Lista de Figuras

1.1	Ganhos proporcionados pela OIFT, mediante o uso da polaridade de borda	2
1.2	Exemplo de execução da IFT-SLIC	7
2.1	Imagem digital	12
2.2	Exemplos de digrafos	13
2.3	Adjacências Euclidianas	14
2.4	Exemplo dos componentes fortemente conexos de um grafo.	15
2.5	Exemplos de Componentes Conexos Direcionados (DCC) para um dado digrafo . . .	16
2.6	Exemplo da representação de <i>superpixels</i> de uma imagem	17
2.7	Exemplo do grafo de regiões adjacentes de uma imagem sintética.	17
2.8	Exemplo da representação de uma hierarquia baseada em região.	18
2.9	Exemplo da representação de uma hierarquia de zona quase plana.	19
2.10	Hierarquia conexa $\mathcal{H} = (P_0, \dots, P_\ell)$ com $\ell = 3$ em um grafo de vizinhança-4.	20
2.11	Cortes $\phi_G(P_i)$ para cada uma das partições P_i para $0 \leq i \leq \ell$	20
2.12	Representação do mapa de saliência $\Phi_G(\mathcal{H})$	20
2.13	Exemplo de segmentação de uma imagem em 2 classes	22
2.14	Coefficiente de Dice representando a sobreposição espacial e acurácia	23
2.15	Coefficiente de Dice segundo a matriz de confusão	23
2.16	Avaliação por usuário robô	25
3.1	Exemplo do cálculo de custo de um caminho para diferentes funções de conectividade em uma imagem 2D em tons de cinza	29
3.2	Floresta de espalhamento	30
3.3	Exemplo de segmentação de imagem sintética com polaridade de borda via OIFT . .	33
3.4	Padrões de variação de brilho ou de cor para diferentes localizações da janela	35
3.5	Exemplo de interferência por proximidade de múltiplas bordas	37
3.6	Exemplos de áreas de influência interna e externa para a caracterização de cantos . .	39
3.7	Exemplo de estimativa de curvatura	40
3.8	Medida de k -curvatura via variação de k -inclinação ao longo de um trecho do contorno. 41	
3.9	Medida de k -curvatura via norma de vetor resultante em diferentes situações.	41
4.1	Segmentação pela IFT via o método de <i>Live Markers</i>	44
4.2	Os pixels vizinhos à esquerda e à direita de cada arco $\langle s, t \rangle$, considerando um grafo de vizinhança-8.	45
4.3	Perseguição de borda com polaridade de borda em um grafo de vizinhança-4	45

4.4	Extração de informação de contorno em regiões circulares no domínio da imagem . . .	46
4.5	A segmentação da superfície externa do cérebro em uma imagem RM-T1	47
4.6	A segmentação de um osso do pé e do fígado	48
4.7	As quatro diferentes orientações consideradas para o cálculo do gradiente de brilho .	49
4.8	As curvas de acurácia média (Dice) usando diferentes gradientes com e sem o uso das característica de curvatura local	50
5.1	Exemplo da ideia central da UOIFT	53
5.2	Exemplo da UOIFT em uma imagem sintética com inomogeneidade	54
5.3	Exemplo do cálculo de d_s , d_t e d	58
5.4	Exemplos da primeira partição do Algoritmo da UOIFT para diferentes valores de α	60
5.5	Exemplo do mapa de conexidade com a função f_ω	61
5.6	Exemplo da hierarquia $\mathcal{H} = (P_0, P_1, P_2, P_3, P_4)$ computada pela UOIFT sobre o grafo da Figura 5.5a.	62
5.7	Exemplo de dois mapas de conexidade distintos que podem ser obtidos a partir de um mesmo grafo G	64
5.8	Exemplo das possíveis divisões binárias sucessivas resultantes do primeiro caso do mapa de conexidade da Figura 5.7	65
5.9	Exemplo das divisões binárias sucessivas do segundo caso do mapa de conexidade da Figura 5.7.	66
5.10	Exemplo de tratamento de empate	67
5.11	Tratamento de empate na Etapa 2 do algoritmo da UOIFT	68
5.12	Ilustração de uma hierarquia $\mathcal{H} = (P_0, P_1, P_2, \dots, P_9)$	70
5.13	Representações alternativas a partir da hierarquia \mathcal{H}	70
5.14	Processo de reorganização hierárquica pelo cálculo dos valores de extinção do atributo de volume na <i>Min-Tree</i>	72
5.15	Exemplos de reorganização das hierarquias geradas pela UOIFT pelo filtro de extinção de volume para realce dos padrões Aruco	74
5.16	Exemplo com imagem médica 3D	76
5.17	Exemplo da segmentação não supervisionada de uma fatia de um estudo torácico por tomografia computadorizada para segmentar o fígado com diferentes métodos e valores de k	77
5.18	Exemplo da segmentação não supervisionada das letras e números de uma placa de carro para diferentes métodos e valores de k	78
5.19	Exemplo da segmentação não supervisionada de uma imagem real de ressonância magnética do pé para diferentes métodos e valores de k	79
5.20	Exemplos de imagens para os experimentos quantitativos	80
5.21	As curvas de acurácia média, pelo coeficiente de similaridade de Dice e o coeficiente generalizado de Jaccard (BG com $\delta = 1$), da melhor união das regiões produzidas para diferentes métodos e valores de k , para segmentar o osso tálus.	81
5.22	As curvas de acurácia média, pelo coeficiente de similaridade de Dice e o coeficiente generalizado de Jaccard (BG com $\delta = 1$), da melhor união das regiões produzidas para diferentes métodos e valores de k , para segmentar a vértebra espinhal.	81

5.23	O efeito de reduzir o tamanho dos <i>superpixels</i> para 5×5 pixels. A curva de acurácia média, pelo coeficiente de similaridade de Dice, da melhor união das regiões produzidas para diferentes métodos e valores de k , para segmentar o osso tálus. . . .	82
5.24	Experimentos qualitativos para imagens coloridas	82
5.25	As curvas de acurácia média, pelo coeficiente de similaridade de Dice e o coeficiente geral de Jaccard (BG com $\delta = 1$), da melhor união das regiões produzidas para diferentes métodos e valores de k , para segmentar um texto específico em uma imagem colorida sujeita a diferentes condições de iluminação.	83
6.1	Exemplo de penalização da curvatura na segmentação não supervisionada	85
6.2	Sequência de passos para calcular a segmentação automática com penalização de curvatura.	86
6.3	Ilustração de curvatura na fronteira de <i>superpixels</i>	86
6.4	Ilustração da função de aumento linear de contraste (<i>Linear Stretch</i>) f_{LS}	89
6.5	Exemplo comparativo entre a UOIFT e UOIFT+LCF, a partir da imagem sintética da Figura 6.1a	91
6.6	Experimentos comparativos entre a UOIFT e UOIFT+LCF em imagens reais	92
6.7	Exemplo de dois casos de curvatura na borda entre dois <i>superpixels</i> vizinhos X e Y .	93
6.8	Exemplo de objetos com diferentes curvaturas	96
6.9	Grafo de Regiões Adjacentes (RAG) sem penalização de curvatura	97
6.10	Grafo de Regiões Adjacentes com penalização de curvatura (RAG+LCF)	98
6.11	Grafo de Regiões Adjacentes com penalização apenas de curvaturas côncavas (RAG+LCCF)	99
6.12	Exemplo comparativo entre a UOIFT, UOIFT+LCF e UOIFT+LCCF, a partir da imagem sintética da Figura 6.8a	100
6.13	Comparação dos pesos dos arcos $\langle p, q \rangle$ nos grafos de regiões adjacentes (RAG, RAG+LCF e RAG+LCCF)	101
6.14	Experimento comparativo entre a UOIFT, UOIFT+LCF e UOIFT+LCCF em uma imagem real	102
6.15	Experimento comparativo entre a UOIFT, UOIFT+LCF e UOIFT+LCCF em uma imagem real	103
7.1	Página web dos materiais suplementares da publicação [Bejar <i>et al.</i> (2020)].	107
7.2	Marcadores fiduciais (<i>landmarks/fiducial markers</i>) com molduras em múltiplas cores.	108

Lista de Tabelas

1.1	Número de IFTs necessárias para implementar diferentes métodos da literatura. . . .	9
2.1	Matriz de confusão gerada a partir de um conjunto de instâncias.	21
2.2	Matriz de confusão do exemplo da Figura 2.13.	22

Capítulo 1

Introdução

A segmentação de imagens consiste em dividir uma imagem em regiões que a compõem, como, por exemplo, para isolar os pixels de um objeto alvo de uma dada aplicação. A segmentação é um problema muito desafiador em visão computacional e processamento de imagens, tendo que lidar com adversidades, tais como ruído, volume parcial, inomogeneidade, variação de iluminação, baixo contraste, variações de resolução da câmera, heterogeneidade do plano de fundo, objetos complexos com diferentes formas e tamanhos [Madabhushi e Udupa (2005); Suetens (2009)]. No contexto de segmentação não supervisionada, a dificuldade é agravada pela ausência de pixels rotulados que poderiam nos fornecer pistas para a correta identificação dos objetos de interesse, tornando assim a tarefa menos sujeita a ambiguidades. A segmentação é útil em várias aplicações, tais como na análise de imagens médicas ou biológicas, detecção de objetos ou de atividade humana, rastreamento e na composição de fotos digitais em publicidade [Lézoray e Grady (2012)].

O uso de grafos é adequado para representar qualquer dado discreto, como imagens digitais, modelando relações de vizinhança. No contexto de segmentação supervisionada, a segmentação interativa pode ser formulada como sendo um problema de partição de um grafo derivado da imagem sujeito à restrições fortes, dadas por pixels sementes selecionados no domínio da imagem [Boykov e Funka-Lea (2006); Ciesielski *et al.* (2007); Cousty *et al.* (2010)], tal como ilustrado na Figura 1.1. A integração de técnicas de aprendizado de máquina com métodos construídos sobre o forte formalismo de partições em grafos tornou-se um tópico de pesquisa muito relevante, como atesta um dos métodos pioneiros proposto em [Farabet *et al.* (2013)] que alcançou resultados de ponta em várias bases de dados em sua época, usando uma combinação de Aprendizado Profundo (DL - *Deep Learning*)¹ e abordagens hierárquicas por técnicas baseadas em Bacias Hidrográficas (WS - *Watershed*). Exemplos desta integração de sucesso têm se popularizado até mesmo em tarefas de mais baixo nível, tal como na geração de superpixels [Belém *et al.* (2019, 2022); Belém *et al.* (2021)]. Existem diferentes maneiras de integrar mapas de probabilidade por Redes Neurais Convolucionais (CNN - *Convolutional Neural Network*) com arcabouços de otimização em grafos, como no aprendizado dos pesos do grafo e na definição de sementes ou outras restrições [Oliveira *et al.* (2021); Wolf *et al.* (2017)]. Nesse sentido, métodos de partição em grafos podem ser utilizados no topo de um pipeline de segmentação, combinando com técnicas de *deep learning*, garantindo o estabelecimento teórico das propriedades formais dos objetos gerados.

Uma classe importante dos métodos interativos de segmentação de imagem baseados em grafos compreende os métodos por região baseados em sementes, incluindo diferentes arcabouços, tais como Bacias Hidrográficas (WS - *Watershed*) [Cousty *et al.* (2010, 2008); Lotufo *et al.* (2002)], Passeios Aleatórios (RW - *Random Walks*) [Grady (2006)], Conexidade Fuzzy (FC - *Fuzzy Con-*

¹O desenvolvimento de redes neurais profundas constitui um avanço disruptivo em visão computacional [Farabet *et al.* (2012)] tornando efetiva a resolução de muitos problemas, como detecção de objetos [Zhao *et al.* (2019)], localização [Oquab *et al.* (2015)] e classificação [Wang *et al.* (2016)]. Por outro lado a segmentação de imagens continua sendo um problema difícil de resolver, partindo por exemplo do fato de que em um cenário genérico de segmentação é impossível obter o objeto de interesse com elevada qualidade de segmentação (e.g., para gerar um gabarito de segmentação), sem o uso de alguma restrição predefinida pelo usuário, por se tratar de um problema mal definido, com muitas ambiguidades para se identificar qual o objeto de interesse para um usuário específico.

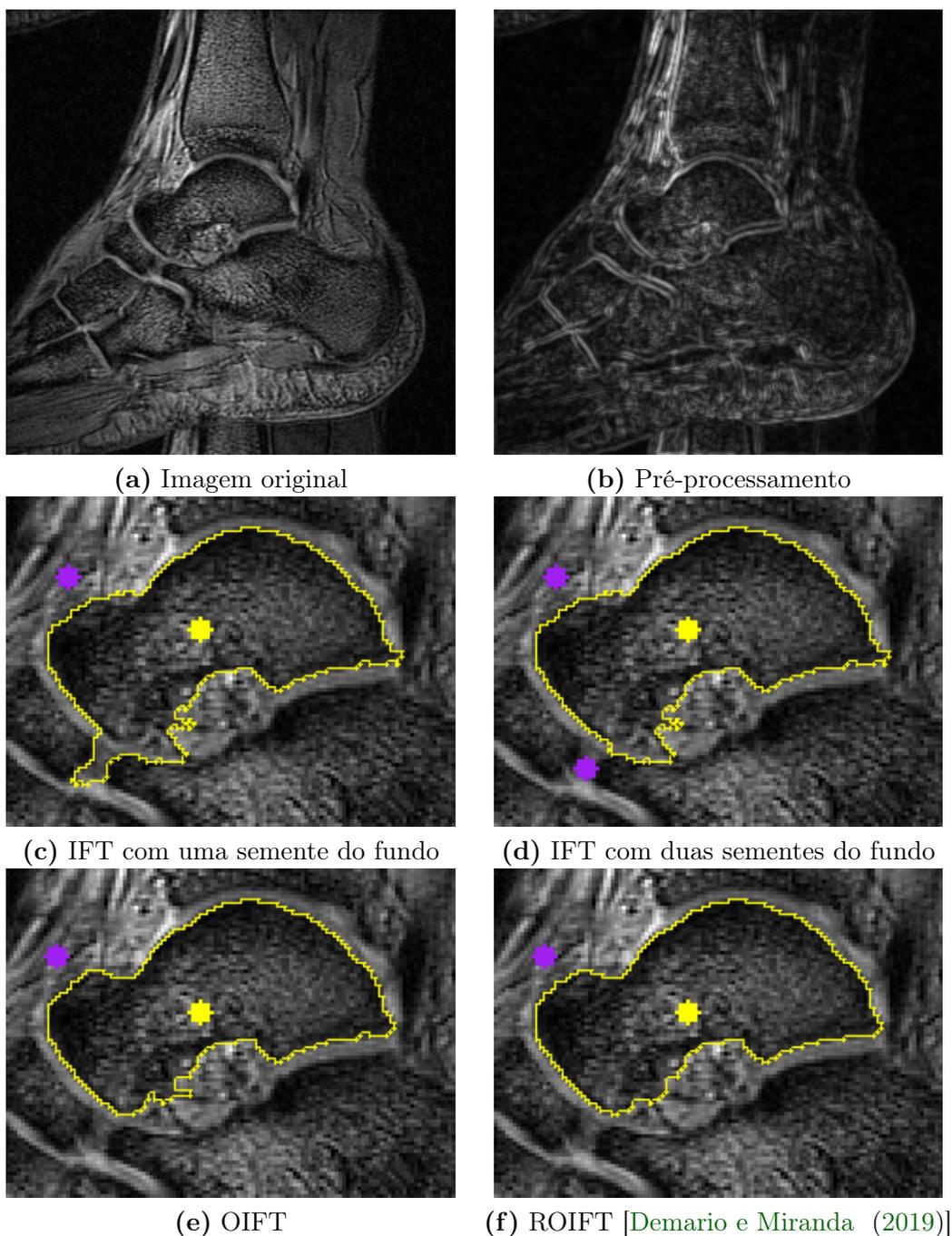


Figura 1.1: Ganhos proporcionados pela OIFT, mediante o uso da polaridade de borda: (a) Imagem de ressonância magnética do pé humano. (b) Pré-processamento para a construção de um grafo derivado da imagem com pesos obtidos com base no gradiente da imagem. (c-f) Segmentação interativa via o arcabouço da IFT do osso tálus com sementes de objeto e fundo, utilizando diferentes restrições. (c) Resultado por IFT Watershed em grafo não direcionado apresenta erros de falso positivo na parte inferior esquerda. (d) Resultado atualizado da IFT Watershed após adição de uma semente extra de fundo. (e) Um resultado superior é obtido pela OIFT favorecendo a polaridade de borda de pixels escuros para claros nas transições de borda, utilizando uma única semente de fundo. (f) Resultado final após pós-processamento de relaxação da OIFT (ROIFT [Demario e Miranda (2019)]) para suavizar o contorno resultante.

nectedness) [Audigier e Lotufo (2007); Ciesielski *et al.* (2007); Saha e Udupa (2001)], Corte em Grafo (GC - Graph Cut) [Boykov e Funka-Lea (2006)] e Transformada Imagem-Floresta (IFT - Image Foresting Transform) [Falcão *et al.* (2004)]. Várias destas soluções podem ser descritas de

modo unificado dentro de um arcabouço em comum, às vezes referido como Corte em Grafo Generalizado (GGC - *Generalized Graph Cut*) [C.Coupric *et al.* (2010); Ciesielski *et al.* (2012b)], que inclui os problemas de minimização das funções objetivo de soma dos pesos das arestas de corte (energia ε_1) e norma máxima (*max-norm*), correspondendo ao maior peso de aresta do corte (energia ε_∞).

Cada função objetivo de energia tem suas próprias vantagens e desvantagens. Enquanto os métodos da família ε_∞ têm problemas relacionados a fronteiras irregulares e “vazamento”² através de bordas mal definidas, os métodos de ε_1 sofrem de problemas de viés de encolhimento e maior tempo computacional³ [C.Coupric *et al.* (2010)].

Alguns trabalhos propõem o uso de métodos híbridos ou baseados em múltiplas minimizações de energia, utilizando iterações de diferentes classes de energia para abordar distintas partes da imagem [Bejar e Miranda (2015); Ciesielski *et al.* (2013)](ou métodos baseados na fusão iterativa dos resultados de uma única classe de energia aplicada sobre diferentes subgrafos recalculados a cada passo [Peng *et al.* (2011)]), que tendem a aliviar os problemas acima mencionados, mas eles perdem o ótimo global, uma vez que não realizam uma única minimização de energia sobre todos elementos da imagem. Como consequência, é mais difícil incorporar restrições de alto nível para segmentação de objetos nessas abordagens.

Seguindo a histórica definição de adjacência simétrica, a maioria dos métodos em grafos, baseiam-se em grafos não direcionados. Alguns trabalhos mais recentes têm como objetivo estender estes métodos para além da hipótese de simetria, a fim de melhorar os algoritmos populares de segmentação de imagens, utilizando arcos antiparalelos com pesos assimétricos. A Transformada Imagem-Floresta Orientada (OIFT- *Oriented Image Foresting Transform*) [Mansilla e Miranda (2013a); Miranda e Mansilla (2014)] e a Conexidade Fuzzy Relativa Orientada (ORFC - *Oriented Relative Fuzzy Connectedness*) [Bejar e Miranda (2015)] são extensões para grafos ponderados direcionados de alguns métodos do *framework* de Corte em Grafo Generalizado (GGC - *Generalized Graph Cut*) [Ciesielski *et al.* (2012b)], incluindo Conexidade Fuzzy (*Fuzzy Connectedness*) [Ciesielski *et al.* (2007)] e Watersheds [Cousty *et al.* (2010, 2008)]. A OIFT gera um corte ótimo no grafo direcionado de acordo com uma medida de corte em grafo de norma máxima, tendo um custo computacional menor em comparação com o algoritmo do fluxo máximo-corte mínimo (*min-cut/max-flow*) [Boykov e Funka-Lea (2006)], dado que sua complexidade computacional é a mesma de uma IFT regular.

A formulação de energia da OIFT em dígrafos, a torna um método muito versátil, suportando várias restrições de alto nível para segmentação de objetos, incluindo propriedades globais dos objetos, tais como conexidade do objeto segmentado [Mansilla e Miranda (2016); Mansilla *et al.* (2016)], restrições de forma [Braz *et al.* (2022); de Moraes Braz e Miranda (2014); de Moraes Braz *et al.* (2020); Mansilla e Miranda (2013b)], polaridade de borda [Mansilla e Miranda (2013b); Miranda e Mansilla (2014)], tamanho máximo permitido [Condori e Miranda (2022)] e restrições hierárquicas [Leon e Miranda (2017); Leon e de Miranda (2019); Leon *et al.* (2021b)], que permitem a personalização da segmentação para um determinado objeto alvo ou grupo de objetos de interesse [Lézoray e Grady (2012)]. Os ganhos proporcionados pela OIFT, mediante o uso da polaridade de borda, são exemplificados nas Figuras 1.1c-e, quando comparados aos resultados de uma IFT Watershed em grafo não direcionado. Observe que para segmentar o osso tálus, a IFT requer sementes adicionais de fundo (Figuras 1.1c-d), ao passo que um resultado superior pode ser obtido pela OIFT com esforço de usuário reduzido (Figura 1.1e). No entanto, por ser uma extensão do problema da minimização de ε_∞ para grafos direcionados, a OIFT herda todas as desvantagens dessa classe de energia, incluindo a geração de bordas irregulares. Portanto, é recomendado o uso de

²O termo “vazamento” é comumente usado para designar erros de falso positivo em regiões de baixo contraste na borda do objeto. O termo se popularizou no contexto da segmentação por *Watershed* no paradigma de imersão, na qual a propagação de rótulos reflete a elevação do nível das águas.

³Os métodos de minimização ε_∞ têm complexidade $O(N \log N)$ com relação ao tamanho da imagem N , com implementações em tempo linear $O(N)$ podendo ser alcançadas em alguns casos, dependendo da estrutura de dados da fila de prioridade [Ciesielski *et al.* (2012a)], enquanto que o tempo de execução do problema de minimização ε_1 é $O(N^{2.5})$ para grafos esparsos [Boykov e Funka-Lea (2006)].

um pós-processamento de relaxação para suavizar o delineamento resultante [Demario e Miranda (2019)] (Figura 1.1f).

OIFT também pertence a uma classe de algoritmos da *General Fuzzy Connectedness* (GFC) descrita em Ciesielski *et al.* (2016) e, portanto, possui várias boas propriedades teóricas, como robustez para a colocação de sementes, tal como provado em de Moraes Braz *et al.* (2020). De fato, verificou-se experimentalmente que a OIFT tem um bom equilíbrio entre acurácia e robustez quando comparada a outros métodos semelhantes em grafos [Tavares *et al.* (2017b)]. A OIFT tem também alguma relação com a extensão do algoritmo da *max-tree* para grafos direcionados de Perret *et al.* (2015). No entanto, a abordagem de segmentação descrita em Perret *et al.* (2015) está mais fortemente relacionada ao método ORFC [Bejar e Miranda (2015)]. A OIFT tem sido utilizada com sucesso na prática, por exemplo, para segmentar centenas de imagens tridimensionais para criar um atlas anatômico para regularizar o problema de impedância elétrica de tomografia [Choi *et al.* (2023)].

Em vista de seus resultados positivos obtidos, de sua versatilidade e eficiência, nesta tese visamos estender a OIFT, a fim de ampliar seu leque de aplicações, conforme será discutido nas próximas seções.

1.1 Motivação e Objetivos

A transformada imagem-floresta (IFT - Image Foresting Transform) reduz problemas de processamento de imagem baseados em conectividade ao cálculo:

- de uma floresta de caminhos ótimos no grafo derivado da imagem,
- seguido de um pós-processamento simples de atributos da floresta resultante.

Assim, a IFT tem como principais motivações os conceitos de simplicidade e unificação.

Simplicidade: Os operadores de imagem são reduzidos a escolha de poucos parâmetros no algoritmo da IFT e um processamento local de sua saída.

Unificação: Vários operadores de imagem são derivados de um algoritmo geral, nos favorecendo compreender suas relações e suas possíveis extensões.

Em trabalhos mais recentes, a ideia da unificação foi estendida para incorporar operadores de imagem baseados em múltiplas execuções da IFT. Como vantagens de se reduzir problemas de processamento de imagens ao cálculo de uma sequência de IFTs, temos os ganhos de eficiência, podendo tirar proveito de:

- implementações em hardware [Cappabianco *et al.* (2007)],
- cálculo diferencial [Condori *et al.* (2020); Falcão e Bergo (2004)] e
- paralelo [Bergo e Falcão (2007)].

Nesta pesquisa, adotamos a estratégia da unificação por meio da redução de problemas a múltiplas execuções da IFT, a partir da identificação dos seguintes pontos de pesquisa ainda em aberto do arcabouço:

- a ausência de estudos sobre a incorporação de restrições de curvatura dos contornos produzidos no paradigma de segmentação por região via OIFT e
- falta de estudos sobre como estender a segmentação pela OIFT para o contexto de segmentação não supervisionada.

Restrições de curvatura na IFT foram alvo de estudo apenas no paradigma de segmentação por perseguição de bordas pelos trabalhos [Condori *et al.* (2017); Kang (2005)], que estendem o método do *Live Wire* [Falcão *et al.* (2000, 1998)]. Nossos estudos de curvatura no paradigma por região indicam que o uso de grafos direcionados, como feito pela OIFT, nos permitem distinguir partes côncavas e convexas da forma, assim permitindo a filtragem de formas indesejadas. Com relação ao segundo ponto levantado, a extensão pretendida da OIFT traria como benefícios a incorporação de restrições de alto nível na segmentação não supervisionada, mediante a escolha de poucos parâmetros para a especificação das propriedades esperadas do objeto alvo de interesse, ao invés do tedioso processo da construção de uma base segmentada de treinamento, típico de abordagens de aprendizagem de máquina. Mais especificamente, temos como objetivo incorporar restrições de alto nível na geração de hierarquias.

Métodos de análise de imagens e vídeos que utilizam representações hierárquicas de imagens visam explorar a representação visual como um espaço-escala orientado por regiões, isto é, um conjunto de representações baseadas, por exemplo, em grafos, com diferentes níveis de detalhe, nos quais as representações em níveis mais finos são aninhadas para obter níveis mais grosseiros produzindo assim uma hierarquia de partições [Cousty *et al.* (2018)]. A estrutura apresentada em nosso trabalho lida com hierarquias conexas e, portanto, um grafo precisa ser especificado para definir a conectividade das regiões das partições nas hierarquias [Cousty *et al.* (2018)]. A conectividade das regiões é a principal diferença entre segmentação e classificação hierárquica. Apesar das várias abordagens existentes para computar hierarquias de partições, desenvolver métodos eficientes e efetivos não é uma tarefa fácil, devido às informações semânticas necessárias para realizar a segmentação. Nesta tese visamos avançar no estado da arte em hierarquia de partições, considerando relações assimétricas para calcular hierarquias de partições.

Nossos objetivos de se encontrar algoritmos eficientes, escaláveis (e.g., para tratar uma grande quantidade de dados que não podem ser armazenados em memória primária), extensíveis a imagens multidimensionais e versáteis para a geração de hierarquias em grafos se alinham aos objetivos do projeto CNPq (407242/2021-0)⁴.

1.2 Hipóteses de Pesquisa

Dado o problema de se obter uma segmentação com baixa curvatura ao longo do seu contorno de modo computacionalmente eficiente no paradigma de segmentação por região, adotamos a seguinte hipótese de pesquisa.

Hipótese 1. *Se múltiplas execuções da IFT são empregadas, é possível obter uma segmentação baseada em região com contornos suaves a um baixo custo computacional.*

Dado o problema de se obter uma segmentação não supervisionada sujeita a restrições de alto nível de modo computacionalmente eficiente, temos a seguinte hipótese de pesquisa.

Hipótese 2. *Se múltiplas execuções da IFT são empregadas, é possível estender a OIFT para o paradigma não supervisionado com cortes ótimos, via minimização da energia ε_∞ em grafos direcionados, sujeitos a restrição da polaridade de borda dos objetos.*

A viabilidade da Hipótese 1 é baseada na premissa de que o problema pode ser tratado em duas etapas. Em um primeiro momento, múltiplas execuções locais da IFT são empregadas para a extração de características curvilíneas por toda a imagem, de modo a antever problemas futuros na etapa de rotulação. Então, em uma segunda etapa, a segmentação final por região via OIFT é realizada penalizando segmentos de borda com acentuada curvatura. As múltiplas IFTs da primeira etapa podem ser paralelizadas, além de outras otimizações que serão discutidas, tornando viável o baixo custo computacional.

⁴Projeto intitulado "Processamento e análise de imagens e vídeos baseada em grafos (PAIG)" do Edital CNPq UNIVERSAL 18/2021 (Conselho Nacional de Desenvolvimento Científico e Tecnológico), coordenado pelo Prof. Silvio Jamil F. Guimarães (PUC Minas).

Já a Hipótese 2, que corresponde a parte central da tese, é baseada na premissa de que a OIFT não supervisionada pode ser obtida, mediante divisões binárias sucessivas do domínio da imagem, conservando a consistência da polaridade de borda das partições, via minimização da energia ε_∞ em grafos direcionados, através de múltiplas execuções da IFT. Na sequência, mostramos como os métodos resultantes das Hipóteses 1 e 2 podem ser integrados em um único método, consideravelmente mais poderoso, que provê a segmentação não supervisionada com restrições de curvatura, diferenciando formas com partes côncavas e convexas, além da restrição de polaridade de borda e com favorecimento de objetos com maior área/volume.

Dado que múltiplas execuções da IFT são empregada em nossas hipóteses de pesquisa, a seguir apresentamos uma revisão bibliográfica dos métodos existentes na literatura que são baseados no emprego de IFTs múltiplas, a fim de evidenciar o sucesso de tais abordagens, bem como para facilitar a verificação da originalidade dos métodos que serão propostos nesta tese.

1.3 Revisão bibliográfica: IFTs múltiplas

Métodos de segmentação pela Transformada Imagem-Floresta (IFT - *Image Foresting Transform*) [Falcão *et al.* (2004)], em geral, são compostos por um pré-processamento (Figura 1.1b), seguido por uma rotulação dos pixels (Figura 1.1c) e opcionalmente um pós-processamento [Demario e Miranda (2019)]. A IFT pode ser empregada em qualquer uma das etapas mencionadas. Em segmentação interativa de imagens, múltiplas execuções da IFT podem ser necessárias na etapa de rotulação, a medida que o usuário adiciona ou remove marcadores, porém em geral apenas uma única execução da IFT tem sido utilizada em cada interação do usuário, podendo a atualização ser calculada de modo diferencial para um melhor desempenho de tempo [Falcão e Bergo (2004)]. Na etapa de pré-processamento, uma única execução da IFT é normalmente empregada tal como para implementar um filtro conexo [Falcão *et al.* (2001)] (por exemplo, reconstrução superior ou inferior da imagem) e na etapa de pós-processamento podemos ter o uso de um esqueleto multiescala via IFT [Falcão *et al.* (2002)], que requer uma execução da IFT.

Por outro lado, a implementação de algumas restrições na OIFT requerem múltiplas execuções da IFT a cada interação do usuário. Por exemplo, o algoritmo da OIFT com restrição de forma por Convexidade Geodésica em Estrela (OIFT-GSC - *Oriented Image Foresting Transform with Geodesic Star Convexity*) [Mansilla e Miranda (2013b)] requer duas execuções do algoritmo convencional da IFT, enquanto que a OIFT com restrição de Conexidade (COIFT - *Connected Oriented Image Foresting Transform*) [Mansilla e Miranda (2016); Mansilla *et al.* (2016)] requer pelo menos três execuções do algoritmo da IFT. No entanto, métodos baseados em múltiplas execuções do algoritmo da IFT ainda são pouco explorados na literatura.

No contexto de segmentação não supervisionada de imagens, os únicos trabalhos baseados em múltiplas execuções da IFT correspondem aos métodos de geração de superpixels, baseados no método IFT-SLIC (*Image Foresting Transform - Simple Linear Iterative Clustering*) [Alexandre *et al.* (2015)]. IFT-SLIC calcula os custos dos caminhos entre pixels e os centros dos agrupamentos, ao invés de suas distâncias diretas tal como empregado no SLIC [Achanta *et al.* (2010)], para gerar os superpixels como regiões de pixels fortemente conexas, tal que cada agrupamento corresponde a uma árvore na floresta de espalhamento calculada pela IFT. Inicialmente, os centros começam como uma grade regularmente espaçada de sementes (Figura 1.2a). Um procedimento de atualização dos pixels sementes é então aplicado a cada iteração, gerando superpixels conexos com melhor aderência às bordas dos objetos presentes na imagem (Figuras 1.2b-d). O processo é então repetido por um total de dez iterações, o que requer dez execuções da IFT (Figuras 1.2e-f). Ganhos de desempenho podem ser obtidos via o cálculo diferencial de IFTs, pelo método DIFT [Condori *et al.* (2020)]. Como exemplos de trabalhos derivados da IFT-SLIC, temos sua versão generalizada dada pelo arcabouço da ISF (*Iterative Spanning Forest*) [Vargas-Muñoz *et al.* (2019)], RISF (*Recursive Iterative Spanning Forest*) [Lemes Galvão *et al.* (2018)], DISF (*Dynamic Iterative Spanning Forest*) [Belém *et al.* (2020)] e SICLE (*Superpixels through Iterative CLearcutting*) [Belém *et al.*

(2022)].

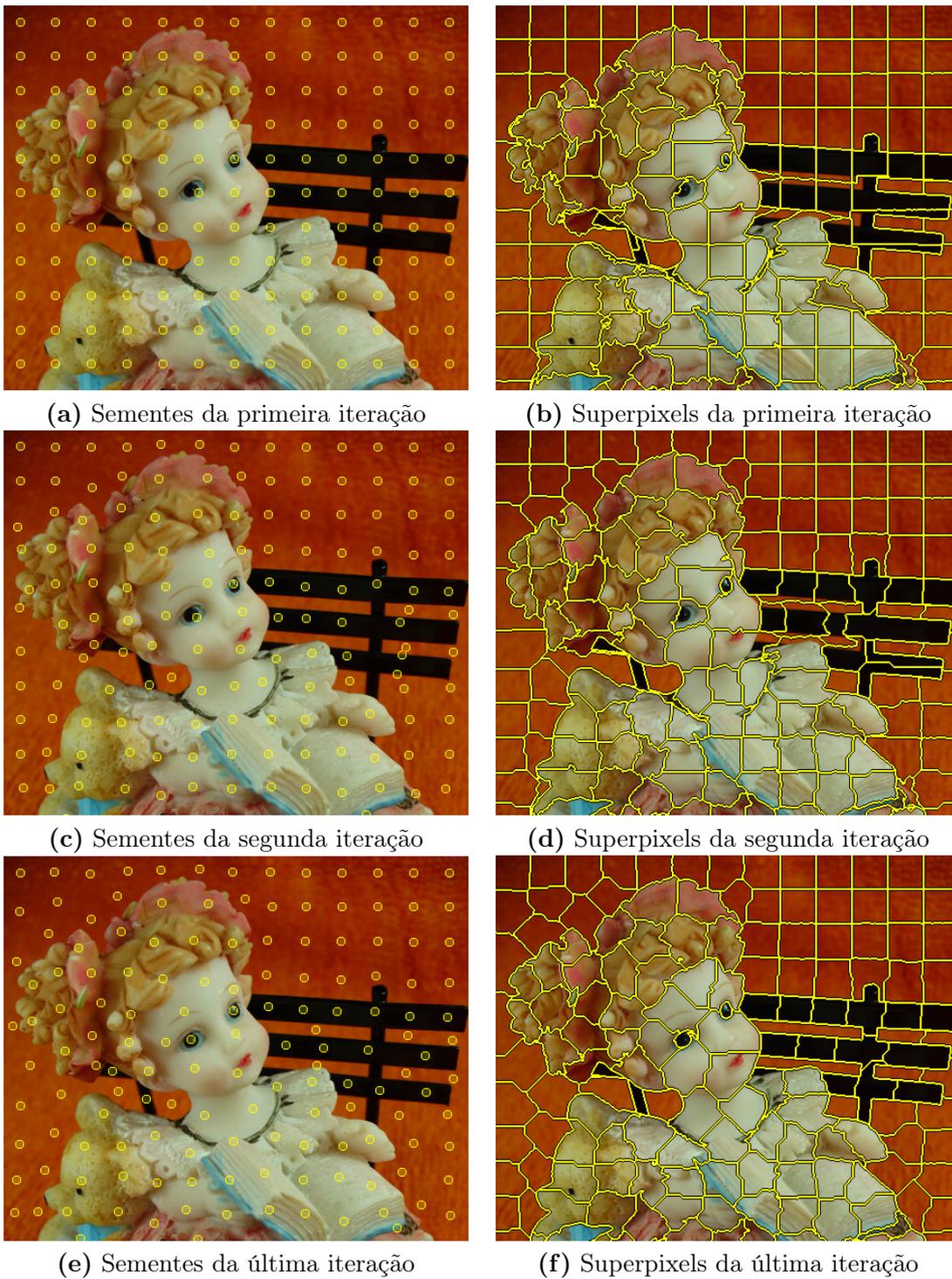


Figura 1.2: Exemplo de execução da IFT-SLIC. (a) Sementes iniciais. (b) Resultado da primeira IFT. (c) Sementes atualizadas da segunda iteração. (d) Resultado da segunda IFT. (e) Configuração final de sementes da décima iteração. (f) Resultado da décima IFT.

RISF calcula uma segmentação hierárquica, chamando ISF múltiplas vezes para um número decrescente de regiões [Lemes Galvão *et al.* (2018)]. As ISFs são computadas em grafos de adjacência de regiões (RAGs - *Region Adjacency Graphs*), criando um novo RAG a partir das regiões resultantes para cada execução subsequente, até atingir o número de regiões desejado para a escala final. No

caso da RISF com amostragem em grade (*grid sampling*), temos um total de execuções da IFT dado por $K \times 10$, em que K é o número de escalas. No caso da RISF com amostragem de sementes geodésicas (*geodesic seed sampling*), temos um total de execuções da IFT dado por $(K \times 10 + \sum_{k=1}^K n_k)$, sendo n_k o número de regiões na k -ésima escala e $n_k > n_{k+1}$, $k = 1, 2, \dots, K - 1$. Observe no entanto que o método é computacionalmente eficiente, pois as IFTs para escalas subsequentes são calculadas em grafos cada vez menores não impactando, portanto, no tempo computacional.

DISF [Belém *et al.* (2020)] e SICLE [Belém *et al.* (2022)] são métodos para calcular superpixels baseados na estratégia de sobreamostragem de semente, para melhorar a probabilidade de selecionar sementes relevantes que promovam o delineamento preciso do objeto, seguida pela redução do conjunto de sementes (removendo as sementes menos importantes de acordo com uma medida de relevância) e recomputação da IFT, ao longo de algumas iterações, até obtermos o número desejado de partições de imagem. DISF não possui um parâmetro explícito para controlar o número de iterações. Na prática, o número médio de execuções da IFT exigidas pelo DISF é de cerca de 5 a 7 cálculos. SICLE, por outro lado, permite indicar explicitamente o número de iterações desejado, com uma recomendação padrão de considerar 5 iterações. Posteriormente, foi proposta uma versão melhorada do SICLE, denominada SICLE $^\alpha$ [Belém *et al.* (2022)], que requer apenas duas iterações para $\alpha = 1$, tirando proveito de mapas de saliência de objetos fornecidos.

Alguns métodos clássicos de segmentação também podem ser reduzidos a uma sequência de execuções de IFTs, como é o caso da Conexidade Fuzzy Relativa (RFC - *Relative Fuzzy Connectedness*) [Saha e Udupa (2001)], em que são calculadas as forças de conexidade dos melhores caminhos a partir das sementes de objeto e fundo, separadamente, por meio de duas execuções da IFT [Ciesielski *et al.* (2012a)]. Já em outros casos, uma aproximação pode ser obtida via IFT, tal como no caso da Distância Mínima de Barreira (MBD - *Minimum Barrier Distance*) [Strand *et al.* (2013)]. Em Strand *et al.* (2013), foi considerado um algoritmo que encontra uma aproximação da MBD, calculando duas IFTs separadamente, uma para cada termo da função de custo da MBD.

Métodos baseados em múltiplas execuções da IFT têm sido também alvo de investigações para filtragem de imagem com preservação de bordas, tal como no caso da PTF (*Power Tree Filter*) [Danda *et al.* (2017)], utilizando a IFT de modo proporcional ao número de pixels da imagem N .

Mais recentemente, foi também proposta uma nova restrição de área máxima do objeto de interesse no arcabouço da OIFT, necessitando de múltiplas execuções da OIFT para sua implementação, que podem ser calculadas mais rapidamente mediante o emprego da OIFT diferencial [Condori e Miranda (2022)]. O número de OIFTs necessárias é dado pela quantidade de sementes adicionais de fundo computadas automaticamente ao longo das iterações (isto é, a cardinalidade do conjunto \mathcal{S}'_b , com $|\mathcal{S}'_b| \ll N$).

Na Tabela 1.1 é apresentado um resumo dos métodos de segmentação existentes baseados em múltiplas execuções da IFT, com a indicação das suas respectivas quantidades de IFTs usadas.

1.4 Contribuições

Como resultados, temos primeiro um novo método de segmentação que utiliza múltiplas execuções da IFT na etapa de pré-processamento, via múltiplas execuções locais de perseguição de bordas por *Live Wire* (LW) [Falcão *et al.* (2000)], combinadas com a rotulação por IFT orientada (OIFT), devido às suas forças complementares. Os resultados demonstram que bordas mais suaves e regulares, com baixa curvatura, podem ser favorecidas pelo método proposto, atenuando um dos principais problemas da OIFT original, a um baixo custo computacional em comparação com outros métodos em grafos [Bejar *et al.* (2017)], assim respondendo de modo positivo a primeira hipótese levantada. A solução obtida pode ser facilmente adaptada para um determinado objeto alvo, de acordo com suas características curvilíneas locais e demais restrições já suportadas pela OIFT.

Como segundo resultado, temos um novo método (UOIFT - *Unsupervised OIFT*), que estende a OIFT para a segmentação não supervisionada de imagens, com base em cortes ótimos em grafos

Métodos	# IFTs
RFC [Saha e Udupa (2001)]	$2 \times IFT$
MBD [Strand <i>et al.</i> (2013)]	$2 \times IFT$
OIFT-GSC [Mansilla e Miranda (2013b)]	$2 \times IFT$
COIFT [Mansilla e Miranda (2016); Mansilla <i>et al.</i> (2016)]	$3 \times IFT$
SICLE [Belém <i>et al.</i> (2022)]	$5 \times IFT$
DISF [Belém <i>et al.</i> (2020)]	$7 \times IFT$
IFT-SLIC [Alexandre <i>et al.</i> (2015)]	$10 \times IFT$
ISF [Vargas-Muñoz <i>et al.</i> (2019)]	$10 \times IFT$
RISF [Lemes Galvão <i>et al.</i> (2018)]	$K \times 10 \times IFT$
OIFT com restrição de área [Condori e Miranda (2022)]	$ \mathcal{S}'_b \times IFT$
PTF [Danda <i>et al.</i> (2017)]	$N \times IFT$

Tabela 1.1: Número de IFTs necessárias para implementar diferentes métodos da literatura.

dirigidos, e que pode ser adaptado para diferentes objetos, de acordo com suas polaridades de borda. O método proposto UOIFT engloba, como caso particular, a versão *top-down* do algoritmo de *single-linkage* por árvore geradora mínima (MST - *Minimum Spanning Tree*), estabelecendo contribuições teóricas importantes, e fornece resultados superiores de segmentação em comparação com outras abordagens comumente utilizadas na literatura, geralmente exigindo um número menor de partições da imagem para isolar as regiões de interesse desejadas. O método é suportado por novos resultados teóricos envolvendo o uso de funções de custo Não-Monotonicamente Incrementais em grafos direcionados. Finalmente, incorporamos características locais de curvatura (LCF - *Localized Curvilinear Features*) no método UOIFT para combiná-la com a restrição de polaridade de borda, de modo a penalizar objetos com contornos irregulares. Estes últimos resultados atendem a segunda hipótese levantada positivamente.

Em resumo, temos as seguintes contribuições, ordenadas em ordem cronológica de publicação:

1. Dois artigos publicados sobre cálculo da região de robustez dos métodos OIFT e ORFC [Bejar e Miranda (2015)], em continuidade aos trabalhos do mestrado, no SPIE Medical Imaging 2017 [Tavares *et al.* (2017a)] e no International Symposium on Mathematical Morphology (ISMM17) [Tavares *et al.* (2017b)].
2. Um novo método baseado em múltiplas execuções da IFT, chamado OIFT-LCF (Oriented Image Foresting Transform with Localized Curvilinear Features), publicado no International Conference on Image Analysis and Processing (ICIAP17), Catania, Itália [Bejar *et al.* (2017)].
3. Um novo método de segmentação não supervisionado, chamado UOIFT (Unsupervised OIFT), baseado em cortes ótimos em grafos dirigidos, apresentado no 23rd Iberoamerican Congress on Pattern Recognition (CIARP 2018), Madrid, Espanha [Bejar *et al.* (2018)].
4. Uma publicação na revista *Pattern Recognition Letter* do método UOIFT com tratamento de empates e testes em imagens coloridas e volumes 3D [Bejar *et al.* (2020)].

1.5 Organização do trabalho

Por uma questão de completude na apresentação desta tese, detalhamos os conceitos necessários sobre grafos e imagens na Seção 2. Apresentamos a IFT, o método de perseguição de bordas *Live-Wire*, a OIFT e métodos de estimativa de curvatura na Seção 3.

Na Seção 4 investigamos a extração de características locais de curvatura, via múltiplas execuções locais do método *Live Wire*, combinadas com a rotulação por OIFT, chamado OIFT-LCF (*Oriented Image Foresting Transform with Localized Curvilinear Features*) [Bejar *et al.* (2017)], os

resultados demonstraram que bordas mais suaves e regulares, com baixa curvatura, podem ser favorecidas pelo método proposto, a um baixo custo computacional em comparação com outros métodos em grafos.

Na Seção 5, apresentamos o novo método da UOIFT (*Unsupervised Oriented Image Foresting Transform*) para a segmentação não supervisionada de imagens, com base em cortes ótimos em grafos dirigidos, que pode ser adaptado para diferentes objetos de acordo com suas polaridades de borda, e na Seção 6, as restrições baseadas nas características locais de curvatura são incorporadas para penalizar os arcos no grafo de regiões adjacentes no algoritmo da UOIFT.

Finalmente, na Seção 7 apresentamos a conclusão e trabalhos futuros.

Capítulo 2

Conceitos

Neste capítulo são apresentados os conceitos básicos e notações que serão adotados ao longo deste documento.

2.1 Imagem digital

Uma imagem digital \hat{I} é um par (\mathcal{I}, \vec{I}) tal que \mathcal{I} é um conjunto de pontos do \mathbb{Z}^n (domínio da imagem) denominados pixels¹ e $\vec{I}(t)$ é um mapeamento vetorial que associa a cada pixel $t \in \mathcal{I}$ um vetor $\vec{I}(t) = \{I_1(t), I_2(t), \dots, I_m(t)\} \in \mathbb{Z}^m$ de valores escalares, associados a um conjunto de m propriedades físicas. O valor de n refere-se à dimensão da imagem e o valor de m ao número de bandas. Segundo os parâmetros de uma imagem digital, temos outras definições:

2.1.1 A) Imagem em tons de cinza

Uma imagem $\hat{I} = (\mathcal{I}, I)$ em tons de cinza (por exemplo, foto preto e branco, imagem de ultrassom, fatia tomográfica) e bidimensional ($\mathcal{I} \subset \mathbb{Z}^2$) possui apenas uma banda I ($m = 1$). A imagem em tons de cinza bidimensional é portanto uma matriz de tamanho $N \times M$ pixels (N linhas e M colunas). Se $I(t) \in \{0, 1\}$ para todo $t \in \mathcal{I}$, \hat{I} é dita **binária**.

2.1.2 B) Imagem multidimensional

Uma imagem $\hat{I} = (\mathcal{I}, I)$ em tons de cinza e multidimensional define o domínio de amostragem $\mathcal{I} \subset \mathbb{Z}^n$, para $n > 2$. Por exemplo, uma sequência espacial de fatias tomográficas é uma imagem tridimensional ($n = 3$) e uma sequência espacial e temporal de fatias tomográficas é uma imagem tetradimensional ($n = 4$). No primeiro caso, os elementos de \mathcal{I} são chamados de *voxels* (*elemento de volume*) e a imagem é chamada de volume.

2.1.3 C) Imagem multibanda

Uma imagem $\hat{I} = (\mathcal{I}, \vec{I})$ é multibanda quando o mapeamento vetorial \vec{I} associa a cada pixel t obrigatoriamente mais de um valor escalar ($\vec{I}(t) = \{I_1(t), I_2(t), \dots, I_m(t)\}$, para $m > 1$). Imagens de satélite, por exemplo, possuem bandas que correspondem a imagens cinza obtidas nos comprimentos de onda do azul, vermelho, verde, infravermelho, infravermelho próximo, termal e ultravioleta. O intervalo de amostragem define a resolução espectral. No caso de uma foto colorida temos $m = 3$, com $I_1(t)$, $I_2(t)$ e $I_3(t)$ correspondendo aos componentes vermelho, verde e azul (Figura 2.1). Além disso, o vídeo colorido é uma imagem multidimensional e multibanda.

¹pixel: É o elemento espacial em uma imagem 2-D. Para uma imagem 3-D cada elemento é chamado de *voxel*. De forma mais geral cada elemento é conhecido como *spel - spatial element*. Para o melhor entendimento do trabalho as definições estão baseadas em imagens 2-D.

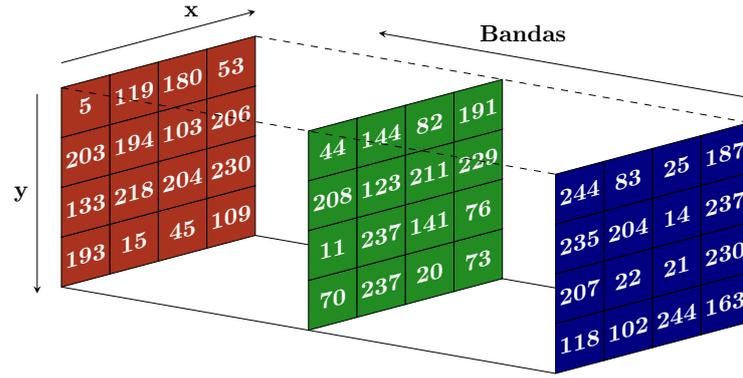


Figura 2.1: Imagem digital de tamanho 4×4 bidimensional ($n = 2$), com três bandas ($m = 3$) RGB.

2.2 Digrafos

Um grafo dirigido ou digrafo $G = \langle \mathcal{V}, \mathcal{A} \rangle$ é uma tupla de conjuntos disjuntos, tal que $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ e $\mathcal{V} \neq \emptyset$. O conjunto \mathcal{V} é o conjunto de vértices e \mathcal{A} é o conjunto de arcos. Para dois vértices de G , s e t , existe no máximo um arco $\langle s, t \rangle$, de s para t . Arcos com direções opostas são chamados de **arcos anti-paralelos**, como por exemplo os arcos $\langle s, t \rangle$ e $\langle t, s \rangle$ na Figura 2.2a.

- Um digrafo G é **simétrico** se $\forall s, t \in \mathcal{V}, s \neq t, \langle s, t \rangle \in \mathcal{A}$ se e somente se $\langle t, s \rangle \in \mathcal{A}$ (Figura 2.2a). Caso contrário, o digrafo é não simétrico (Figura 2.2b).
- Um digrafo G cujos arcos $\langle s, t \rangle$ possuem valores reais associados $\omega(\langle s, t \rangle)$ (nomeados pesos) dados por uma função $\omega : \mathcal{A} \rightarrow \mathbb{R}$, é denominado digrafo **ponderado**, sendo denotado por uma terna ordenada $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$ (Figura 2.2c).
- A **transposta** $G^T = \langle \mathcal{V}, \mathcal{A}^T, \omega^T \rangle$ de um digrafo ponderado $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$ é o único digrafo ponderado sobre o mesmo conjunto de vértices \mathcal{V} com todos os arcos invertidos em comparação com a orientação dos arcos correspondentes em G , isto é, para qualquer um dos seus arcos $\langle s, t \rangle \in \mathcal{A}^T$, o par $\langle t, s \rangle$ é um arco de G , e $\omega^T(\langle s, t \rangle) = \omega(\langle t, s \rangle)$, como nas Figuras 2.2c-d.

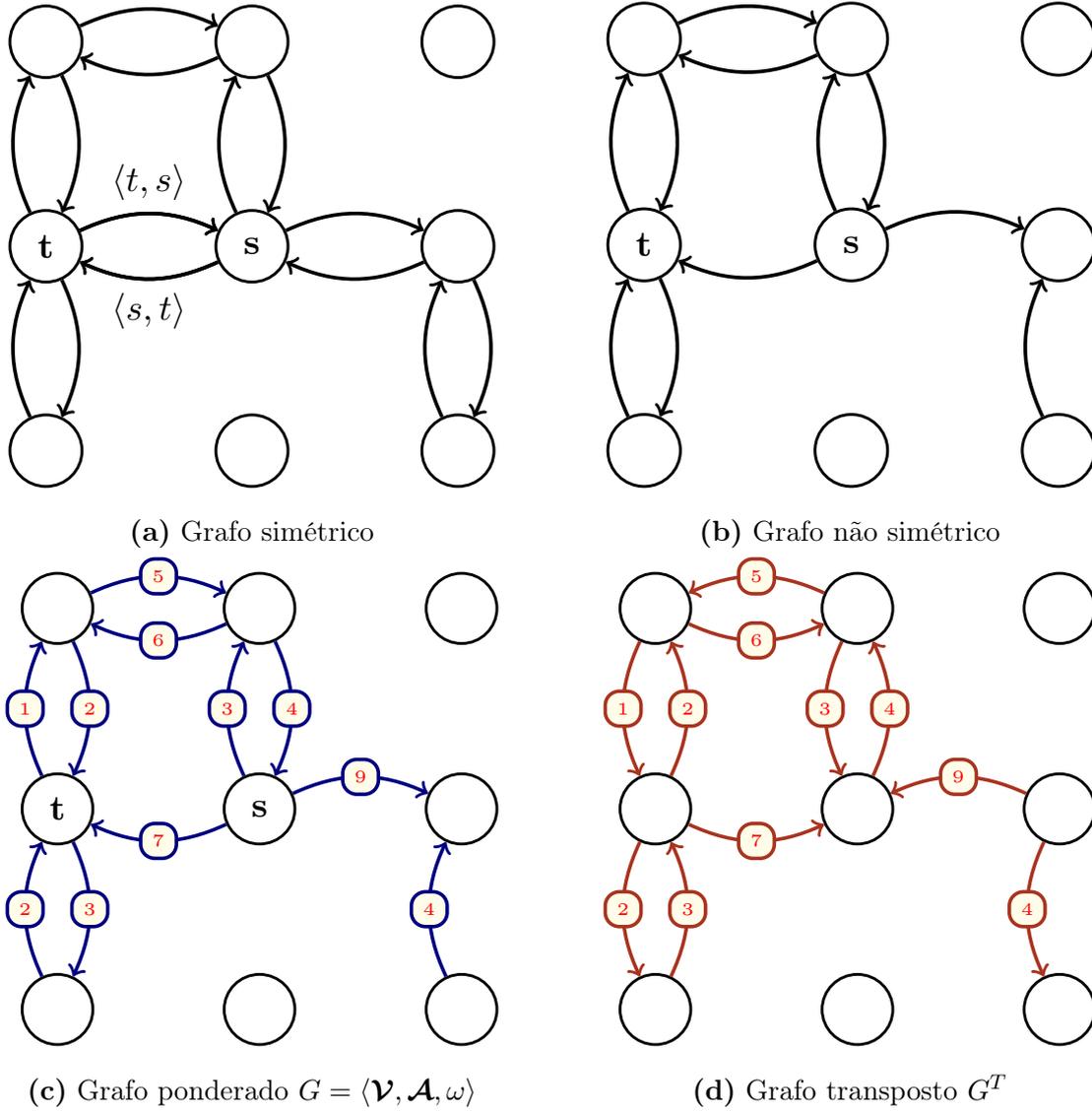


Figura 2.2: Exemplos de digrafos: (a) Grafo simétrico. (b) Exemplo de um grafo não simétrico. (c) Exemplo de um grafo ponderado não simétrico, com os pesos indicados sobre os arcos. Por exemplo, para o arco $\langle s, t \rangle$ temos $\omega(\langle s, t \rangle) = 7$. (d) Grafo transposto do grafo G em (c).

2.3 Grafos a partir de imagens

Uma imagem (ou volume 3D) pode ser interpretada como um digrafo ponderado $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, cujos nós \mathcal{V} são os pixels (voxels) no domínio da imagem $\mathcal{I} \subset \mathbb{Z}^n$ e cujos arcos são os pares ordenados de pixels $\langle s, t \rangle \in \mathcal{A}$. Por exemplo, pode-se tomar \mathcal{A} consistindo de todos os pares de pixels $\langle s, t \rangle$ no produto Cartesiano $\mathcal{I} \times \mathcal{I}$ tal que $d(s, t) \leq \rho$ e $s \neq t$, sendo que $d(s, t)$ denota a distância Euclidiana e ρ é uma constante especificada (por exemplo: vizinhança-4, quando $\rho = 1$, e vizinhança-8 quando $\rho = \sqrt{2}$, no caso de imagens 2-D). Essa forma de definir os arcos é conhecida como relação de adjacência **circular** (Figura 2.3).

A função dos pesos dos arcos ω é dada pela *dissimilaridade* entre as características dos pixels vizinhos (ex: magnitude do gradiente de intensidades), ou pela sua noção dual dada por uma função de *afinidade/similaridade*.

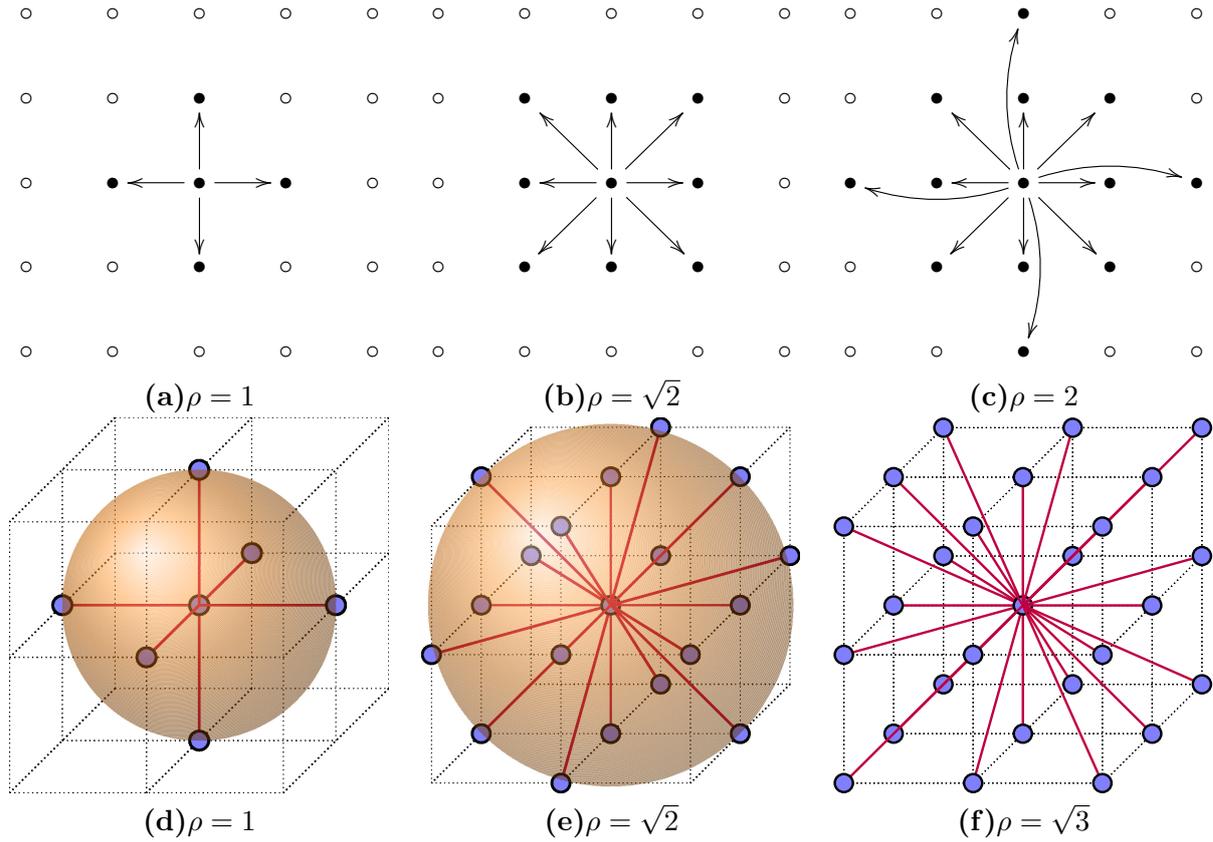


Figura 2.3: *Adjacências Euclidianas.* (a) Imagem 2-D de raio $\rho = 1$ (vizinhança-4), (b) raio $\rho = \sqrt{2}$ (vizinhança-8), (c) raio $\rho = 2$ (vizinhança-12). (d) Imagem 3-D (volume) de raio $\rho = 1$ (vizinhança-6), (e) raio $\rho = \sqrt{2}$ (vizinhança-18), (f) raio $\rho = \sqrt{3}$ (vizinhança-26).

Outros tipos de relação de adjacência são:

- **Retangular:** $\langle s, t \rangle \in \mathcal{A}$ se $|x_t - x_s| \leq \frac{l}{2}$ e $|y_t - y_s| \leq \frac{h}{2}$, tal que l e h são os comprimentos dos lados do retângulo com centro em (x_s, y_s) .
- **Baseado em conjunto:** $\langle s, t \rangle \in \mathcal{A}$ se $t - s \in \{(-1, -1), (1, -1)\}$.
- **Baseada em propriedades locais:** $\langle s, t \rangle \in \mathcal{A}$ se $d(s, t) \leq \rho_i$ e $d(\vec{I}(s), \vec{I}(t)) \leq \rho_j$, onde $d()$ é o operador de distância *Euclidiana* e ρ_i e ρ_j são os raios nos espaços da imagem e de atributos (paramétrico), respectivamente.

2.4 Caminhos no grafo

Para um determinado grafo da imagem $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, um caminho no grafo $\pi = \langle t_1, t_2, \dots, t_n \rangle$ é uma sequência de pixels adjacentes (isto é, $\langle t_i, t_{i+1} \rangle \in \mathcal{A}, i = 1, \dots, n - 1$). Neste trabalho, adotamos as seguintes notações para indicar casos específicos de caminhos:

- π_t : Representa um caminho $\pi_t = \langle t_1, t_2, \dots, t_n = t \rangle$ com um pixel de destino t . Um caminho é *trivial* quando $\pi_t = \langle t \rangle$.
- $\pi_{s \rightsquigarrow t}$: Caminho que indica explicitamente seu pixel de origem. Na notação $\pi_{s \rightsquigarrow t} = \langle t_1 = s, t_2, \dots, t_n = t \rangle$ temos que s representa a origem e t o nó de destino.
- $\pi_{\mathcal{S} \rightsquigarrow t}$: De um modo mais geral, pode-se usar $\pi_{\mathcal{S} \rightsquigarrow t} = \langle t_1, t_2, \dots, t_n = t \rangle$ para indicar um caminho com origem restrita a um conjunto \mathcal{S} (isto é, $t_1 \in \mathcal{S}$).

Um caminho $\pi_t = \pi_s \cdot \langle s, t \rangle$ indica a extensão de um caminho π_s por um arco $\langle s, t \rangle$. Já $\pi_{r \rightsquigarrow t} = \pi_{r \rightsquigarrow s} \cdot \pi_{s \rightsquigarrow t}$ indica a concatenação de dois caminhos.

Usamos a notação $\Pi(G)$ para indicar o conjunto de todos os caminhos possíveis em um digrafo G .

Um digrafo é dito ser **fortemente conexo** se há um caminho de cada vértice para todos os outros vértices.

2.5 Componentes fortemente conexos

Um componente fortemente conexo de um grafo dirigido $G = \langle \mathcal{V}, \mathcal{A} \rangle$ é um subconjunto de vértices $C \subseteq \mathcal{V}$ (Figura 2.4), tal que:

1. Para todo par de vértices s e t em C , existe um caminho $\pi_{s \rightsquigarrow t}$ de s a t e um caminho $\pi_{t \rightsquigarrow s}$ de t a s .
2. C é maximal com respeito à propriedade (1).

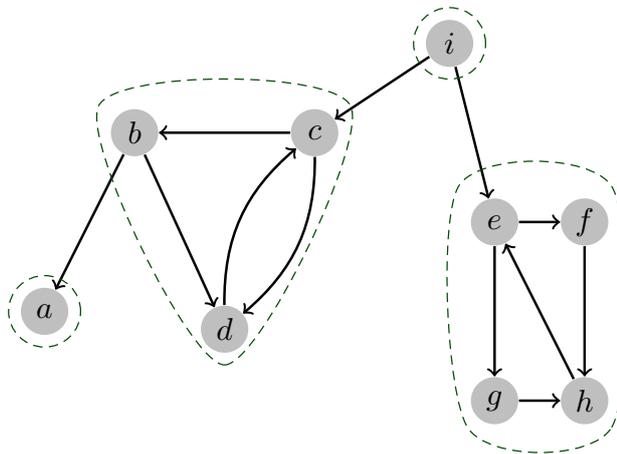


Figura 2.4: Exemplo dos componentes fortemente conexos de um grafo.

2.6 Componente conexo direcionado

Para um dado vértice x de um digrafo G , o componente conexo direcionado (DCC - *Directed Connected Component*) do ponto base x é o conjunto, denotado por $DCC_G(x)$, de todos os sucessores de x em G , ou seja, todos os nós que são alcançáveis a partir do vértice x por algum caminho (Figura 2.5). Ou seja, temos:

$$DCC_G(x) = \{t \in \mathcal{V} : \exists \pi_{x \rightsquigarrow t} \in \Pi(G)\} \quad (2.1)$$

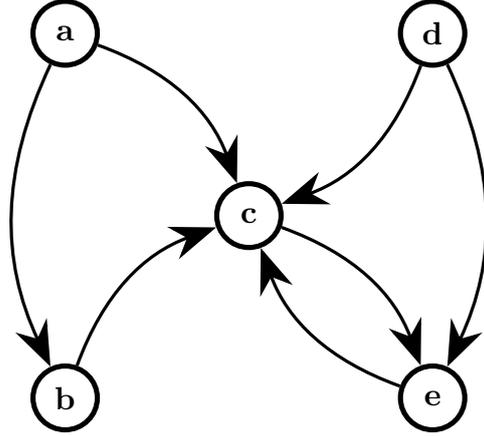


Figura 2.5: Exemplos de Componentes Conexos Direcionados (DCC) para um dado digrafo: Para o ponto base **a** temos $DCC_G(a) = \{a, b, c, e\}$, para o ponto base **b** temos $DCC_G(b) = \{b, c, e\}$, para o ponto base **c** temos $DCC_G(c) = \{c, e\}$, para o ponto base **d** temos $DCC_G(d) = \{c, d, e\}$, e para o ponto base **e** temos $DCC_G(e) = \{c, e\}$.

Note que se $c \in DCC_G(a) \implies DCC_G(c) \subseteq DCC_G(a)$.

2.7 Segmentação de imagem

A segmentação de imagem pode ser vista como um problema de classificação dos pixels em diferentes classes. Para um dado digrafo ponderado $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, em que \mathcal{V} é o conjunto finito de vértices, seja $x : \mathcal{V} \rightarrow \{0, 1, \dots, c-1\}$ uma função que classifica cada pixel em uma dentre c classes existentes. Por exemplo, no caso da segmentação em duas classes, objeto e fundo, temos $c = 2$ e $x : \mathcal{V} \rightarrow \{0, 1\}$.

2.7.1 Partição

A *partição* de um conjunto finito \mathcal{V} é um conjunto P de subconjuntos disjuntos e não vazios de \mathcal{V} , cuja união é \mathcal{V} (isto é, $\forall X, Y \in P, X \cap Y = \emptyset$ se $X \neq Y$ e $\cup\{X \in P\} = \mathcal{V}$). Qualquer elemento de uma partição P de \mathcal{V} é chamado de *região de P*. Se x é um elemento de \mathcal{V} , há uma única região de P que contém x ; esta única região é denotada por $[P]_x$. Dada duas partições P e P' de um conjunto \mathcal{V} , dizemos que P' é um *refinamento* de P se qualquer região de P' está incluída em uma região de P [Cousty *et al.* (2018)].

2.7.2 Superpixel

Um *superpixel* refere-se a uma região compacta de pixels obtida a partir do processo de particionamento de uma imagem em grupos ou agrupamentos de pixels fortemente conexos, que compartilham propriedades similares como intensidade, cor ou textura.

Idealmente, as regiões de *superpixels* devem preservar as bordas de transição entre os diferentes objetos da imagem. A técnica de *superpixels* diminui a complexidade no processamento de imagens, pois no nível de *superpixels* temos menor número de elementos da imagem. Por exemplo, a Figura 2.6a é de tamanho $520 \times 450 = 234000$ pixels, no entanto, utilizando *superpixels* de 200 pixels cada em média, o número de elementos cai para apenas de $234000/200 = 1170$ (Figura 2.6), reduzindo em 99,5% os elementos a serem processados.

Algoritmos de cálculo de *superpixels* podem ser empregados para particionar uma imagem em c regiões X_0, X_1, \dots, X_{c-1} , tal que $\cup_{j=0}^{c-1} X_j = \mathcal{I}$, sendo cada X_j um *superpixel* [Alexandre *et al.* (2015)]. *Superpixels* podem ser representados por uma imagem de rótulos $\hat{L} = (\mathcal{I}, L)$, com

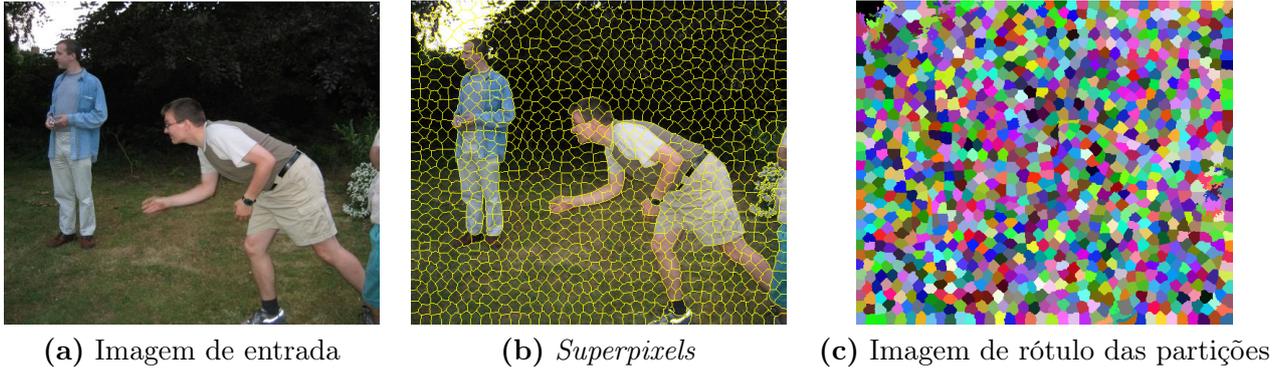


Figura 2.6: Exemplo da representação de superpixels de uma imagem. (a) Imagem original [Gulshan et al. (2010)]. (b) Contorno de superpixels. (c) Imagem de rótulos dos superpixels computados pelo método IFT-SLIC [Alexandre et al. (2015)].

$L : \mathcal{I} \rightarrow \{0, \dots, c - 1\}$, tal que $L(t) = l$ se e somente se $t \in X_l$.

Superpixels podem ser usados para a construção de um grafo de regiões adjacentes (RAG - *Region Adjacency Graph*), tal como ilustrado na Figura 2.7. O RAG é o grafo $G = \langle \mathcal{V}, \mathcal{A} \rangle$ tal que o conjunto de vértices \mathcal{V} é composto por c vértices, numerados de v_0 até v_{c-1} , sendo que cada vértice corresponde a um *superpixel*, ou seja $v_i = X_i$. Já os arcos $\langle v_i, v_j \rangle \in \mathcal{A}$ são definidos entre *superpixels* vizinhos X_i e X_j , tal que existe $p \in X_i$, $q \in X_j$ e $\|p - q\| \leq 1$.

Na Figura 2.7c, os nós do RAG estão sendo exibidos como pequenos círculos no centro geométrico dos seus *superpixels* correspondentes.

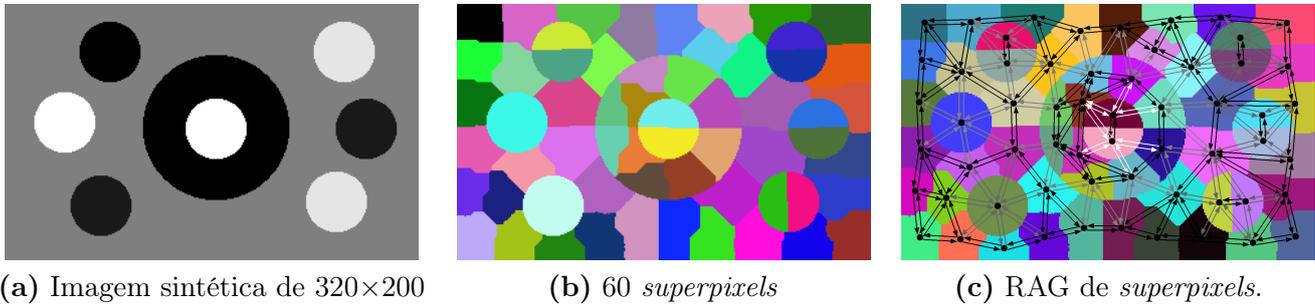


Figura 2.7: Exemplo do grafo de regiões adjacentes de uma imagem sintética.

2.7.3 Hierarquia

Uma *hierarquia* em \mathcal{V} é uma sequência $\mathcal{H} = (P_0, \dots, P_\ell)$ de partições de \mathcal{V} , tal que P_{i-1} é o refinamento de P_i , para qualquer $i \in \{1, \dots, \ell\}$. Se $\mathcal{H} = (P_0, \dots, P_\ell)$ é uma hierarquia, o número inteiro ℓ é chamado a *profundidade* de \mathcal{H} . Uma hierarquia $\mathcal{H} = (P_0, \dots, P_\ell)$ é *completa* se $P_\ell = \{\mathcal{V}\}$ e P_0 contém todos os elementos individuais de \mathcal{V} como regiões separadas (isto é, $P_0 = \{\{x\} : x \in \mathcal{V}\}$), como apresentada na Figura 2.8.

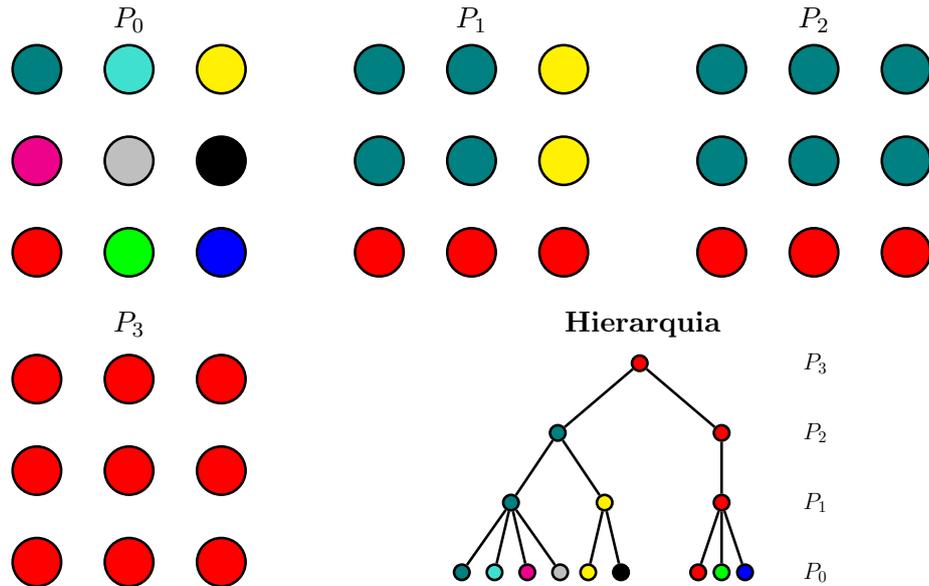


Figura 2.8: Exemplo da representação de uma hierarquia baseada em região.

Para um dado grafo derivado de uma imagem, uma relação binária entre pixels contida no produto cartesiano $\mathcal{V} \times \mathcal{V}$ que satisfaz as propriedades de reflexividade, simetria e transitividade, induz naturalmente uma segmentação da imagem, pois uma relação de equivalência permite particionar o grafo em classes de equivalência.

Por exemplo, para um grafo não direcionado, considere a relação binária $a \overset{\kappa}{\rightsquigarrow} b$, tal que existe um caminho interligando os pixels a e b , contendo apenas arestas com pesos abaixo de um dado limiar κ (isto é, \exists um caminho $\pi_{a \rightsquigarrow b} = \langle p_1, \dots, p_n \rangle$ com $a = p_1$ e $b = p_n$, tal que $\omega(\langle p_i, p_{i+1} \rangle) < \kappa$, para qualquer $i \in \{1, \dots, n - 1\}$).

1. $a \overset{\kappa}{\rightsquigarrow} a$ (reflexividade)
2. $a \overset{\kappa}{\rightsquigarrow} b \Rightarrow b \overset{\kappa}{\rightsquigarrow} a$ (simetria)
3. $a \overset{\kappa}{\rightsquigarrow} b$ e $b \overset{\kappa}{\rightsquigarrow} c \Rightarrow a \overset{\kappa}{\rightsquigarrow} c$ (transitividade)

As partições geradas por essas relações binárias correspondem aos componentes conexos do grafo, após a remoção de todas as arestas com pesos maiores ou iguais à κ . Variando o parâmetro κ , é possível construir uma hierarquia de partições (Figura 2.9).

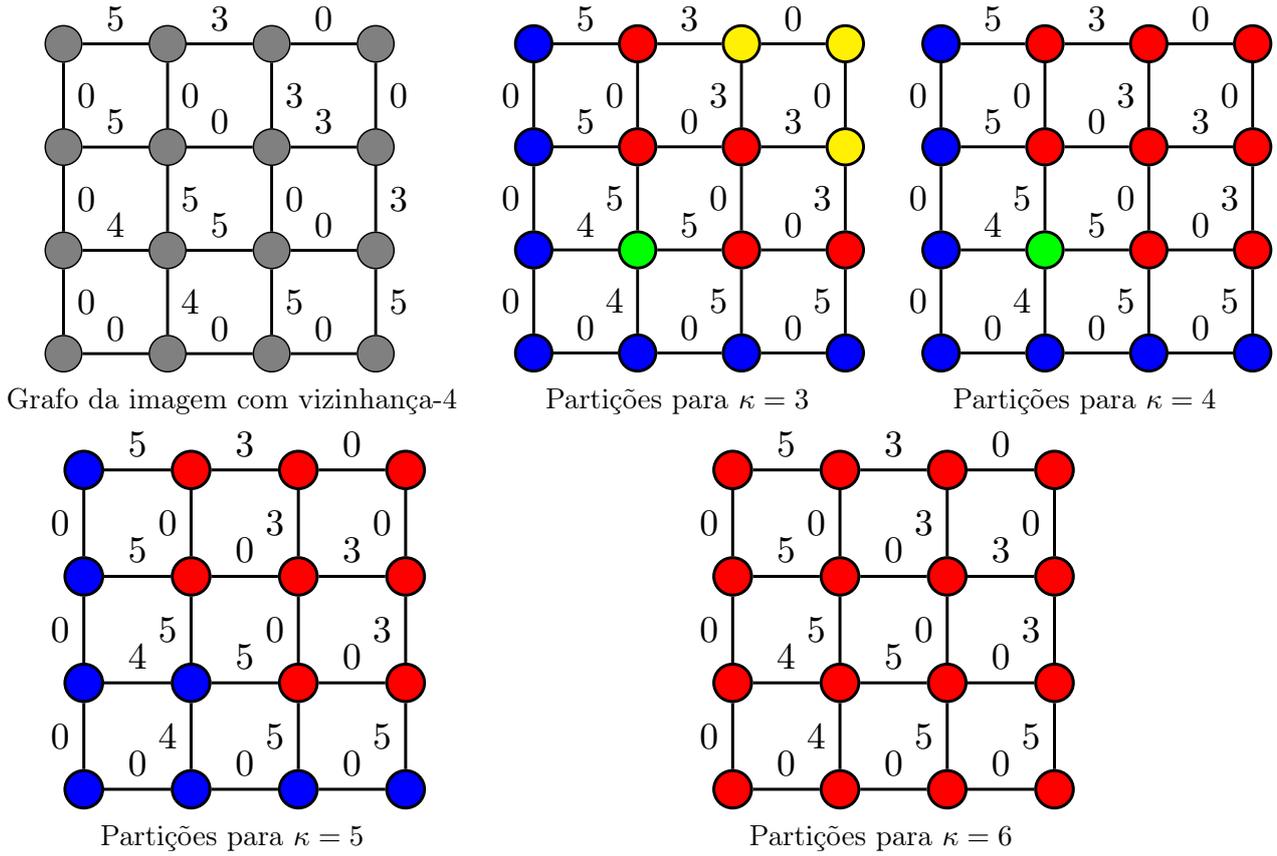


Figura 2.9: Exemplo da representação de uma hierarquia de zona quase plana.

O procedimento acima, para a criação de hierarquias tal como descrito na Figura 2.9, é conhecido como hierarquia de zona quase plana (*quasi-flat zone hierarchy* [Cousty *et al.* (2018)]).

Dado um Grafo $G = (\mathcal{V}, \mathcal{A}, \omega)$, uma partição de \mathcal{V} é conexa (em G) se todas as suas regiões são conexas e uma hierarquia de \mathcal{V} é conexa (em G) se todas as suas partições são conexas. Uma hierarquia conexa pode ser tratada de forma equivalente por meio de um grafo ponderado nas arestas, com $\omega : \mathcal{A} \rightarrow \mathbb{R}$ sendo a função de peso, tal como explicado na próxima seção.

2.7.4 Correspondência entre hierarquias e mapas de saliência

Vimos que qualquer grafo ponderado nas arestas induz uma hierarquia conexa de partições (chamada de hierarquia de zona quase plana). Nessa seção, abordamos o problema inverso, isto é, dada uma hierarquia conexa \mathcal{H} , encontre um mapa ω , de modo que a hierarquia de zona quase plana para ω é precisamente \mathcal{H} . Veremos que os mapas de saliência (*saliency map*) fornecem uma solução a esse problema [Cousty *et al.* (2018)].

Seja P uma partição de \mathcal{V} , o *corte* de P em um grafo G , definido por $\phi_G(P)$, é o conjunto de arestas de G composta por dois vértices em diferentes regiões de P (Equação 2.2):

$$\phi_G(P) = \{\{x, y\} \in \mathcal{A} : [P]_x \neq [P]_y\}. \quad (2.2)$$

Seja $\mathcal{H} = (P_0, \dots, P_\ell)$ a hierarquia em \mathcal{V} . O mapa de saliência de \mathcal{H} é o mapa $\Phi_G(\mathcal{H})$ do conjunto de arestas \mathcal{A} para $\{0, \dots, \ell\}$, de modo que o peso de qualquer aresta u para $\Phi_G(\mathcal{H})$ é o máximo valor λ para o qual u pertence ao corte de P_λ (Figura 2.12):

$$\Phi_G(\mathcal{H})(u) = \max\{\lambda \in \{0, \dots, \ell\} : u \in \phi_G(P_\lambda)\} \quad (2.3)$$

Por exemplo, considere o exemplo abaixo de uma hierarquia conexa $\mathcal{H} = (P_0, \dots, P_\ell)$ com $\ell = 3$ em

um grafo de vizinhança-4 (Figura 2.10).

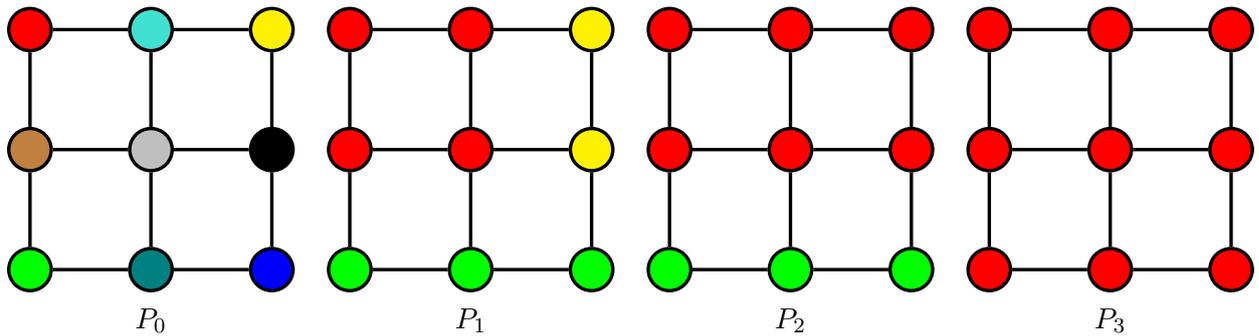


Figura 2.10: Hierarquia conexa $\mathcal{H} = (P_0, \dots, P_\ell)$ com $\ell = 3$ em um grafo de vizinhança-4.

Abaixo são mostradas as arestas pertencentes aos cortes nas partições $\phi_G(P_i)$ para cada uma das partições P_i da hierarquia para $0 \leq i \leq \ell$ (Figura 2.11).

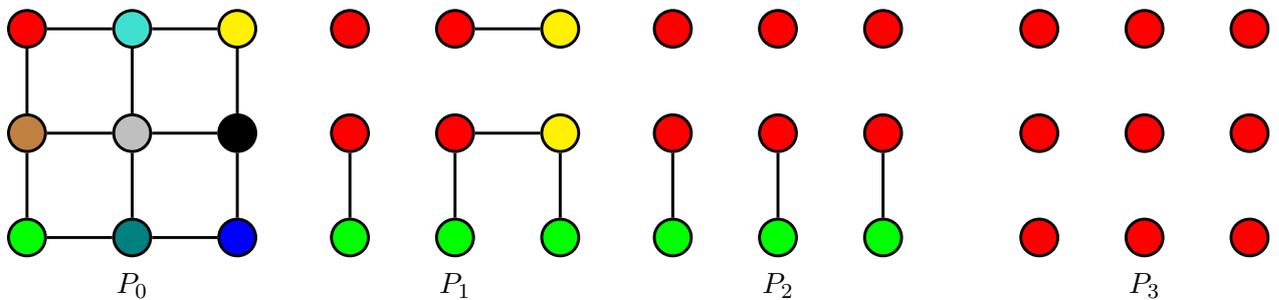


Figura 2.11: Cortes $\phi_G(P_i)$ para cada uma das partições P_i para $0 \leq i \leq \ell$.

O referido mapa de saliência $\Phi_G(\mathcal{H})$ da hierarquia apresentada na Figura 2.11a é apresentado na Figura 2.12. Além disso, a visualização de um mapa de saliência na forma de uma imagem é possível quando o grafo possui vizinhança-4. A visualização é obtida através da *grade de Khalimsky* [Cousty *et al.* (2018); Kovalevsky (1989)] para uma melhor análise da imagem (Figuras 2.12b-c).

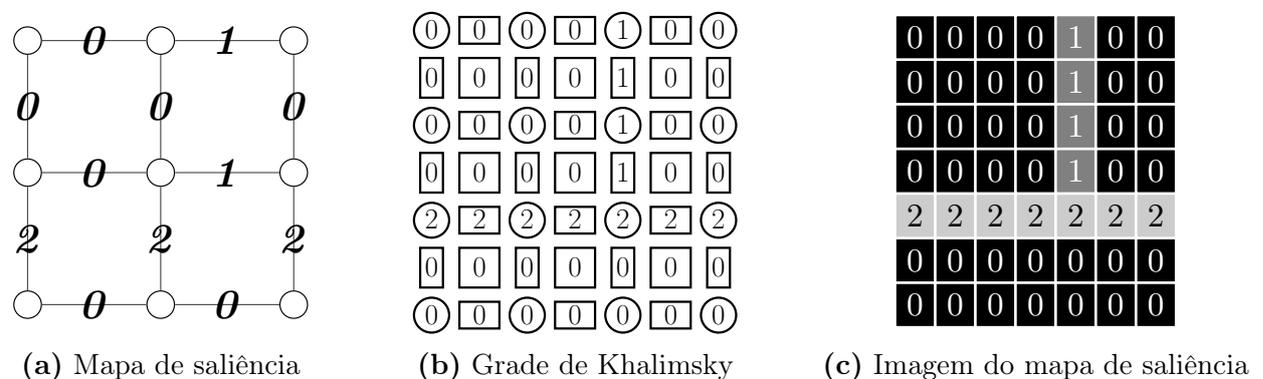


Figura 2.12: Representação do mapa de saliência $\Phi_G(\mathcal{H})$: (a) Mapa de saliência da hierarquia apresentada na Figura 2.10. (b) Grade de Khalimsky calculada a partir do mapa de saliência. (c) Imagem do mapa de saliência.

2.8 Medidas de avaliação estatística

Nesta seção, apresentamos as medidas de avaliação que serão utilizadas ao longo desta tese.

2.8.1 Matriz de confusão

Considera-se um conjunto de amostras, em que uma instância b (elemento que não se sabe a qual classe pertence e deseja-se classificá-lo) pode assumir valores no conjunto $\{0, 1\}$, **negativo** e **positivo** respectivamente. No contexto de segmentação binária, temos $b \in \mathcal{I}$ e queremos descobrir a sua classe $x(b) \in \{0, 1\}$. Considere $\hat{x}(b) : \mathcal{I} \rightarrow \{0, 1\}$ como sendo a classificação ideal de uma segmentação perfeita (gabarito).

Se a instância b é **positiva** ($\hat{x}(b) = 1$) e é classificada como **positiva** ($x(b) = 1$), conta-se como **verdadeira positiva** (b pertence ao conjunto VP); se é classificada como **negativa** ($x(b) = 0$), conta-se como **falsa negativa** (conjunto FN). Se a instância é **negativa** ($\hat{x}(b) = 0$) e é classificada como tal, conta-se como **verdadeira negativa** (conjunto VN); se é classificada como **positiva** ($x(b) = 1$), conta-se como **falsa positiva** (conjunto FP). Portanto, dado um método de segmentação (classificador) e um conjunto de instâncias pode-se construir uma matriz de confusão de 2 por 2, no caso de 2 classes. Esta matriz serve como base para muitas medidas de avaliação da qualidade da classificação. A Tabela 2.1 mostra a matriz de confusão.

		Classe Verdadeira	
		1	0
Classe hipotética	1	#Verdadeiro Positivo = VP	#Falso Positivo = FP
	0	#Falso Negativo = FN	#Verdadeiro Negativo = VN
Total colunas		VP + FN	FP + VN

Tabela 2.1: Matriz de confusão gerada a partir de um conjunto de instâncias.

O conjunto de Equações 2.4 demonstra algumas métricas que podem ser calculadas a partir da matriz de confusão, tais como a taxa de acerto (**recall** ou **tp rate**), a taxa de falso alarme (**fp rate**), a precisão (**precision**) e a acurácia (**accuracy**).

$$\begin{aligned}
 precision &= \frac{|VP|}{|VP| + |FP|} & fp\ rate &= \frac{|FP|}{|FP| + |VN|} \\
 accuracy &= \frac{|VP| + |VN|}{|VP| + |FN| + |FP| + |VN|} & recall = tp\ rate &= \frac{|VP|}{|VP| + |FN|}
 \end{aligned} \tag{2.4}$$

Na Figura 2.13, um exemplo de uma imagem de 15×13 pixels é exibida, sendo a imagem segmentada em duas classes (classe 0 em cor preta e Classe 1 em cor branca). O número de elementos das classes 0 e 1 são respectivamente, na segmentação obtida $|A_0| = 103$ e $|A_1| = 92$; e no gabarito $|B_0| = 107$ e $|B_1| = 88$.

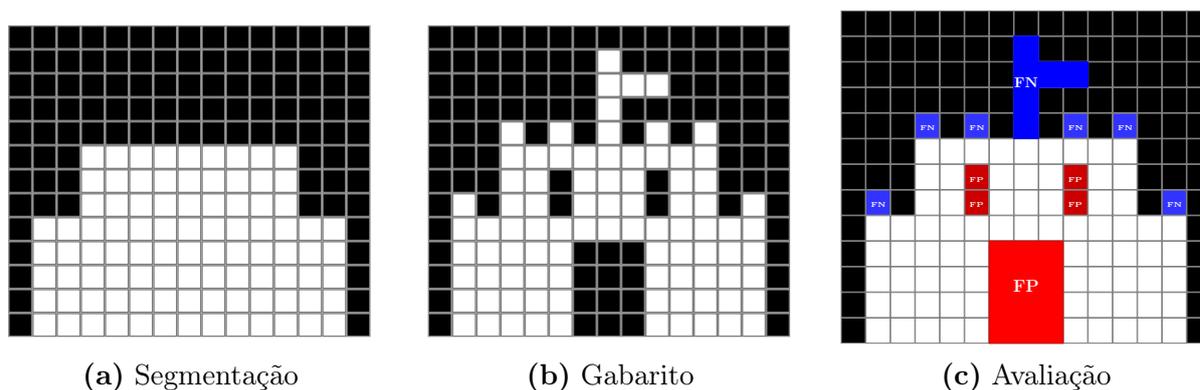


Figura 2.13: Exemplo de segmentação de uma imagem em 2 classes. (a) segmentação obtida, (b) segmentação esperada (gabarito) e (c) instâncias segmentadas de forma errada, sendo as caixas em vermelho os falsos positivos (FP) e os de cor azul os falsos negativos (FN).

A matriz de confusão correspondente ao exemplo da Figura 2.13 é apresentada na Tabela 2.2:

	Pixels que correspondendo à classe 1...	Pixels que correspondendo à classe 0...
...foram classificados como 1	76	16
...foram classificados como 0	12	91

Tabela 2.2: Matriz de confusão do exemplo da Figura 2.13.

Infelizmente, as medidas definidas pela Equação 2.4 não são apropriadas para o problema de segmentação de imagem, de modo que outras medidas mais sofisticadas têm sido empregadas na prática, conforme descrito a seguir.

2.8.2 Coeficiente de similaridade de *Dice*

O coeficiente de similaridade *Dice* (DSC) é usado como uma medida de avaliação estatística da acurácia de desempenho das segmentações de imagens (dados binários), baseado na comparação entre o número de acertos e o número total de elementos dos objetos envolvidos. O seu valor varia entre 0 (sem sobreposição) e 1 (sobreposição completa), sendo calculado a partir da seguinte fórmula: $DSC = 2|A \cap B|/(|A| + |B|)$, onde A representa a saída de uma segmentação particular ($A = \{a \in \mathcal{I} : x(a) = 1\}$), B é o gabarito ($B = \{b \in \mathcal{I} : \hat{x}(b) = 1\}$) e “ \cap ” é a interseção, conforme apresentado na Figura 2.14:

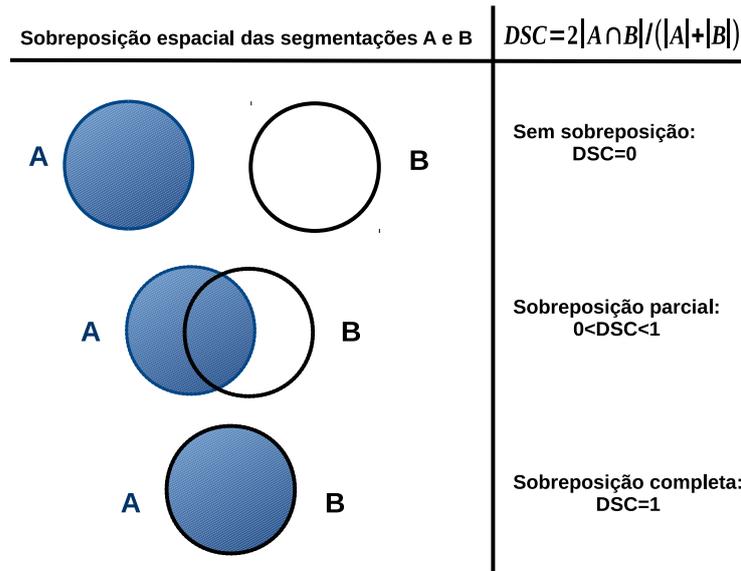


Figura 2.14: O coeficiente de Dice representando a sobreposição espacial e acurácia, tal que $DSC = 2 \times (\text{cardinalidade da região de interseção}) / (\text{soma da cardinalidade da região A e da região B})$.

Segundo a Figura 2.15, o coeficiente de Dice pode ser também calculado baseado nos valores nos valores da matriz de confusão pela equação:

$$DSC = \frac{2 \times |VP|}{|FP| + |FN| + 2 \times |VP|}$$

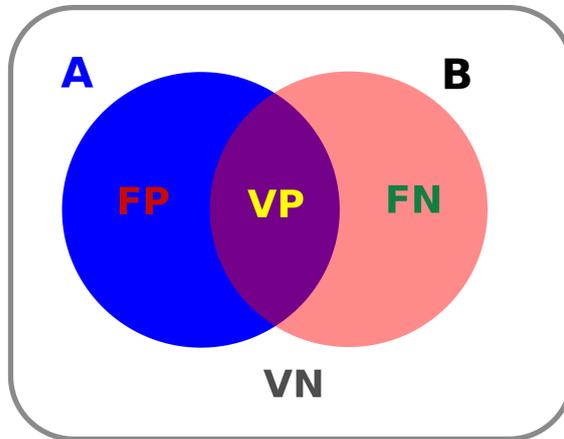


Figura 2.15: O coeficiente de Dice segundo a matriz de confusão avaliando a classe 1 (objeto). O conjunto A é a segmentação obtida e o conjunto B é a segmentação esperada (gabarito).

Para o exemplo da Figura 2.13, o cálculo do coeficiente de Dice da classe 1 (pixels brancos - objeto) é:

$$DSC = \frac{2 \times 76}{16 + 12 + 2 \times 76} = 0,84$$

2.8.3 Coeficiente de Jaccard

O Coeficiente de similaridade de Jaccard esta definido pela Equação 2.5:

$$JC = \frac{|VP|}{|VP| + |FN| + |FP|} \tag{2.5}$$

Para o exemplo da Figura 2.13, o cálculo do coeficiente de *Jaccard* da classe 1 (pixels brancos - objeto) é:

$$JC = \frac{76}{76 + 12 + 16} = 0,73$$

2.8.4 Coeficiente Geral Balanceado

O Coeficiente Geral Balanceado (GBC - *General Balanced Coefficient*) [Cappabianco *et al.* (2019)] é uma generalização de várias medidas de avaliação, incluindo *Jaccard* e *Dice*. De forma geral ela é definida pela Equação 2.6, a qual utiliza uma constante δ que a ajuda em uma maior capacidade de generalização:

$$GB_n = \frac{\delta|VP|}{\delta|VP| + |FN| + FPS_n} \quad (2.6)$$

$$FPS_n = \sum_{i=0}^n \frac{|FP|^{i+1}}{|\mathcal{O}|^i} \quad (2.7)$$

tal que VP, FP e FN representam os conjuntos de pixels verdadeiros positivos, falsos positivos e falsos negativos, respectivamente, e o \mathcal{O} é o conjunto de pixels que representa o objeto de interesse na imagem do gabarito. GB_n com $\delta = 1$ é uma generalização do coeficiente de *Jaccard*. Note que para $\delta = 1$ temos $GB_0 = JC$ e GB_1 equivale a medida do *Jaccard* balanceado (*Balanced Jaccard*) tal como proposta em Cappabianco *et al.* (2017). Já para $\delta = 2$, temos uma generalização do coeficiente de *Dice*. Observe que para $\delta = 2$, temos $GB_0 = DSC$.

2.8.5 Avaliação de segmentação interativa por usuário robô

Neste método de avaliação [Gulshan *et al.* (2010)], o comportamento das interações de um usuário humano é simulado. A cada iteração, novos marcadores são selecionados no centro geodésico (ponto mais afastado da borda do componente) da maior região conexa erroneamente rotulada em relação ao seu gabarito. Esses marcadores possuem tamanho fixo (e.g., 4 pixels de raio), sendo que no caso de marcadores próximos ao contorno do gabarito, seu tamanho é reduzido para evitar a sua seleção além do limite. A Figura 2.16 mostra um exemplo da dinâmica de execução de um usuário robô considerando apenas os erros de falso negativo.

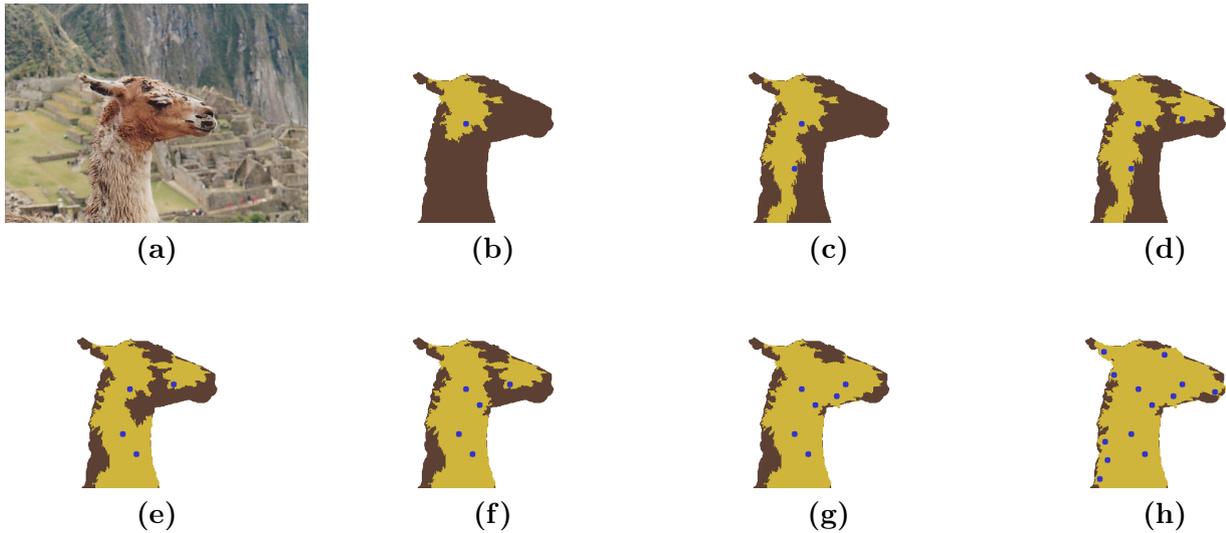


Figura 2.16: Avaliação por usuário robô: (a) Imagem original, (b-h) simulação de segmentação baseada no gabarito com adição de novas sementes nas componentes conexas de maior erro de falso negativo a cada iteração. A cada iteração, as regiões em marrom representam os falso negativos e as regiões em amarelo mostarda representam o objeto segmentado, sendo as sementes de objeto representadas em azul.

Embora, nosso objetivo seja a segmentação não supervisionada de imagens, inicialmente conduzimos um estudo da viabilidade da incorporação de restrições de curvatura no paradigma de segmentação interativa com seleção de marcadores pelo usuário. Portanto, a avaliação por usuário robô é utilizada em um primeiro momento no Capítulo 4. Posteriormente, esses resultados são estendidos para o paradigma de segmentação não supervisionada.

Capítulo 3

Métodos Relacionados

3.1 Transformada Imagem-Floresta como Algoritmo Base

A Transformada Imagem-Floresta (IFT - *Image Foresting Transform*) é uma extensão do algoritmo de Dijkstra em grafos que trata o problema da floresta de caminhos ótimos para funções de conectividade mais genéricas [Ciesielski *et al.* (2018); Falcão *et al.* (2004)].

A IFT trata a imagem como um grafo, no qual, normalmente, os pixels são os nós e os arcos são definidos por uma relação de adjacência entre os pixels vizinhos. Usa-se geralmente a vizinhança-4 (ou vizinhança de quarteirão - *city-block*), em que cada pixel é considerado adjacente aos pixels pertencentes ao disco euclidiano de raio unitário centrado nele, ou a vizinhança-8 (ou vizinhança de xadrez - *chessboard*), em que cada pixel/nó tem arcos para os pixels/nós situados a uma distância inferior ou igual a $\sqrt{2}$ (Figura 2.3). Observe porém, que outros grafos derivados de imagem, tal como discutido na Seção 2.3, poderiam ser igualmente considerados. Em especial, é comum o uso de um grafo de regiões adjacentes (RAG), tal como ilustrado na Figura 2.7. Nesta seção, por questões de simplicidade, vamos apresentar a IFT em grafos no nível de pixels (isto é, $\mathcal{V} = \mathcal{I}$).

A IFT calcula uma partição ótima dos pixels/nós em árvores de custo mínimo. Portanto, cada pixel/nó recebe os seguintes atributos na floresta computada: um rótulo específico da árvore à qual ele pertence, um predecessor na árvore e um valor de custo de caminho associado.

Tal como previamente discutido na Seção 1.1, temos que a IFT possui as seguintes características:

- **Unificação:** Vários operadores de imagem são derivados de um algoritmo geral. Isto favorece implementações baseadas em hardware [Cappabianco *et al.* (2007)], facilitando a compreensão das relações entre alguns operadores de imagem e favorecendo possíveis extensões.
- **Eficiência:** A maioria dos operadores de imagem podem ser implementados em tempo linear e otimizações adicionais são possíveis com cálculo diferencial [Falcão e Bergo (2004)] e paralelo [Bergo e Falcão (2007)].
- **Simplicidade:** Os operadores de imagem são reduzidos à escolha de poucos parâmetros no algoritmo da IFT, seguido por um processamento local dos atributos da floresta obtida.

A seguir, é apresentada uma relação de aplicações que podem ser atualmente tratadas via IFT:

- **Transformadas de distância e operadores relacionados:** Transformada da distância euclidiana, esqueleto multiescala [Falcão *et al.* (2002)], dimensões fractais [Torres *et al.* (2004)], filtragem de forma, detecção de saliências de forma, descritores de forma [Andaló *et al.* (2007); da S. Torres e Falcão (2007)], caminhos geodésicos [de Moraes Braz e Miranda (2014)].
- **Filtragem e segmentação de imagens:** Reconstrução morfológica [Falcão *et al.* (2001)], segmentação de imagem baseada na transformada de *Watershed* [Lotufo e Falcão (2000)],

perseguição de bordas por *Live Wire* [Falcão *et al.* (1998)] e *Riverbed* [Miranda *et al.* (2012)], *Growcut* por autômato celular [Miranda e Falcão (2011); Vezhnevets e Konouchine (2005)] e métodos de conectividade *fuzzy* [Ciesielski *et al.* (2012a)].

- **Reconhecimento de padrões:** *Clustering* de dados [Rocha *et al.* (2009)] e classificação de padrões supervisionada [Papa *et al.* (2007)].

Neste trabalho, a IFT será apresentada de modo mais específico para o problema da segmentação de imagens.

3.1.1 Função de conectividade

Uma função de conectividade $f(\pi)$ associa um valor escalar a qualquer caminho π no grafo $G = \langle \mathcal{V} = \mathcal{I}, \mathcal{A}, \omega \rangle$, com base em propriedades da imagem ao longo deste caminho. Em segmentação, por exemplo, propriedades locais da imagem (e.g., cor, intensidade, vetor gradiente e posição dos pixels ao longo do caminho) e globais do objeto desejado (e.g., modelo esperado de textura/cor do objeto a partir de uma base de treinamento e restrições de forma do domínio da aplicação) podem ser exploradas no cálculo de $f(\pi_{s \rightsquigarrow t})$ para indicar o custo de conectividade entre seus nós terminais s e t , através do caminho $\pi_{s \rightsquigarrow t}$. As funções de conectividade são normalmente especificadas por uma regra de inicialização e uma regra de extensão de caminho. Por exemplo, considere as funções f_{max} , f_{sum} , f_{ω} e f_{euc} :

$$\begin{aligned} f_{max}(\langle t \rangle) &= H(t) \\ f_{max}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{max}(\pi_s), \omega(\langle s, t \rangle)\} \end{aligned} \quad (3.1)$$

$$\begin{aligned} f_{sum}(\langle t \rangle) &= H(t) \\ f_{sum}(\pi_s \cdot \langle s, t \rangle) &= f_{sum}(\pi_s) + \omega(\langle s, t \rangle) \end{aligned} \quad (3.2)$$

$$\begin{aligned} f_{\omega}(\langle t \rangle) &= H(t) \\ f_{\omega}(\pi_s \cdot \langle s, t \rangle) &= \omega(\langle s, t \rangle) \end{aligned} \quad (3.3)$$

$$f_{euc}(\langle p_1, \dots, p_n \rangle) = \begin{cases} \|p_n - p_1\|^2 & \text{se } p_1 \in \mathcal{S} \\ +\infty & \text{caso contrário} \end{cases} \quad (3.4)$$

tal que $H(t)$ é um valor inicial para caminhos triviais e temos $\omega(\langle s, t \rangle) \geq 0$ para f_{sum} . Em segmentação de imagens, geralmente usamos:

$$H(t) = \begin{cases} 0 & \text{se } t \in \mathcal{S} \\ +\infty & \text{caso contrário} \end{cases} \quad (3.5)$$

em que $\mathcal{S} \subseteq \mathcal{V}$ é o conjunto de sementes.

Exemplos de custo das funções f_{max} , f_{sum} , f_{ω} e f_{euc} são apresentados na Figura 3.1. A função f_{max} é utilizada na implementação de filtros conexos e em métodos de segmentação por transformada de *Watershed* e conectividade *fuzzy*. Já a função f_{sum} é muito empregada em métodos de perseguição de bordas. Ao passo que a função f_{euc} é empregada para calcular uma aproximação da Transformada de distância Euclidiana, esqueleto multiescala e descritores de forma. A IFT com função f_{ω} degenera em um algoritmo de Prim para o cálculo de uma floresta de espalhamento mínima ou, no caso de uma única semente, em uma árvore de espalhamento mínima (MST - *Minimum Spanning Tree*).

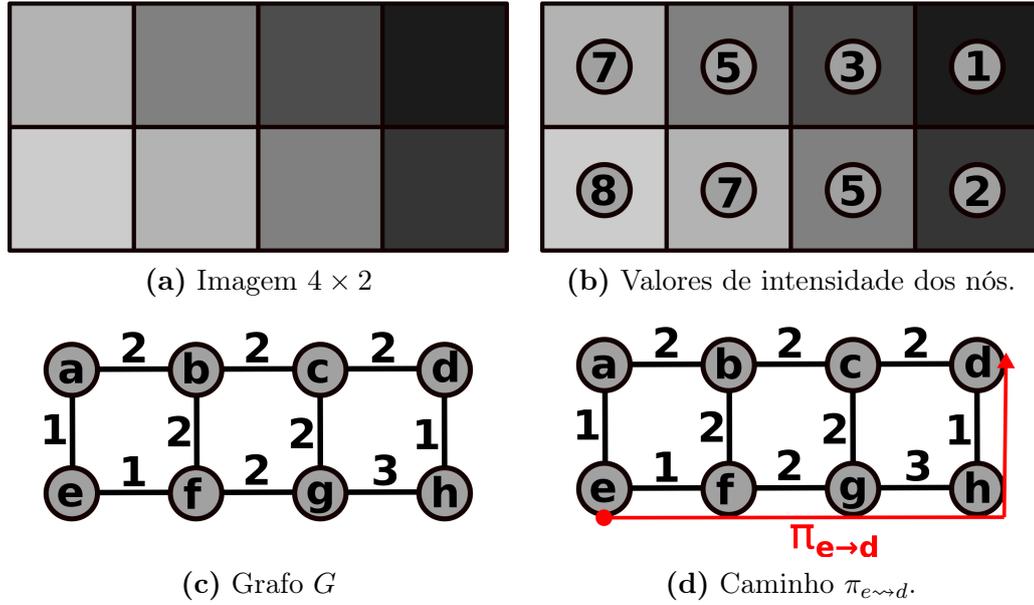


Figura 3.1: Exemplo do cálculo de custo de um caminho para diferentes funções de conectividade em uma imagem 2D em tons de cinza: (a) Imagem de tamanho 4×2 pixels. (b) Criação dos nós do grafo para cada pixel t da imagem, com a indicação dos valores de intensidade $I(t)$ dentro dos nós. (c) Grafo $G = \langle \mathcal{I}, \mathcal{A}, \omega \rangle$ não direcionado, com vizinhança-4 e pesos $\omega(\langle s, t \rangle) = |I(t) - I(s)|$. (d) Caminho $\pi_{e \rightsquigarrow d} = \langle e, f, g, h, d \rangle$, com custos para diferentes funções de conectividade dados por: $f_{max}(\pi_{e \rightsquigarrow d}) = \max\{0, 1, 2, 3, 1\} = 3$ para $H(e) = 0$, $f_{sum}(\pi_{e \rightsquigarrow d}) = 0 + 1 + 2 + 3 + 1 = 7$ para $H(e) = 0$, $f_{\omega}(\pi_{e \rightsquigarrow d}) = \omega(\langle h, d \rangle) = 1$ e $f_{euc}(\pi_{e \rightsquigarrow d}) = 3^2 + 1^2 = 10$, considerando $\mathcal{S} = \{e\}$.

3.1.2 Função Monotonicamente Incremental

Uma função de conectividade é Monotonicamente Incremental (MI - *Monotonic-Incremental cost function*), se ela pode ser descrita pela forma:

$$\begin{aligned} f_{mi}(\langle t \rangle) &= H(t), \\ f_{mi}(\pi_s \cdot \langle s, t \rangle) &= f_{mi}(\pi_s) \odot \langle s, t \rangle, \end{aligned}$$

em que \odot é uma operação binária entre o custo de um caminho e um arco que satisfaz as condições:

- (M1) $f_{mi}(\pi_s) \geq f_{mi}(\tau_s) \Rightarrow f_{mi}(\pi_s) \odot \langle s, t \rangle \geq f_{mi}(\tau_s) \odot \langle s, t \rangle$,
- (M2) $f_{mi}(\pi_s) \odot \langle s, t \rangle \geq f_{mi}(\pi_s)$,

Uma característica essencial deste modelo de funções é que \odot depende apenas do custo de π_s e não de qualquer outra propriedade deste caminho.

As funções f_{max} e f_{sum} são casos particulares de funções MI, enquanto que f_{ω} e f_{euc} são exemplos de funções *Não-Monotonicamente Incrementais* (NMI - *Non-Monotonic-Incremental cost functions*).

3.1.3 Caminho ótimo

Para um dado digrafo G , um caminho π_t é **ótimo** se $f(\pi_t) \leq f(\tau_t)$ para qualquer outro caminho τ_t , independentemente de sua origem. Para cada nó $t \in \mathcal{V}$, temos um valor único $V_{opt}(t)$ que armazena o valor de um caminho ótimo com término em t :

$$V_{opt}(t) = \min_{\pi_t \in \Pi(G)} \{f(\pi_t)\}$$

3.1.4 Floresta de espalhamento como mapa de predecessores

Um mapa de predecessores é um função Pr que atribui para cada nó t em \mathcal{V} algum outro nó adjacente em \mathcal{V} , ou uma marca distintiva $nil \notin \mathcal{V}$, caso em que t é dito ser uma raiz do mapa. Uma floresta de espalhamento é um mapa de predecessores que não contém ciclos, isto é, um mapa que leva cada pixel para nil em um número finito de iterações. Para qualquer nó $t \in \mathcal{V}$, uma floresta de espalhamento Pr define um caminho π_t^{Pr} recursivamente como $\langle t \rangle$ se $Pr(t) = nil$ e $\pi_t^{Pr} = \pi_s^{Pr} \cdot \langle s, t \rangle$ se $Pr(t) = s \neq nil$ (Figura 3.2).

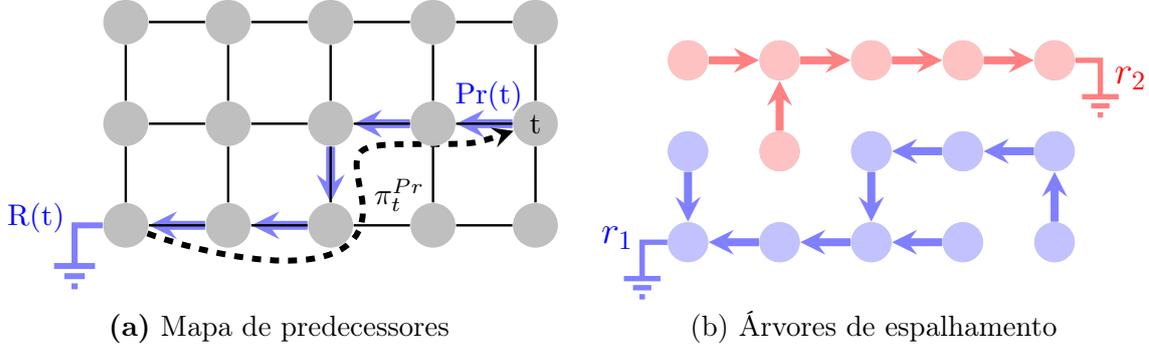


Figura 3.2: Floresta de espalhamento: (a) O mapa de predecessores Pr , armazena os caminhos em ordem inversa. Para um dado nó t , seguindo pelo mapa de predecessores chegamos a um nó raiz $R(t)$ tal que $Pr(R(t)) = nil$. Um caminho π_t é trivial quando $\pi_t = \langle t \rangle$ (isto é, $Pr(t) = nil$). (b) Duas árvores de espalhamento, uma em vermelho e a outra em azul.

Na IFT, as raízes são formadas por caminhos ótimos triviais. Estas raízes competem entre si pela conquista dos demais nós de acordo com os custos dos caminhos fornecidos, particionando a imagem em uma floresta em que cada raiz forma uma árvore de caminhos ótimos.

3.1.5 Algoritmo geral da IFT

A Transformada Imagem-Floresta (Algoritmo 1) calcula o mapa de custos V que é precisamente V_{opt} no caso de funções MI [Ciesielski *et al.* (2018)]. Também é otimizado para lidar com custos infinitos, armazenando em \mathcal{Q} apenas os nós com caminho de custo finito, assumindo sem perda de generalidade que $V_{opt}(t) < +\infty$ para todos os $t \in \mathcal{V}$.

Algoritmo 1. – TRANSFORMADA IMAGEM-FLORESTA

ENTRADA: Imagem $\hat{I} = (\mathcal{I}, \vec{I})$, $G = \langle \mathcal{I}, \mathcal{A}, \omega \rangle$, função de conectividade f e uma função de rotulação inicial $\mathcal{L} : \mathcal{I} \rightarrow \{0, \dots, c-1\}$.
 SAÍDA: Mapa de predecessores $Pr : \mathcal{I} \rightarrow \mathcal{I} \cup \{nil\}$, mapa de rótulos $L : \mathcal{I} \rightarrow \{0, \dots, c-1\}$ e mapa de custo de conectividade $V : \mathcal{I} \rightarrow \mathbb{R}$, que converge para V_{opt} no caso de funções MI.
 AUXILIARES: Fila de prioridade \mathcal{Q} , variável tmp e vetor de estado inicialmente zerado.

1. **Para Cada** $t \in \mathcal{I}$, **Faça**
2. $Pr(t) \leftarrow nil$, $V(t) \leftarrow f(\langle t \rangle)$ e $L(t) \leftarrow \mathcal{L}(t)$.
3. **Se** $V(t) \neq \infty$, **Então**
4. insira t em \mathcal{Q} .
5. **Enquanto** $\mathcal{Q} \neq \emptyset$, **Faça**
6. Remova um pixel s de \mathcal{Q} cujo valor $V(s)$ seja mínimo.
7. estado(s) $\leftarrow 1$.
8. **Para Cada** t , tal que $\langle s, t \rangle \in \mathcal{A}$ e estado(t) = 0, **Faça**
9. $tmp \leftarrow f(\pi_s^{Pr} \cdot \langle s, t \rangle)$.
10. **Se** $tmp < V(t)$, **Então**
11. **Se** $V(t) \neq \infty$, **Então**
12. remova t de \mathcal{Q} .
13. $Pr(t) \leftarrow s$, $V(t) \leftarrow tmp$, $L(t) \leftarrow L(s)$ e insira t em \mathcal{Q} .

O algoritmo inicia com todos os caminhos triviais (Linhas 1 e 2). Na Linha 4, todos os candidatos a raízes (sementes) são inseridos na fila Q . As raízes são os mínimos de V , primeiramente detectados nas Linhas 5-6, pois, ao remover qualquer nó s de Q na Linha 6, o caminho π_s^{Pr} se torna definitivo, não podendo ser mais alterado. As demais linhas calculam e propagam caminhos melhores para os nós adjacentes, aplicando a seguinte regra de extensão: se $f(\pi_s^{Pr} \cdot \langle s, t \rangle) < f(\pi_t^{Pr})$ (Linha 10), então π_t^{Pr} é trocado por $\pi_s^{Pr} \cdot \langle s, t \rangle$ (Linha 13).

Considere o problema de segmentar c objetos em uma imagem, incluindo o plano de fundo. Cada objeto \mathcal{O}_l é associado a um rótulo $l \in \{0, \dots, c-1\}$, sendo $l = 0$ usado para indicar o fundo. No contexto de segmentação por região baseada em sementes, pixels são amostrados no interior de cada objeto e no fundo, definindo um conjunto de pixels sementes \mathcal{S} . Cada semente $s \in \mathcal{S}$ é inicializada com o rótulo $\mathcal{L}(s) = l \in \{0, \dots, c-1\}$ de seu objeto \mathcal{O}_l correspondente, sendo $\mathcal{L}: \mathcal{S} \rightarrow \{0, \dots, c-1\}$ a função de rotulação inicial das sementes¹. Ao final do algoritmo, cada objeto segmentado \mathcal{O}_l é definido como o conjunto $L^{-1}(l) := \{t \in \mathcal{I}: L(t) = l\}$. Por exemplo, no caso de segmentação binária (isto é, $\mathcal{L}: \mathcal{S} \rightarrow \{0, 1\}$), o objeto segmentado de primeiro plano é definido como $\mathcal{O}_1 = L^{-1}(1) = \{t \in \mathcal{I}: L(t) = 1\}$ e o fundo é dado por $\mathcal{O}_0 = L^{-1}(0) = \mathcal{I} \setminus \mathcal{O}_1$.

Vários algoritmos clássicos de segmentação podem ser obtidos a partir do Algoritmo 1 para diferentes funções de conexidade. Por exemplo, a *Watershed* de marcadores em grafos não direcionados [Lotufo e Falcão (2000)], com resolução de arestas, pode ser obtida utilizando a função de conexidade f_{max} (Equação 3.1).

3.2 Live Wire (LW)

O método de perseguição de bordas *Live Wire* [Falcão *et al.* (2000, 1998)] é um método de segmentação interativa para a extração da borda de um objeto de interesse com restrições selecionadas ao longo do seu contorno, conhecidas como pontos de ancoragem. A cada interação, um caminho de custo mínimo interligando dois pontos de ancoragem consecutivos é calculado, podendo este ser computado pela Transformada Imagem-Floresta (IFT) [Falcão *et al.* (2004)].

A função de custo de caminho do *Live Wire* é dada por:

$$\begin{aligned} f_{LW}(\langle t \rangle) &= \begin{cases} 0 & \text{se } t \in \mathcal{S} \\ +\infty & \text{caso contrário} \end{cases} \\ f_{LW}(\pi_s \cdot \langle s, t \rangle) &= f_{LW}(\pi_s) + \omega(\langle s, t \rangle) \end{aligned} \quad (3.6)$$

em que \mathcal{S} é um conjunto de sementes, geralmente composto por um único ponto de ancoragem inicial/anterior.

Conforme já foi explicado anteriormente na Figura 3.2, no arcabouço da IFT os caminhos são armazenados em ordem inversa em um mapa de predecessores $Pr: \mathcal{V} \rightarrow \mathcal{V} \cup \{nil\}$, de modo que para qualquer pixel $t \in \mathcal{V}$, o caminho π_t^{Pr} é definido recursivamente como $\langle t \rangle$ se $Pr(t) = nil$, e como $\pi_s^{Pr} \cdot \langle s, t \rangle$ se $Pr(t) = s \neq nil$.

Na segmentação dirigida pelo usuário [Falcão *et al.* (2000); Kang (2005); Miranda *et al.* (2012)], o caminho calculado do ponto de ancoragem anterior para a posição atual do cursor do mouse é mostrado ao usuário, enquanto ele move o cursor, para que o usuário possa selecionar de forma interativa o caminho desejado, que melhor se enquadra à borda do objeto. O processo então se repete e a busca de um novo caminho a partir desse ponto é então iniciada. Todos os caminhos selecionados anteriormente são mantidos inalterados (congelados) durante o algoritmo, de modo que seus nós não possam ser revisitados.

¹O valor de $\mathcal{L}(t)$, para $t \notin \mathcal{S}$, pode ser configurado de maneira arbitrária, pois o valor de $L(t)$, que inicialmente é dado por $\mathcal{L}(t)$ (Linha 2 do Algoritmo 1), será sobrescrito posteriormente durante a execução do Algoritmo 1, no caso de grafos fortemente conexos.

3.3 Transformada Imagem-Floresta Orientada (OIFT)

Seja $\mathcal{U} = \{\mathcal{O} : \mathcal{O} \subseteq \mathcal{V}\}$ o espaço de todos os possíveis objetos binários segmentados \mathcal{O} . Uma segmentação baseada em sementes utiliza $\mathcal{S} = \mathcal{S}_o \cup \mathcal{S}_b \subseteq \mathcal{V}$, tal que \mathcal{S}_o e \mathcal{S}_b são conjuntos de sementes do objeto ($\mathcal{S}_o \subseteq \mathcal{O}$) e fundo ($\mathcal{S}_b \subseteq \mathcal{V} \setminus \mathcal{O}$), respectivamente. Essas sementes restringem \mathcal{U} da seguinte forma:

$$\mathcal{U}(\mathcal{S}_o, \mathcal{S}_b) = \{\mathcal{O} \in \mathcal{U} : \mathcal{S}_o \subseteq \mathcal{O} \subseteq \mathcal{V} \setminus \mathcal{S}_b\} \quad (3.7)$$

O conjunto de arcos de corte é definido como segue:

$$\mathcal{C}(\mathcal{O}) = \{\langle s, t \rangle \in \mathcal{A} : s \in \mathcal{O} \wedge t \notin \mathcal{O}\} \quad (3.8)$$

Podemos associar um valor de energia $\varepsilon(\mathcal{O})$ para cada objeto (e seu corte) e restringir o conjunto de soluções para aqueles que o minimizam ou maximizam. Sendo que a energia é definida como:

$$\varepsilon_q(\mathcal{O}) = \sqrt[q]{\sum_{\langle s, t \rangle \in \mathcal{C}(\mathcal{O})} [\omega(\langle s, t \rangle)]^q} \quad (3.9)$$

Certamente, o algoritmo original de Corte em Grafo, por fluxo máximo/corte mínimo [Boykov e Funka-Lea (2006)], minimiza $\varepsilon_1(\mathcal{O})$. Já a Transformada Imagem-Floresta Orientada (OIFT - *Oriented Image Foresting Transform*) [Mansilla e Miranda (2013a); Miranda e Mansilla (2014)] minimiza $\varepsilon_\infty(\mathcal{O}) = \max_{\langle s, t \rangle \in \mathcal{C}(\mathcal{O})} \omega(\langle s, t \rangle)$ em um dígrafo simétrico e fortemente conexo $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, entre todas possíveis soluções em $\mathcal{U}(\mathcal{S}_o, \mathcal{S}_b)$.

Aqui apresentaremos a implementação dual da OIFT, que maximiza a energia:

$$\bar{\varepsilon}_\infty(\mathcal{O}) = \min_{\langle s, t \rangle \in \mathcal{C}(\mathcal{O})} \omega(\langle s, t \rangle) \quad (3.10)$$

Nessa implementação da OIFT, os pesos $\omega(\langle s, t \rangle)$ são a combinação de uma medida de dissimilaridade não direcionada $\delta(\langle s, t \rangle)$ entre pixels vizinhos s e t , multiplicada por um fator de orientação [Mansilla e Miranda (2013a)], tal como definido abaixo:

$$\omega(\langle s, t \rangle) = \begin{cases} \delta(\langle s, t \rangle) \times (1 + \alpha) & \text{se } I(s) > I(t) \\ \delta(\langle s, t \rangle) \times (1 - \alpha) & \text{se } I(s) < I(t) \\ \delta(\langle s, t \rangle) & \text{caso contrário} \end{cases} \quad (3.11)$$

tal que $\alpha \in [-1, 1]$ e $I(t)$ é a intensidade da imagem no pixel t . Diferentes procedimentos podem ser adotados para $\delta(\langle s, t \rangle)$, conforme discutido em [Ciesielski e Udupa (2010a); Miranda *et al.* (2010)], tal como o valor absoluto da diferença de brilho $\delta(\langle s, t \rangle) = |I(t) - I(s)|$ (Figuras 3.3a-b) ou a magnitude média do gradiente $\delta(\langle s, t \rangle) = \frac{\|\nabla I(s)\| + \|\nabla I(t)\|}{2}$. Note que usualmente se tem $\omega(\langle s, t \rangle) \neq \omega(\langle t, s \rangle)$ quando $\alpha \neq 0$ (Figura 3.3c). Para imagens coloridas, um mapa de referência deve ser considerado para $I(t)$ (por exemplo, a banda de cor verde), na Equação 3.11 ou α deve ser ajustado para zero [Mansilla e Miranda (2013a)].

A OIFT é construída sobre o arcabouço da IFT considerando a seguinte função de custo de caminho [Mansilla e Miranda (2013a)]:

$$f^\sigma(\langle t \rangle) = \begin{cases} -1 & \text{se } t \in \mathcal{S}_o \cup \mathcal{S}_b \\ +\infty & \text{caso contrário} \end{cases}$$

$$f^\sigma(\pi_{r \rightsquigarrow s} \cdot \langle s, t \rangle) = \begin{cases} \omega(\langle s, t \rangle) & \text{se } r \in \mathcal{S}_o \\ \omega(\langle t, s \rangle) & \text{caso contrário} \end{cases} \quad (3.12)$$

O objeto \mathcal{O} segmentado pela OIFT é definido a partir da floresta Pr calculada pelo Algoritmo 1, com f^σ , tomando como pixels do objeto o conjunto de pixels que foram conquistados por caminhos enraizados em \mathcal{S}_o . Para obter este efeito, basta setar a função de rotulação inicial $\mathcal{L} : \mathcal{I} \rightarrow \{0, 1\}$,

de modo que $\mathcal{L}(t) = 1$ para todo $t \in \mathcal{S}_o$ e $\mathcal{L}(t) = 0$ caso contrário, na chamada do algoritmo da IFT (Algoritmo 1). Deste modo, um rótulo, $L(t) = 1$ para todo $t \in \mathcal{S}_o$ e $L(t) = 0$ para todo $t \in \mathcal{S}_b$, é propagado para todos pixels não sementes durante a execução do algoritmo da IFT.

Para $\alpha > 0$, a segmentação por OIFT favorece transições de pixels de claro para escuro e $\alpha < 0$ favorece na orientação oposta [Mansilla e Miranda (2013a); Miranda e Mansilla (2014)]. Em geral, consideramos $\alpha = 50\%$ (ou $\alpha = -50\%$) a fim de obter uma solução mais balanceada. Esta propriedade de favorecer diferentes orientações de intensidade (claro/escuro ou escuro/claro) nas transições da borda é conhecida como polaridade de borda, tal como ilustrado na Figura 3.3 (ver também Figura 1.1e).

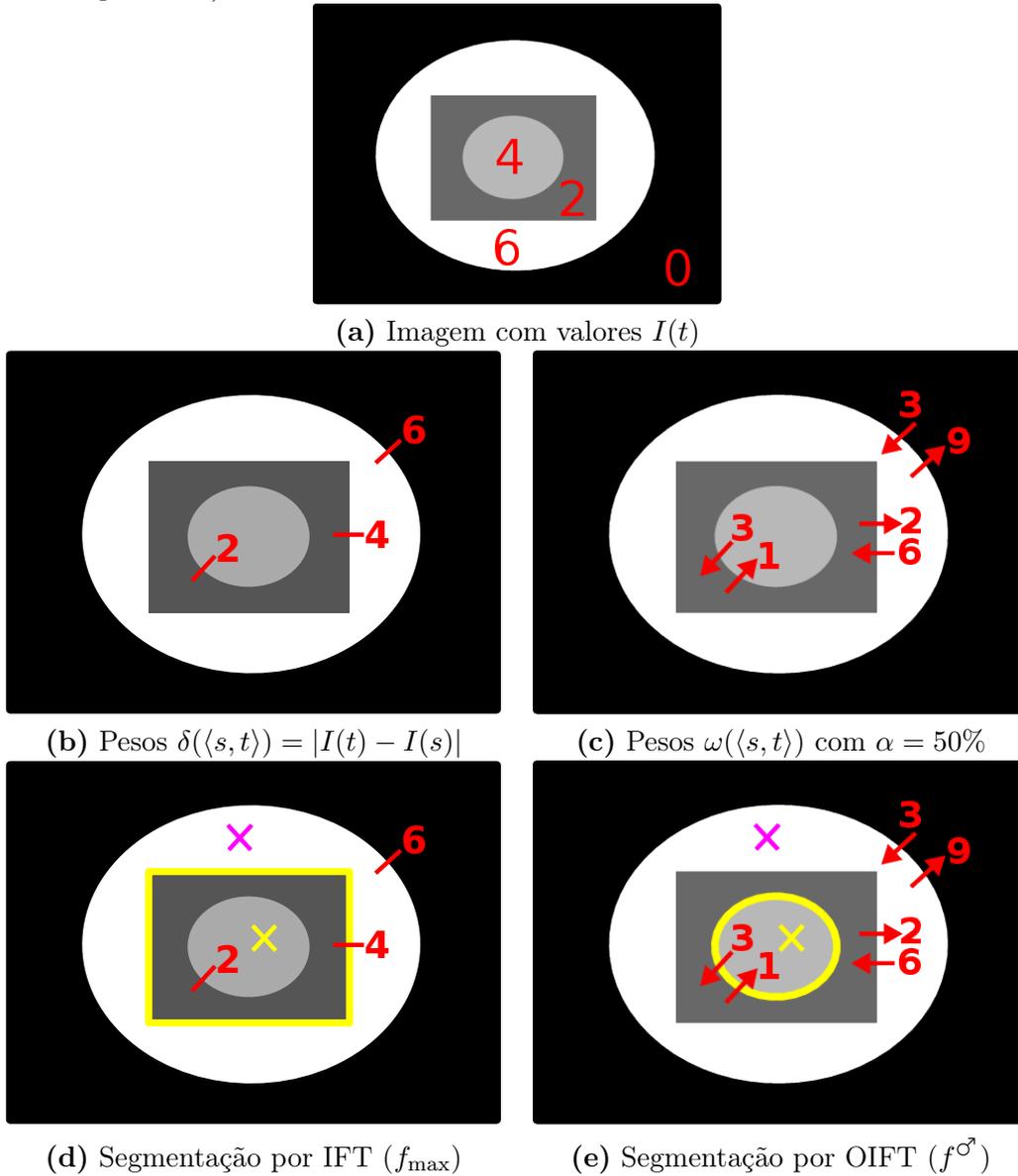


Figura 3.3: Exemplo de segmentação de imagem sintética com polaridade de borda via OIFT. (a-c) Construção do grafo pela Equação 3.11, usando $\delta(\langle s, t \rangle) = |I(t) - I(s)|$. (d) Segmentação pelo Algoritmo 1 com função de conexão f_{\max} (Equação 3.1), usando o grafo não direcionado com pesos $\delta(\langle s, t \rangle)$ da figura (b). (e) Segmentação pela OIFT, usando o grafo direcionado da figura (c) e função de conexão f^{σ} . Observe que a OIFT favoreceu uma borda com polaridade de claro para escuro, uma vez que foi utilizado $\alpha > 0$.

3.4 Métodos de Estimativa de Curvatura em Imagens Digitais

Dado que um dos objetivos desta tese é a incorporação de restrições de curvatura nos contornos dos objetos segmentados, nesta seção são apresentados os principais métodos utilizados para a extração de características de curvatura em imagens digitais. As características de curvatura têm um papel importante na análise topológica de objetos, por isso são muito utilizadas em descritores de forma [J.A. *et al.* (2022); Zeng *et al.* (2019)], embora não seja este o tema do presente trabalho.

Intuitivamente, podemos considerar a curvatura como a taxa de variação na direção da borda. Os pontos em que a direção da borda muda rapidamente são considerados cantos (*corners*). Tais pontos extremos são úteis para descrição e correspondência de formas, pois representam informações significativas com dados reduzidos [Nixon e Aguado (2019)]. Por exemplo, pontos de alta curvatura ditos como cantos podem ser usados para aproximar uma forma por um polígono [Latecki e Lakamper (2000)].

Na literatura, existem métodos de detecção de pontos de canto para aplicações como rastreamento, registro ou reconhecimento de objetos [Guru e Dinesh (2004); Harris e Stephens (1988); Shi e Tomasi (1994)]. Algumas destas técnicas utilizam janelas (regiões) com conteúdo de alta frequência espacial, com as pesquisas sugerindo a seleção de características via o cálculo de autovaleiros para identificar os cantos ou pontos de controle em regiões de alta entropia. Por outro lado, alguns algoritmos de seleção de características calculam a variação do ângulo em cada pixel ao longo de um segmento de curva (caminho) para identificar um ponto de canto [Nixon e Aguado (2019)], o que pressupõe uma pré-segmentação do contorno.

3.4.1 Estimativa de curvatura pelo cálculo de diferenças na direção da borda

Uma forma simples de calcular curvatura é através da variação angular ao longo de um caminho de pixels em uma imagem digital. Nesta abordagem calculamos a diferença na direção da borda entre os pixels conectados formando uma curva discreta, assim aproximando a derivada como as diferenças de ângulo entre pixels vizinhos. Esta abordagem foi considerada nas primeiras técnicas de detecção de cantos [Bennett e Mac Donald (1975); Groen e Verbeek (1978); Kitchen e Rosenfeld (1982)]. Para o cálculo utiliza a seguinte aproximação da definição de curvatura:

$$k(t) = \phi_{t+1} - \phi_{t-1} \quad (3.13)$$

tal que a sequência $\dots, \phi_{t-2}, \phi_{t-1}, \phi_t, \phi_{t+1}, \phi_{t+2}, \dots$ representa as direções dos gradientes de uma sequência de pixels que formam o segmento de curva. A direção do gradiente pode ser obtida pelo ângulo dado por um operador detector de borda. Ou seja, se aplicarmos um operador detector de bordas a uma imagem, teremos para cada pixel um valor de direção de gradiente que representa a direção normal a cada ponto em uma curva. Alternativamente, podemos considerar o ângulo tangente à curva $\phi_t = \arctan((y_{t+1} - y_{t-1}) / (x_{t+1} - x_{t-1}))$, tal que (x_t, y_t) é o pixel t da sequência². O cálculo da curvatura da Equação 3.13 não é robusto, por isso consideramos a diferença da direção angular média para n pixels ao longo do segmento de curva à esquerda e à direita de t , com a Equação 3.14.

$$k_n(t) = \frac{1}{n} \sum_{i=1}^n \phi_{t+i} - \frac{1}{n} \sum_{i=-n}^{-1} \phi_{t+i} \quad (3.14)$$

Para calcular a variação angular é necessário determinar bordas conexas, infelizmente esta abordagem não fornece resultados confiáveis. Uma das maiores dificuldades com esta abordagem é que as medições dos ângulos podem ser severamente afetadas pelo erro de quantização e a precisão é limitada [Nixon e Aguado (2019)].

²O ângulo do vetor gradiente corresponde ao ângulo do vetor normal à curva, que faz 90° com o vetor tangente à curva. Para efeito do cálculo de curvatura, que corresponde a uma diferença de ângulos, podemos considerar qualquer um dos dois, dado a sua relação fixa, diferindo por uma constante.

3.4.2 Medida de canto baseada em região

Os pontos de controle chamados também pontos de canto, pontos de interesse ou pontos de referência são utilizados em aplicações de registro, fusão ou rastreamento de imagens, porque são bons para calcular os parâmetros de correspondência para descobrir as transformações.

Um ponto de canto tem padrão de variação de brilho ou de cor em diferentes direções como na Figura 3.4a. Os cantos apresentam características únicas que os distinguem das áreas homogêneas (Figura 3.4b) e bordas com contraste em uma única direção (Figura 3.4c).

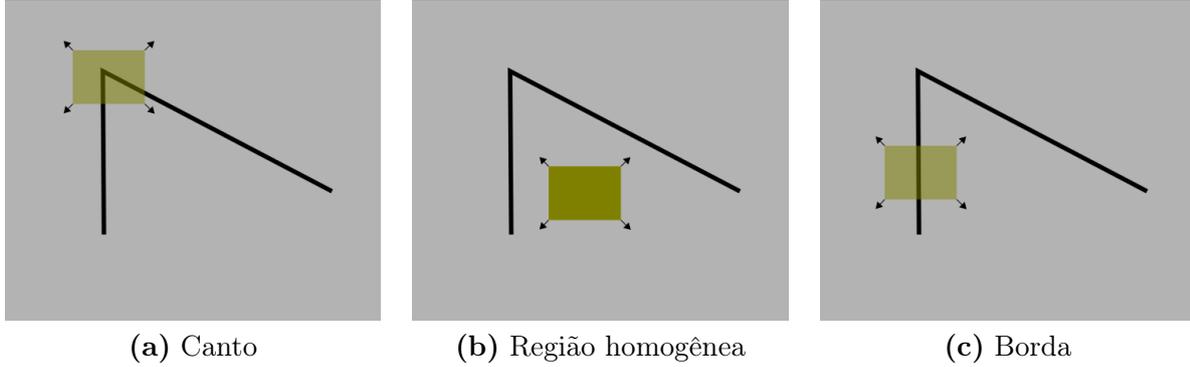


Figura 3.4: Padrões de variação de brilho ou de cor para diferentes localizações da janela: (a) Canto com mudança de contraste em duas direções (ponto de controle). (b) Região homogênea. (c) Borda.

Para detectar um ponto de canto e diferenciá-lo de uma borda ou uma região homogênea, uma abordagem inicial é dada pelo operador de Moravec [Moravec (1980)]. Esse operador calcula a variação de intensidade da imagem quando uma janela é deslocada em várias direções. A equação abaixo calcula a variação na direção (u, v) , utilizando uma janela de tamanho $(2w + 1) \times (2w + 1)$ com centro em (x, y) :

$$E_{u,v}(x, y) = \sum_{i=-w}^w \sum_{j=-w}^w [I_{(x+i,y+j)} - I_{(x+i+u,y+j+v)}]^2 \quad (3.15)$$

Uma medida de curvatura é dada pelo valor mínimo de $E_{u,v}(x, y)$ obtido considerando os deslocamentos (u, v) nas quatro direções principais, ou seja, para $(u, v) \in \{(1, 0), (0, -1), (0, 1), (-1, 0)\}$ (vizinhança-4).

O mínimo é escolhido porque concorda com as duas observações seguintes. Primeiro, se o pixel estiver em uma borda, então definindo uma linha reta, $E_{u,v}(x, y)$ é pequeno para um deslocamento ao longo da borda e grande para um deslocamento perpendicular à borda. Logo, o mínimo será pequeno. Em segundo lugar, se a borda definir um canto, todos os deslocamentos produzirão um valor alto. Assim, se escolhermos o mínimo, esse valor será elevado, indicando alta curvatura.

O principal problema dessa abordagem é que ela considera apenas um pequeno conjunto limitado de possíveis deslocamentos, o que gera um problema de precisão.

Uma solução melhorada é o método de Harris [Harris e Stephens (1988)] que utiliza a série de Taylor para representar a variação em um ponto da imagem (Equação 3.16):

$$I_{x+i+u,y+j+v} = I_{x+i,y+j} + \frac{\partial I_{x+i,y+j}}{\partial x} u + \frac{\partial I_{x+i,y+j}}{\partial y} v \quad (3.16)$$

Das Equações 3.15 e 3.16 é dedutível a seguinte Equação 3.17:

$$E_{u,v}(x, y) = \sum_{i=-w}^w \sum_{j=-w}^w \left(\frac{\partial I_{x+i,y+j}}{\partial x} u + \frac{\partial I_{x+i,y+j}}{\partial y} v \right)^2 \quad (3.17)$$

Finalmente, expandindo o termo quadrático e reorganizando na forma de matrizes, obtemos:

$$E_{u,v}(x, y) = A(x, y)u^2 + 2C(x, y)uv + B(x, y)v^2 = \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A(x, y) & C(x, y) \\ C(x, y) & B(x, y) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.18)$$

sendo que:

$$\begin{aligned} A(x, y) &= \sum_{i=-w}^w \sum_{j=-w}^w \left(\frac{\partial I_{x+i, y+j}}{\partial x} \right)^2 \\ B(x, y) &= \sum_{i=-w}^w \sum_{j=-w}^w \left(\frac{\partial I_{x+i, y+j}}{\partial y} \right)^2 \\ C(x, y) &= \sum_{i=-w}^w \sum_{j=-w}^w \left(\frac{\partial I_{x+i, y+j}}{\partial x} \right) \left(\frac{\partial I_{x+i, y+j}}{\partial y} \right) \\ M &= \begin{bmatrix} A(x, y) & C(x, y) \\ C(x, y) & B(x, y) \end{bmatrix} \end{aligned}$$

Dado que M é uma matriz diagonalizável, outra função $F_{u,v}(x, y)$ é obtida rotacionando $E_{u,v}(x, y)$ por uma transformação R que gira o eixo do sistema, para obter uma matriz diagonal Q :

$$F_{u,v}(x, y) = (RD)^T M (RD) = D^T R^T M R D = D^T Q D \quad (3.19)$$

sendo que R é uma matriz de rotação e

$$Q = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

tal que, λ_1 e λ_2 são autovalores, assim é possível deduzir que $M = RQR^T$ e $Q = R^T M R$.

Em uma região centrada em um ponto de canto, espera-se que os autovalores λ_1 e λ_2 possuam valores altos. Logo, uma boa medida de avaliação de canto é dada pelo menor autovalor [Tomasi e Kanade (1992)]: $\min\{\lambda_1, \lambda_2\}$.

Outras opções alternativas para evitar o cálculo explícito dos autovalores, são dadas na literatura com base nos valores do determinante e traço da matriz M :

1. Através do produto dos autovalores: $\det(M) = \det(Q) = \lambda_1 \lambda_2 = A(x, y)B(x, y) - C^2(x, y)$.
2. [Förstner e Gülch (1987)]: $\det(M)/\text{tr}(M)$, sendo $\text{tr}(M)$ o traço da matriz M .
3. [Harris e Stephens (1988)]: $\det(M) - k \cdot \text{tr}^2(M)$, sendo $k \in [0, 04; 0, 15]$.

A desvantagem destes métodos baseados em região é que na presença de duas ou mais bordas próximas, a identificação é comprometida, pois um canto pode ser identificado erroneamente, dado que autovalores elevados em mais de uma direção podem ser obtidos como consequência de múltiplas bordas passando pela janela, como ilustrado na Figura 3.5.

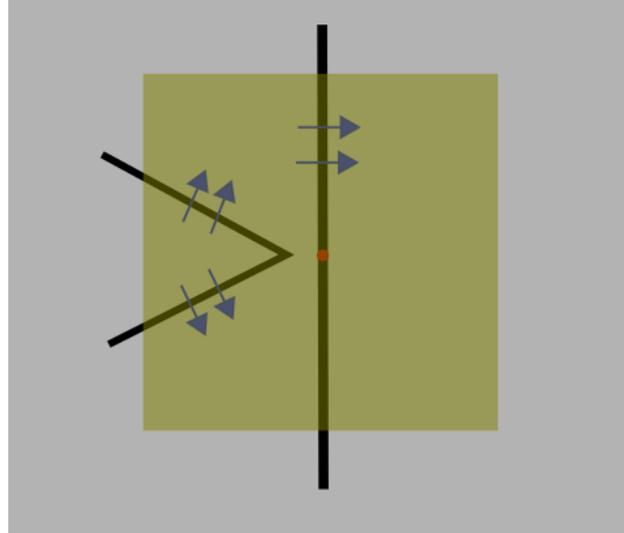


Figura 3.5: Exemplo de interferência por proximidade de múltiplas bordas, gerando mudanças de contraste em várias direções da janela.

3.4.3 Método de detecção de cantos por braços assimétricos

O método de [Guru e Dinesh \(2004\)](#) para detecção de cantos assume que um contorno é fornecido como uma curva fechada descrita no sentido horário, consistindo de uma sequência de pixels vizinhos, p_1, \dots, p_n . Para um dado ponto de interesse p_i , as sequências conexas de pontos em ambos os lados de p_i são chamados de braços do ponto. O braço esquerdo L_i e o braço direito R_i são, respectivamente, as sequências de pontos que descrevem os segmentos da curva, estendendo-se do ponto p_i no sentido anti-horário e no sentido horário:

$$L_i = \{p_j : p_j = (x_j, y_j) \in \mathbb{Z}^2, j = i - 1, i - 2, \dots, i - l\}$$

$$R_i = \{p_j : p_j = (x_j, y_j) \in \mathbb{Z}^2, j = i + 1, i + 2, \dots, i + r\}$$

onde l e r denotam, respectivamente, os tamanhos dos braços esquerdo e direito, que são decididos adaptativamente com base nas propriedades locais da curva e não são necessariamente iguais. Os pontos no braço esquerdo L_i , no braço direito R_i e no próprio ponto p_i juntos constituem a região de suporte desse ponto, sendo $l + r + 1$ o comprimento da região de suporte [[Guru e Dinesh \(2004\)](#); [Tsai et al. \(1999\)](#)]. Observe que a região de suporte não é necessariamente simétrica ($l \neq r$).

Para cada ponto de fronteira p_i , três características são calculadas:

1. o tamanho da região de suporte de p_i ,
2. a curvatura em p_i que é o recíproco do ângulo feito em p_i devido aos seus braços esquerdo e direito e
3. o valor limite, isto é, o número de pontos de fronteira para os quais o ponto p_i é um ponto final de sua região de suporte.

Com base nessas características um conjunto de regras é elaborado para orientar o processo de localização de cantos verdadeiros.

As propriedades estatísticas e geométricas associadas ao menor autovalor da matriz de covariância de uma sequência de pontos conexos são exploradas para determinação da região de suporte. Assim, em ambos os lados do ponto de interesse p_i , propõe-se escolher as sequências máximas de pontos para os respectivos braços, que resultem em valores aproximadamente nulos para o menor autovalor da matriz de covariância.

O menor autovalor λ_S da matriz de covariância de um conjunto S de pontos, pode ser calculado como:

$$\lambda_S = \frac{1}{2}(c_{11} + c_{22} - \sqrt{(c_{11} - c_{22})^2 + 4c_{12}^2}) \quad (3.20)$$

descrito pelos componentes da matriz de covariância C da sequência dos pontos conexos S de comprimento $|S|$:

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

cada componente é calculado a partir de cada pixel de S :

$$\begin{aligned} c_{11} &= \frac{1}{|S|} \sum_{(x_i, y_i) \in S} (x_i - \bar{x})^2 \\ c_{22} &= \frac{1}{|S|} \sum_{(x_i, y_i) \in S} (y_i - \bar{y})^2 \\ c_{21} = c_{12} &= \frac{1}{|S|} \sum_{(x_i, y_i) \in S} (x_i - \bar{x})(y_i - \bar{y}) \end{aligned}$$

sendo (\bar{x}, \bar{y}) o centroide geométrico:

$$\bar{x} = \frac{1}{|S|} \sum_{(x_i, y_i) \in S} x_i; \quad \bar{y} = \frac{1}{|S|} \sum_{(x_i, y_i) \in S} y_i$$

Para calcular o braço direito para o ponto de interesse p_i , inicialmente começamos com o conjunto $S_{i+1} = \{p_i, p_{i+1}\}$ e o menor autovalor da matriz de covariância de S_{i+1} denotado por $\lambda_{S_{i+1}}$ é calculado, que inicialmente é zero no caso de dois pontos. O conjunto S_{i+1} é então atualizado para $S_{i+2} = \{p_i, p_{i+1}, p_{i+2}\}$, adicionando o próximo ponto da sequência p_{i+2} , e o menor autovalor $\lambda_{S_{i+2}}$ da matriz de covariância atualizada de S_{i+2} é calculado. Esse processo de atualização do conjunto pela adição do próximo ponto na sequência, seguido pelo cálculo atualizado do menor autovalor associado, é repetido até que a condição $\lambda_{S_{i+r}} < \lambda_{S_{i+r+1}} < \lambda_{S_{i+r+2}}$ seja satisfeita para algum r . Uma vez satisfeita esta condição, o valor de r é tomado como o tamanho do braço direito e $R_i = S_{i+r}$ define o braço direito de p_i . Da mesma forma, o braço esquerdo L_i é determinado escolhendo pontos no sentido anti-horário até que $\lambda_{S_{i-l}} < \lambda_{S_{i-l-1}} < \lambda_{S_{i-l-2}}$ para algum l que é então considerado o tamanho do braço esquerdo de p_i .

Portanto, a região de suporte de p_i é uma sequência de pontos conexos $\{p_{i-l}, p_{i-l+1}, p_{i-l+2}, \dots, p_i, p_{i+1}, p_{i+2}, \dots, p_{i+r}\}$. Aqui, os pontos p_{i-l} e p_{i+r} são chamados de extremidades da região de suporte de p_i e seus valores limite são incrementados em uma unidade. Uma vez determinada a região de suporte de um ponto, a curvatura nesse ponto é então estimada como o recíproco do ângulo feito nesse ponto devido aos seus braços esquerdo e direito.

A existência de um máximo local em todas as três características (valor limite, tamanho da região de suporte e a curvatura) em um ponto é então usada como critério para indicar um verdadeiro ponto de canto.

A desvantagem desse método é que tem problema com a discretização e detecta cantos irrelevantes, sendo o critério de parada para definição dos braços um tanto específica (ad hoc), não apresentando boa generalização para casos mais desafiadores.

3.4.4 Detecção de cantos por áreas de influência

No trabalho de [Costa et al. \(2001\)](#) é realizado um crescimento de região com propagação de rótulos a partir dos pixels de contorno do objeto, seguindo uma transformada de distância euclídi-

ana, a fim de associar os elementos mais próximos para cada pixel do contorno. Este processo define duas áreas para cada pixel de contorno, sendo uma interna e outra externa. A área de influência de cada ponto de saliência (interna ou externa) refere-se ao ângulo de abertura, pela seguinte fórmula:

$$\text{Área} = \frac{\theta \times r^2}{2} \quad (3.21)$$

onde r é o raio de dilatação utilizado.

Espera-se que as áreas de influência dos pontos de maior curvatura, nomeadamente os pontos de saliência, sejam maiores do que as áreas de influência dos pontos restantes da forma.

Cada canto pode ser côncavo ou convexo, dependendo da relação entre suas áreas de influência interna e externa. Por exemplo, na Figura 3.6a, temos um ponto A de canto convexo, com uma área de influência externa (triângulo verde) maior do que sua área de influência interna (triângulo em azul), ao contrário do que acontece com o canto côncavo no ponto B.

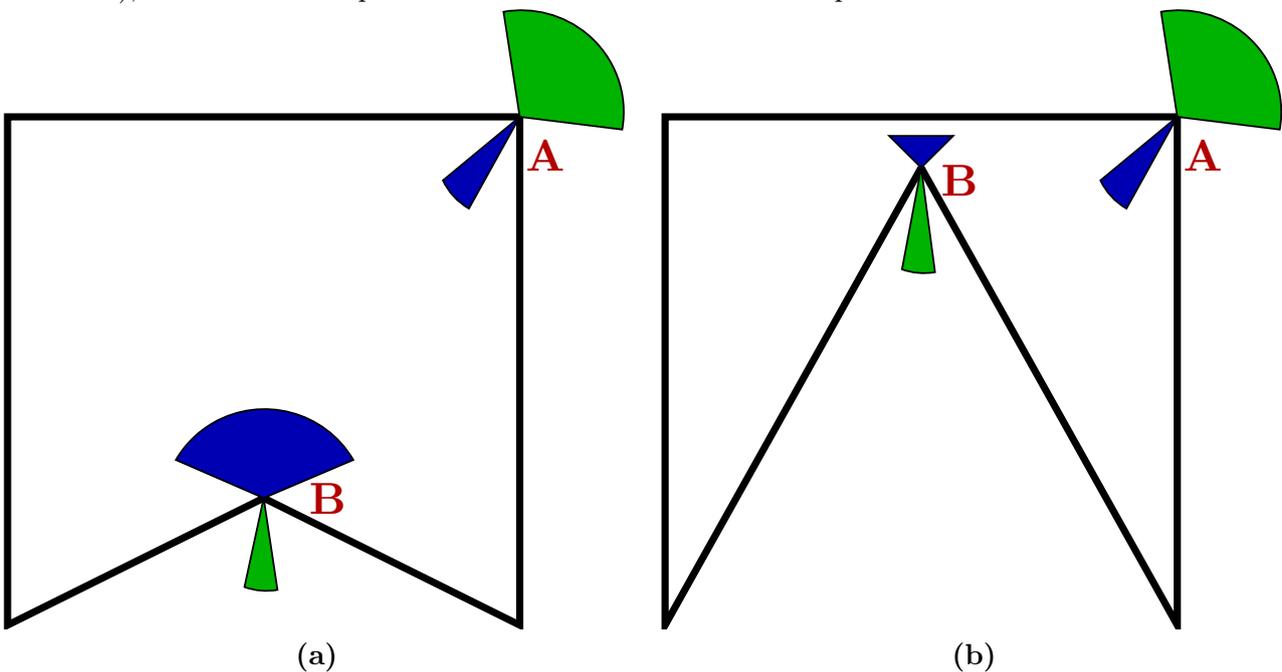


Figura 3.6: Exemplos de áreas de influência interna e externa para a caracterização de cantos: (a) Áreas de influência internas (azul) e externas (verde) dos pontos A (convexo) e B (côncavo). (b) Interferência no cálculo da área de influência interna de B no caso de contornos próximos.

No entanto, em contornos mais complexos pontos, importantes podem ser descartados, tal como quando um canto está próximo a uma segunda borda, tal como ilustrado na Figura 3.6b. Note que o canto B possui área de influência interna pequena (triângulo em azul) apesar de ser um canto com alta curvatura, devido a interferência gerada pela proximidade de uma outra parte da borda que restringe sua área de influência computada durante o crescimento de região.

A fim de corrigir o problema mencionado acima, no trabalho de Torres *et al.* (2004) são utilizados um esqueleto multiescala interno e outro externo para estimar os cantos (convexos e côncavos) de um objeto fornecido. No entanto, a atualização do esqueleto em um cenário de uma segmentação dinâmica se tornaria uma operação custosa.

3.4.5 Suavização de curva via minimização da energia Elástica

O *Elástica* é um modelo de regularização de curva que integra a curvatura ao quadrado além do comprimento da curva, que mostrou-se útil para suavização de contorno e interpolação, por exemplo, na presença de elementos finos [Antunes *et al.* (2021)].

O trabalho de Antunes *et al.* (2021) descreve um método de segmentação baseado no modelo

Elástica, no arcabouço de corte em grafo (*Graph Cut*), utilizando um coeficiente de balanceamento $\mu_r(D, p)$ para curvaturas acentuadas (Equação 3.22).

$$\mu_r(D, p) = \left(\frac{\pi r^2}{2} - \widehat{Area}(D(B_r(p)) \cap D) \right)^2 \quad (3.22)$$

sendo que D é o conjunto de pixels que compõem o objeto digitalizado, B_r é o conjunto de pixels que compõem o disco circular de raio r (isto é, o círculo centrado no ponto p pertencente ao contorno do objeto) e \widehat{Area} é a contagem de pixels no conjunto de interseção.

A ideia é analisar todos os pontos p a longo do contorno do objeto, tal que pontos cuja área de interseção $\widehat{Area}(B_r(p) \cap D)$ difere da metade da área de um círculo de mesmo raio são considerados como curvas acentuadas no contorno do objeto.

Por exemplo, considere os pontos **a**, **b**, **c** e **d** conforme ilustrados na Figura 3.7. Para o ponto **a**, temos que o círculo $B_r(\mathbf{a})$ apresenta interseção com o objeto com área próxima de sua metade, logo caracterizando uma parte do contorno de baixa curvatura (isto é, $\mu_r(D, \mathbf{a}) \approx 0$). Já para o ponto **b**, temos uma curvatura acentuada do tipo côncava, que é indicada por uma elevada área de interseção, resultando em valores elevados de $\mu_r(D, \mathbf{b})$. Por outro lado, pontos de curvatura acentuada do tipo convexa, como o ponto **c**, apresentam baixa área de interseção, o que resulta em um valor elevado para a diferença em relação a área média esperada e consequentemente um valor elevado para $\mu_r(D, \mathbf{c})$. No entanto, esta técnica de estimativa de curvatura apresenta problemas no caso de formas mais complexas, quando partes da borda quase se tocam, como por exemplo no ponto **d** da Figura 3.7, em que a área de interseção se torna elevada, falsamente indicando a presença de uma curvatura do tipo côncava.

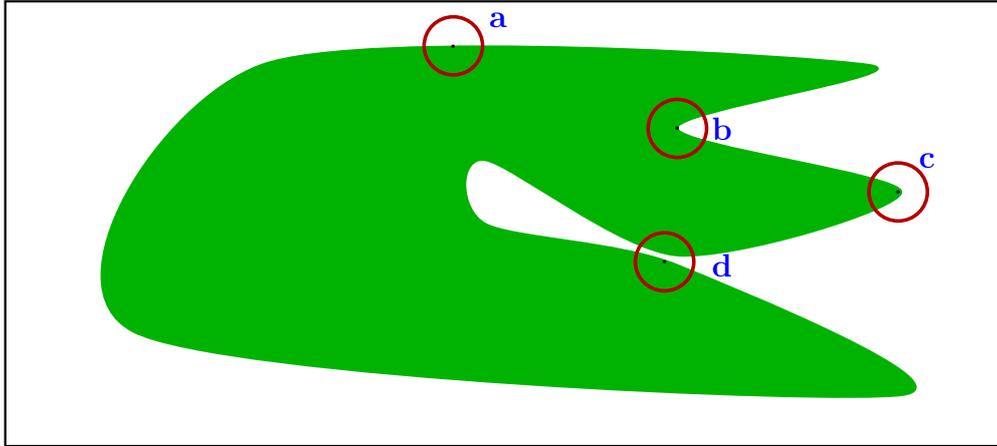


Figura 3.7: Exemplo de estimativa de curvatura utilizando discos circulares em pontos amostrados ao longo do contorno do objeto: Ausência de curvatura no ponto **a** quando a área de interseção é aproximadamente metade do círculo. Curvatura côncava no ponto **b** quando existe maior área de interseção. Curvatura convexa no ponto **c** quando existe menor área de interseção. Região problemática no ponto **d** com maior área de interseção, falsamente indicando uma curvatura côncava.

3.4.6 k -curvatura

A fim de evitar problemas na estimativa de curvatura devido a ruídos presentes na borda e efeitos da discretização do domínio da imagem, a curvatura pode ser melhor estimada com base em um valor de escala k , gerando uma medida conhecida como k -curvatura. Isto pode ser realizado pelo método de k -inclinação (*k-slope*). A k -inclinação em um ponto do contorno (p_i) é definida como a inclinação de uma linha conectando esse ponto com seu k -ésimo vizinho à direita (p_{i+k}). Em seguida, a k -curvatura em um ponto do contorno é definida como a diferença entre a k -inclinação naquele pixel e a k -inclinação de seu k -ésimo vizinho à esquerda (p_{i-k}) [Arica e Yarman Vural (2003)]. Portanto, a k -curvatura é definida como a variação de inclinação ($\theta' - \theta$) ao longo de um trecho do

contorno composto por $2 \times k + 1$ pixels (comprimento da região de suporte). Ou seja, temos um caso particular do método descrito na Seção 3.4.3, utilizando braços simétricos, sem ter que entrar em todas as dificuldades de se estimar o tamanho dos braços, pois k passa a ser um parâmetro fixo de escala (Figura 3.8).

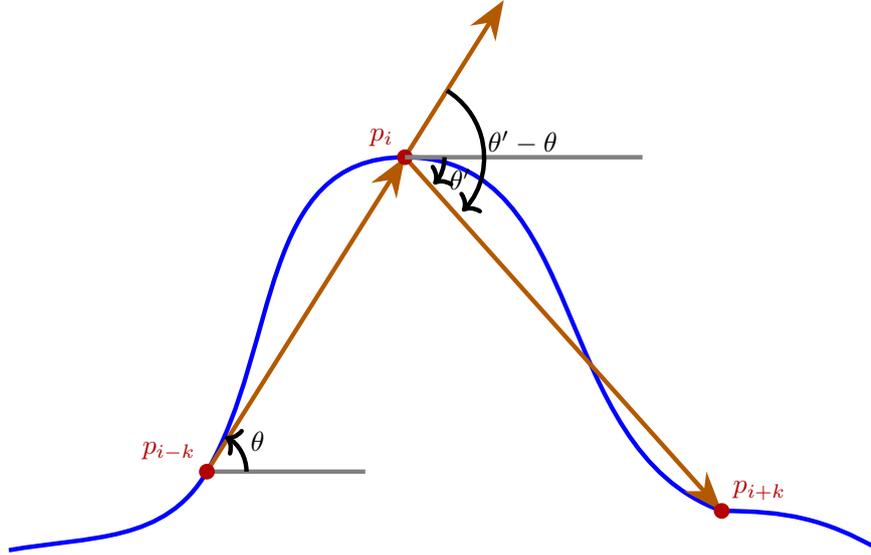


Figura 3.8: Medida de k -curvatura via variação de k -inclinação ao longo de um trecho do contorno.

Uma formulação alternativa para o conceito de k -curvatura, via a norma de um vetor resultante, pode ser obtida pela Equação 3.23, que se baseia no termo de curvatura da energia interna do método *G-Wire* [Kang (2005)], porém estendida para uma escala arbitrária (valor de k), tal como empregado pelo método *Bandeirantes* [Condori *et al.* (2017)], e normalizada para gerar valores no intervalo fechado de 0 a 1.

$$k\text{-curvatura} = \frac{\|p_{i-k} - p_i + p_{i+k} - p_i\|}{\|p_{i-k} - p_i\| + \|p_{i+k} - p_i\|} \quad (3.23)$$

Por exemplo, considere a curva apresentada na Figura 3.9a. O vetor resultante $\vec{r} = \vec{a} + \vec{b}$, tal que $\vec{a} = p_{i-k} - p_i$ e $\vec{b} = p_{i+k} - p_i$, é mostrado em cor verde. Observe que quanto maior for a curvatura, maior será a magnitude do vetor resultante. Ao passo que para regiões de baixa curvatura, os vetores \vec{a} e \vec{b} se anulam, gerando um vetor resultante praticamente nulo (Figura 3.9b). A vantagem do uso do conceito de k -curvatura é a sua maior robustez em relação a presença de ruídos no contorno, conforme ilustrado na Figura 3.9c, em que uma baixa curvatura é estimada apesar dos ruídos existentes.

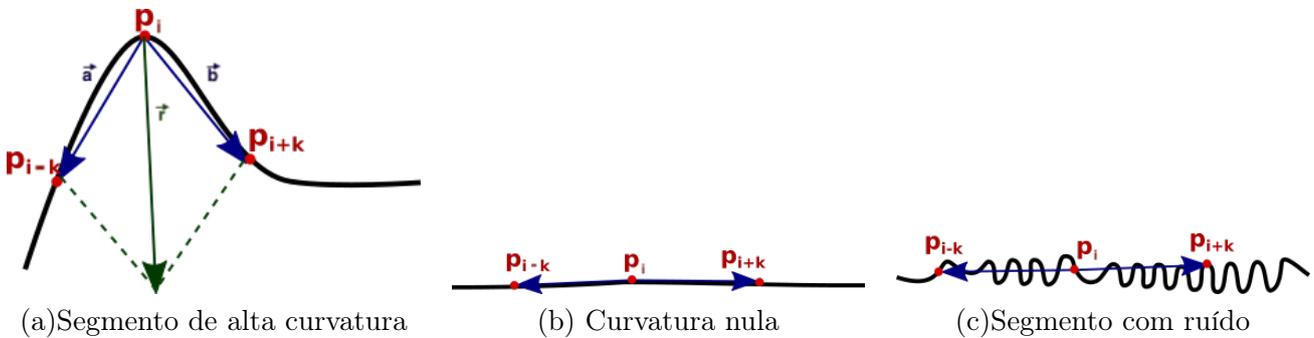


Figura 3.9: Medida de k -curvatura via norma de vetor resultante em diferentes situações.

Neste trabalho, adotamos uma medida de k -curvatura baseada na Equação 3.23, em função de sua maior versatilidade de escala e robustez a ruídos. Porém, por se tratar de um método de estimativa de curvatura baseado em segmento de curva (caminho), o que pressupõe uma pré-

segmentação de segmento de borda, adotamos uma série de delineamentos locais pela técnica do *Live Wire* (LW) [Falcão *et al.* (2000)] de maneira integrada na solução proposta, conforme será explicado em detalhes no próximo capítulo. Porém, cabe adiantar aqui que, a fim de amenizar o problema do uso de um valor fixo de escala para k , um intervalo de valores é utilizado na estimativa, resultando em um valor médio de curvatura que é estatisticamente mais robusto.

Capítulo 4

Segmentação pela Transformada Imagem-Floresta Orientada com Características Locais de Curvatura

Em segmentação de imagens baseada em grafos, incluindo os problemas de minimização das energias ε_1 e ε_∞ , os pesos dos arcos são dados por uma função indicadora de aresta local (por exemplo: magnitude média do gradiente) segundo os atributos da imagem e a informação prévia do objeto [Ciesielski e Udupa (2010a,b); Miranda *et al.* (2010)].

Em métodos de perseguição de bordas (ver Seção 3.2), um processo de integração de arestas, geralmente baseado em uma função de custo do caminho, combina arestas locais em curvas de borda longas e significativas, interligando um conjunto de pontos de ancoragem, de modo que um contorno fechado seja calculado para a segmentação [Falcão *et al.* (2000); Kang (2005)].

A segmentação com sementes baseada em regiões e métodos de perseguição de bordas são geralmente apresentados como diferentes abordagens concorrentes, sendo a primeira mais fácil de se entender para imagens multidimensionais, enquanto a última é mais sensível à erros de posicionamento das sementes (pontos de ancoragem) [Boykov e Funka-Lea (2006)]. Neste capítulo, mostramos que múltiplas integrações de borda de curto alcance podem ser usadas para extrair características de curvatura por toda a imagem a fim de melhorar a segmentação com sementes baseada em região.

Nosso método proposto difere do trabalho de Yang *et al.* (2010) e de abordagens híbridas, tais como o paradigma *Live Markers* [Vallin Spina *et al.* (2014)], porque o nosso método não requer a especificação, por parte do usuário, de restrições de borda (pontos de ancoragem ou outras restrições suaves baseadas em pontos de borda [Jones *et al.* (2016)]), enquanto que o *Live Markers* requer a seleção de pontos de ancoragem sobre a borda do objeto para calcular segmentos ótimos de borda, os quais são transformados em marcadores internos e externos para o delineamento final baseado em região (Figura 4.1).

Em nossa abordagem, o método de perseguição de bordas é aplicado localmente sobre regiões circulares dentro da imagem para extrair características curvilíneas sem a necessidade de qualquer intervenção do usuário. Como resultado, temos uma melhor atribuição de peso de arco, migrando de uma simples função indicadora de aresta local para uma função de integração de aresta de curto alcance mais geral. Como vantagens temos:

- Segmentos por perseguição de borda local com polaridade de borda inconsistente podem ser penalizados.
- Podemos favorecer a segmentação de objetos com formas mais regulares e suaves penalizando arcos em segmentos com curvatura acentuada.
- Podemos preencher lacunas da borda nos pontos com fraca indicação de aresta para evitar os problemas de vazamento na segmentação.

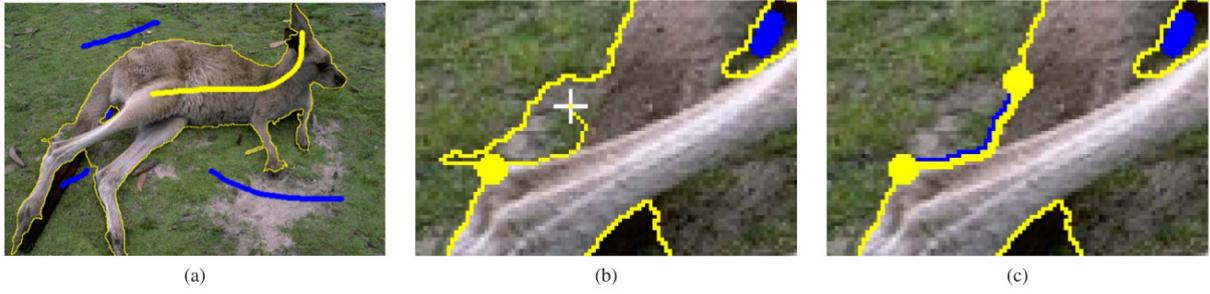


Figura 4.1: Segmentação pela IFT via o método de Live Markers: (a) Segmentação inicial usando restrições de região (marcadores) por IFT. (b) Ativação do Live Wire com um ponto de ancoragem e seu caminho ótimo calculado até a posição atual do cursor (cruz branca). (c) Marcadores por Live Markers, com sementes internas e externas a partir do segmento de fronteira por Live Wire e atualização do delineamento final de região por IFT.

- Mantemos a interface de usuário simples de usar sem a necessidade de múltiplos tipos de entrada do usuário, como acontece no *Live Markers*.

A integração da aresta de curto alcance ajuda a contornar os principais problemas da família da energia ε_∞ (bordas irregulares e “vazamento”). Os resultados são demonstrados utilizando integração de aresta por *Live Wire* (LW) [Falcão *et al.* (2000)], combinada com a OIFT [Mansilla e Miranda (2013a); Miranda e Mansilla (2014)], devido as suas forças complementares, dado que LW pode ser visto como uma versão por perseguição de bordas do problema de minimização da energia ε_1 em um grafo dual [Boykov e Funka-Lea (2006); Miranda *et al.* (2012)]. Como resultado, temos uma segmentação globalmente ótima por OIFT, que pode ser adaptada para um determinado objeto alvo, de acordo com suas características curvilíneas locais e outras restrições de alto nível já suportadas pela OIFT [Condori e Miranda (2022); de Moraes Braz e Miranda (2014); de Moraes Braz *et al.* (2020); Leon *et al.* (2021b); Mansilla e Miranda (2016); Mansilla *et al.* (2016); Mansilla e Miranda (2013b)].

4.1 Transformada Imagem-Floresta Orientada com Características Locais de Curvatura (OIFT-LCF)

No nosso primeiro método proposto, denominado OIFT com Características Locais de Curvatura (OIFT-LCF - *OIFT with Localized Curvilinear Features*) [Bejar *et al.* (2017)], o método *Live Wire* de perseguição de bordas é aplicado localmente sobre a imagem dentro de regiões circulares para extrair características de curvatura sem a necessidade de nenhuma intervenção do usuário. Para cada pixel c da imagem, consideramos um disco circular $\mathcal{D}(c) = \{t \in \mathbb{Z}^n : \|t - c\| \leq R\}$ de raio R com centro em c . O caminho ótimo $\pi_{a \rightsquigarrow c}$ de um pixel a na borda do disco $B(c) = \{s \in \mathcal{D}(c) : \exists t \notin \mathcal{D}(c) \text{ tal que } \|s - t\| \leq 1\}$ para o pixel central c é calculado pelo Algoritmo 1 com a função *Live Wire* f_{LW} (página 31), utilizando $\mathcal{S} = B(c)$ e a função indicadora de aresta local $\omega(\langle s, t \rangle)$ definida por:

$$\omega(\langle s, t \rangle) = \begin{cases} \left(\frac{\bar{G}(s) + \bar{G}(t)}{2} \times (1 + \gamma) \right)^\beta + \|s - t\| & \text{se } I(r) > I(l) \\ \left(\frac{\bar{G}(s) + \bar{G}(t)}{2} \times (1 - \gamma) \right)^\beta + \|s - t\| & \text{se } I(r) < I(l) \\ \left(\frac{\bar{G}(s) + \bar{G}(t)}{2} \right)^\beta + \|s - t\| & \text{caso contrário} \end{cases} \quad (4.1)$$

tal que $\bar{G}(t)$ é o complemento da magnitude do gradiente $G(t)$ no pixel t , l e r são os pixels vizinhos à esquerda e direita do arco $\langle s, t \rangle$ (Figura 4.2) e geralmente usamos $\gamma = 50\%$.

O parâmetro γ é utilizado para penalizar segmentos com polaridade de borda inconsistente,

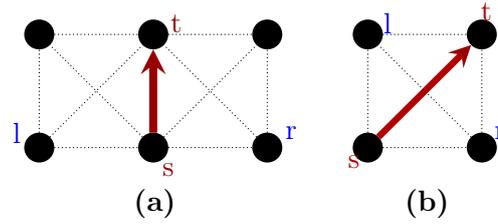


Figura 4.2: Os pixels vizinhos à esquerda e à direita de cada arco $\langle s, t \rangle$, considerando um grafo de vizinhança-8.

favorecendo uma orientação de borda particular. Por exemplo, $\gamma = 50\%$ melhora arcos com o seu pixel direito sendo mais escuro do que o seu pixel esquerdo (Figura 4.3).

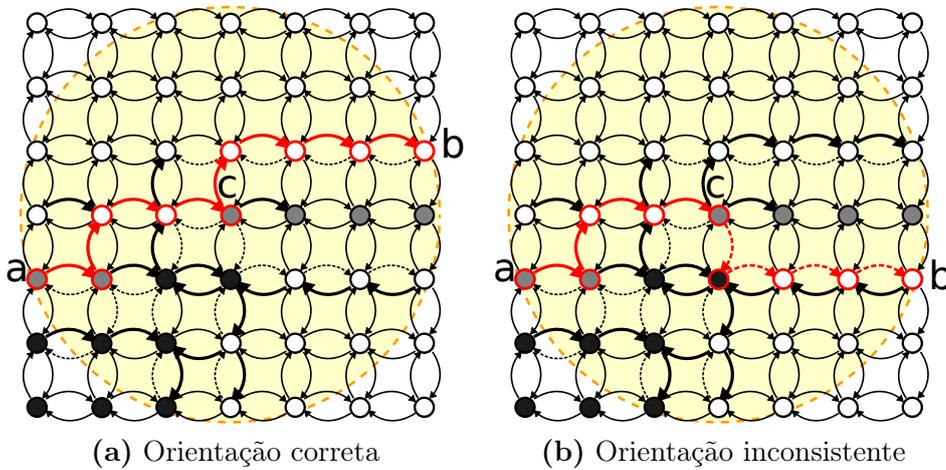


Figura 4.3: Perseguição de borda com polaridade de borda em um grafo de vizinhança-4. Os arcos com a orientação desejada (pixels a direita mais escuros que seus pixels a esquerda) são apresentados com linhas mais grossas, enquanto arcos com a orientação incorreta são apresentados com linhas tracejadas. (a) Segmentos por Live Wire com a orientação correta estão destacados em vermelho. (b) Segmentos com polaridade inconsistente são mostrados em vermelho. Observe que o trecho do pixel a para c apresenta polaridade com pixels claros à esquerda e pixels escuros à direita, ao passo que o trecho de c para b apresenta polaridade oposta.

O caminho ótimo de c até um pixel b na borda do disco $B(c)$, é calculado, evitando passar sobre o caminho anterior $\pi_{a \rightsquigarrow c}$, tal que terminamos com um caminho composto $\pi_{a \rightsquigarrow b} = \pi_{a \rightsquigarrow c} \cdot \pi_{c \rightsquigarrow b} = \langle p_1 = a, p_2, \dots, p_k = c, \dots, p_m = b \rangle$, conectando dois pontos da borda do disco e passando pelo seu centro. Este caminho composto computado passando por c no centro de um disco de raio R será denotado por $LW(c, R)$. Nesta integração de borda de curto alcance, os segmentos de *Live Wire* aderem as bordas dos objetos, de tal modo que podemos extrair informações importantes de contorno dos objetos subjacentes por toda a imagem (Figura 4.4a,d).

Para cada pixel c , podemos extrair características curvilíneas do caminho composto $\pi_{a \rightsquigarrow c} \cdot \pi_{c \rightsquigarrow b}$, tal como a seguinte medida de curvatura média, com base no conceito de k -curvatura apresentado previamente na Equação 3.23:

$$Curv(LW(c, R)) = Curv(\langle p_1, \dots, p_m \rangle) = \frac{1}{(L/2)} \cdot \sum_{i=L/2}^{L-1} \frac{\|p_{k-i} - 2 \cdot c + p_{k+i}\|}{\|p_{k-i} - c\| + \|c - p_{k+i}\|} \quad (4.2)$$

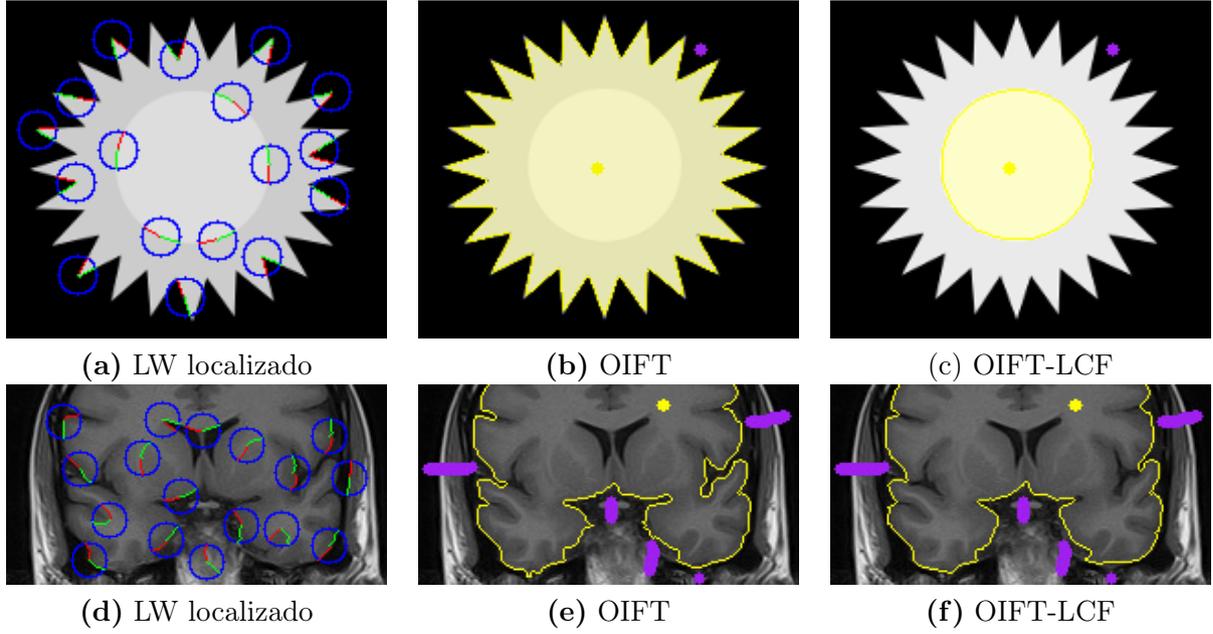


Figura 4.4: *Extração de informação de contorno em regiões circulares no domínio da imagem: (a) Características de curvatura podem ser extraídas pelo cálculo dos segmentos locais por Live Wire. (b-c) A segmentação de objetos com formas mais regulares e suavizadas pode ser favorecida, penalizando arcos em segmentos com curvatura mais acentuada. (d-f) Um exemplo usando imagens de ressonância magnética do cérebro.*

em que $L = \min\{\text{Length}(\pi_{a \rightsquigarrow c}), \text{Length}(\pi_{c \rightsquigarrow b})\} + 1 = \min\{k - 1, m - k\} + 1$.

Podemos então melhorar a atribuição de peso de arco para a OIFT (na Equação 3.11 página 32), migrando de uma função indicadora da aresta local $\delta(\langle s, t \rangle) = \frac{G(s)+G(t)}{2}$ para uma função mais geral de integração de aresta de curto alcance $\delta(\langle s, t \rangle) = \frac{G_{LCF}(s)+G_{LCF}(t)}{2}$, na qual, para todo pixel $c \in \mathcal{I}$, temos:

$$\bar{G}_{LCF}(c) = \bar{G}(c) \times (1 + \text{Curv}^2(\pi_{a \rightsquigarrow c} \cdot \pi_{c \rightsquigarrow b})) \quad (4.3)$$

Nesta abordagem, a OIFT utiliza sua forma dual equivalente como um problema de maximização da energia $\bar{\varepsilon}_{\infty}(\mathcal{O}) = \min_{\langle s, t \rangle \in \mathcal{C}(\mathcal{O})} \omega(\langle s, t \rangle)$, em que a função de peso $\omega(\langle s, t \rangle)$ da Equação 3.11 (página 32) é combinada com $\delta(\langle s, t \rangle)$, tal como definido acima.

A OIFT com Características Locais de Curvatura (OIFT-LCF) ameniza os problemas recorrentes de bordas irregulares da OIFT original (Figuras 4.4 e 4.5). Regularidade de curvatura para segmentação de imagens baseada em região geralmente resulta em um problema NP-difícil, sendo a relaxação por programação linear com limiarização usada para obter apenas uma solução aproximada [Schoenemann *et al.* (2009)]. No nosso método (OIFT-LCF), as características de curvatura podem ser calculadas rapidamente usando execuções locais de *Live Wire*, que podem ser calculadas em paralelo para diferentes regiões da imagem. Dado que ambos, *Live Wire* e OIFT, possuem complexidade linear em cenários de pior caso, usando uma estrutura de **heap** binário para a fila de prioridades \mathcal{Q} no Algoritmo 1, e o *Live Wire* é calculado somente em discos circulares de tamanho fixo constante, a complexidade da OIFT-LCF é linear.

Outra possibilidade é considerar a seguinte formulação alternativa abaixo na OIFT-LCF, ao invés de usar a Equação 4.3:

$$\bar{G}_{LCF}(c) = \left(\frac{f_{LW}(\pi_{a \rightsquigarrow c}) + f_{LW}(\pi_{c \rightsquigarrow b})}{\text{Length}(\pi_{a \rightsquigarrow c} \cdot \pi_{c \rightsquigarrow b})} \right)^{1/\beta} \times (1 + \text{Curv}^2(\pi_{a \rightsquigarrow c} \cdot \pi_{c \rightsquigarrow b})) \quad (4.4)$$

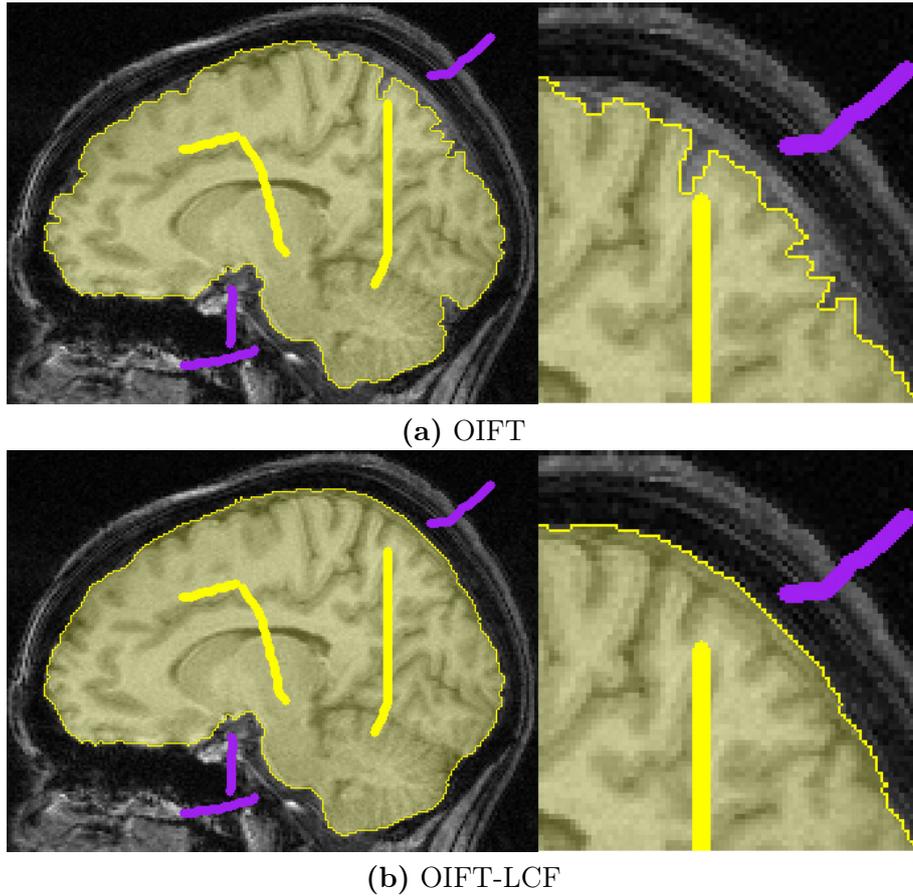


Figura 4.5: A segmentação da superfície externa do cérebro em uma imagem RM-T1. OIFT-LCF gera um contorno mais regular, seguindo de modo mais preciso a dura-máter, devido a sua análise de curvatura.

o que ajuda a contornar os problemas de “vazamento” da OIFT original (Figura 4.6). Observe que o fator $(1 + Curv^2(\pi_{a \rightsquigarrow c} \cdot \pi_{c \rightsquigarrow b}))$ corresponde a uma penalização de curvatura, tal como utilizada na formulação anterior (Equação 4.3), porém a parte $\frac{f_{LW}(\pi_{a \rightsquigarrow c}) + f_{LW}(\pi_{c \rightsquigarrow b})}{Length(\pi_{a \rightsquigarrow c} \cdot \pi_{c \rightsquigarrow b})}$ corresponde ao custo médio da função de conexidade do *Live Wire*, sendo f_{LW} dada pela Equação 3.6, que leva em conta os custos dos caminhos $\pi_{a \rightsquigarrow c}$ e $\pi_{c \rightsquigarrow b}$, ao invés de somente o valor do gradiente em c , como na abordagem anterior (Equação 4.3). Deste modo, em pixels da borda com baixo contraste (isto é, fraca resposta do filtro gradiente), ainda seria possível obter informações de partes fortes da borda nas imediações, a fim de fechar/preencher a falha.

4.2 Resultados Experimentais

Foram realizados experimentos quantitativos, usando um total de 40 fatias de tomografia computadorizada (TC) provenientes de 10 estudos torácicos para segmentar o fígado e 40 fatias de ressonância magnética (RM) reais do pé para segmentar o osso calcâneo. Diversos tipos diferentes de gradientes, com ou sem o uso de características curvilíneas localizadas, foram avaliados utilizando a curva média de acurácia pelo coeficiente de Dice para diferentes conjuntos de sementes e os dados de gabarito obtidos de um especialista do departamento de radiologia da Universidade da Pensilvânia.

No primeiro experimento, foi utilizada a segunda versão da OIFT-LCF, com base na Equação 4.4, para evitar problemas de “vazamento”, com $R = 3$ e $\beta = 2$, e consideramos diferentes conjuntos de sementes obtidos automaticamente por erosão e dilatação do gabarito para diferentes valores do raio. Variando o valor do raio, podemos calcular a segmentação para diferentes conjuntos de

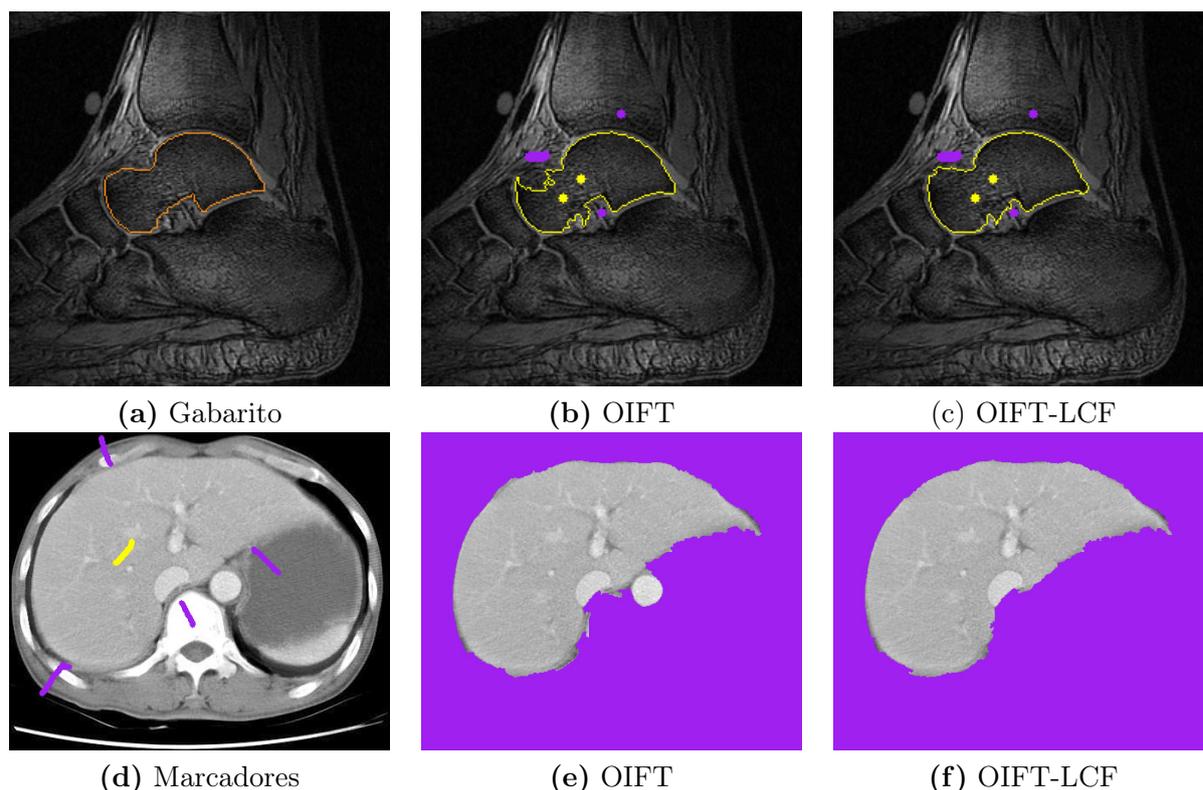


Figura 4.6: A segmentação de: (a-c) Um osso do pé humano em uma imagem de RM e (d-f) o fígado em uma imagem de TC.

sementes e traçar curvas de acurácia usando o coeficiente de similaridade de Dice. Porém, a fim de gerar um cenário mais desafiador, consideramos raios de dilatação maiores para as sementes externas (duas vezes o valor do raio interno), resultando em uma disposição assimétrica das sementes. Para demonstrar a versatilidade da OIFT-LCF e OIFT, que suportam diversas restrições de alto nível, foram considerados esses métodos com restrições de forma pela Convexidade Geodésica em Estrela (GSC - *Geodesic Star Convexity*) [Mansilla e Miranda (2013b)], levando aos métodos GSC-OIFT-LCF e GSC-OIFT.

As Figuras 4.8a-b apresentam as curvas médias de acurácia para todas as imagens do primeiro experimento, usando diferentes gradientes de entrada indicados por um subíndice para cada método. Por questões de simplicidade, consideramos gradientes a partir de atributos locais da imagem sem recorrer a técnicas mais sofisticadas por meio de aprendizagem supervisionada [Wolf *et al.* (2017)]. A seguir, são apresentados os cinco métodos considerados para cada um dos subíndices:

Subíndice 1: Este método é usado para indicar $G(t)$ como a magnitude do gradiente de Sobel.

Subíndice 2: Denota os pesos por atributo de imagem segundo Miranda *et al.* (2010), que são baseados em suavização em quatro diferentes escalas por uma sequência de aberturas por reconstrução e fechamentos por reconstrução.

Subíndice 3: Corresponde a um gradiente morfológico com raio de 1.5 pixels, ou seja, a diferença entre a dilatação e a erosão da imagem.

Subíndice 4: Descreve as características de gradiente de brilho do trabalho de Martin *et al.* (2004), em que, para cada pixel, um círculo de raio r é desenhado e dividido ao longo do seu diâmetro em uma orientação θ . As regiões das metades do disco são descritas por histogramas, que são comparados pela distância chi-quadrado. Uma diferença acentuada entre as metades do disco indica uma descontinuidade na imagem ao longo do diâmetro do disco. Os

histogramas (com 12 bins) são calculados por um *kernel* de estimação de densidade usando um *kernel* Gaussiano com $\sigma = 10$. Foram considerados $r = 4$ e quatro diferentes orientações (Figura 4.7).

Subíndice 5: Este gradiente foi inspirado pelo trabalho de Rauber *et al.* (2013), no qual grafos de *superpixels* demonstraram melhorar a segmentação interativa, explorando a cor/intensidade média dentro dos *superpixels*. Tentamos reproduzir resultados similares no nível de pixel, usando a média de brilho dentro dos *superpixels* de tamanho 5×5 pixels calculados pela IFT-SLIC [Alexandre *et al.* (2015)] como imagem de entrada para o operador Sobel.

Dos resultados (Figuras 4.8a-b) é evidente que GSC-OIFT-LCF supera GSC-OIFT para todos os índices correspondentes (isto é, GSC-OIFT-LCF^{*i*} é melhor do que GSC-OIFT^{*i*}, para $i = 1, \dots, 5$).

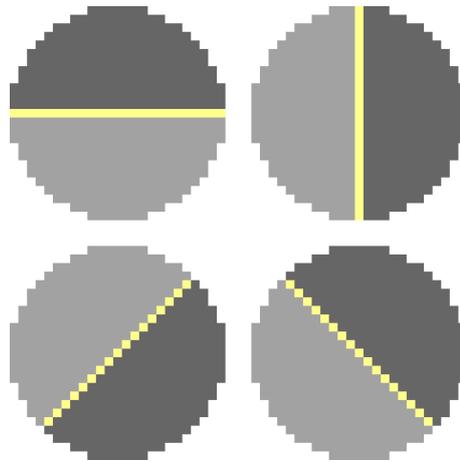


Figura 4.7: As quatro diferentes orientações consideradas para o cálculo do gradiente de brilho do trabalho de Martin *et al.* (2004).

No segundo experimento, foi usada a primeira versão do OIFT-LCF, com base na Equação 4.3, para tratar o problema de bordas irregulares, com $R = 4$ e $\beta = 5$. A Figura 4.8c apresenta a curva experimental usando um usuário robô [Gulshan *et al.* (2010)], confirmando resultados similares aos obtidos no primeiro experimento.

4.3 Considerações parciais

Em conclusão, desenvolvemos neste capítulo uma extensão, intitulada OIFT-LCF, para o algoritmo da OIFT, incorporando restrições curvilíneas localizadas em sua formulação, ajudando a contornar contornos irregulares da OIFT. Dado que o método proposto obtém bordas suavizadas, ao preço de múltiplas execuções locais do *Live Wire* (que é baseado no cálculo da IFT), temos portanto uma indicação favorável à primeira hipótese (Hipótese 1) levantada na Seção 1.2. Isto é, se múltiplas execuções da IFT são empregadas, é possível obter uma segmentação baseada em região com contornos suaves a um baixo custo computacional. O baixo custo computacional se deve a complexidade linearítmica da OIFT-LCF, uma vez que o *Live Wire* é calculado somente em discos circulares de tamanho fixo constante, tal como discutido anteriormente, sendo ainda que as múltiplas execuções do *Live Wire* podem ser calculadas em paralelo.

No próximo capítulo, estudamos a viabilidade da Hipótese 2, que corresponde a uma extensão da OIFT para o paradigma de segmentação não supervisionada.

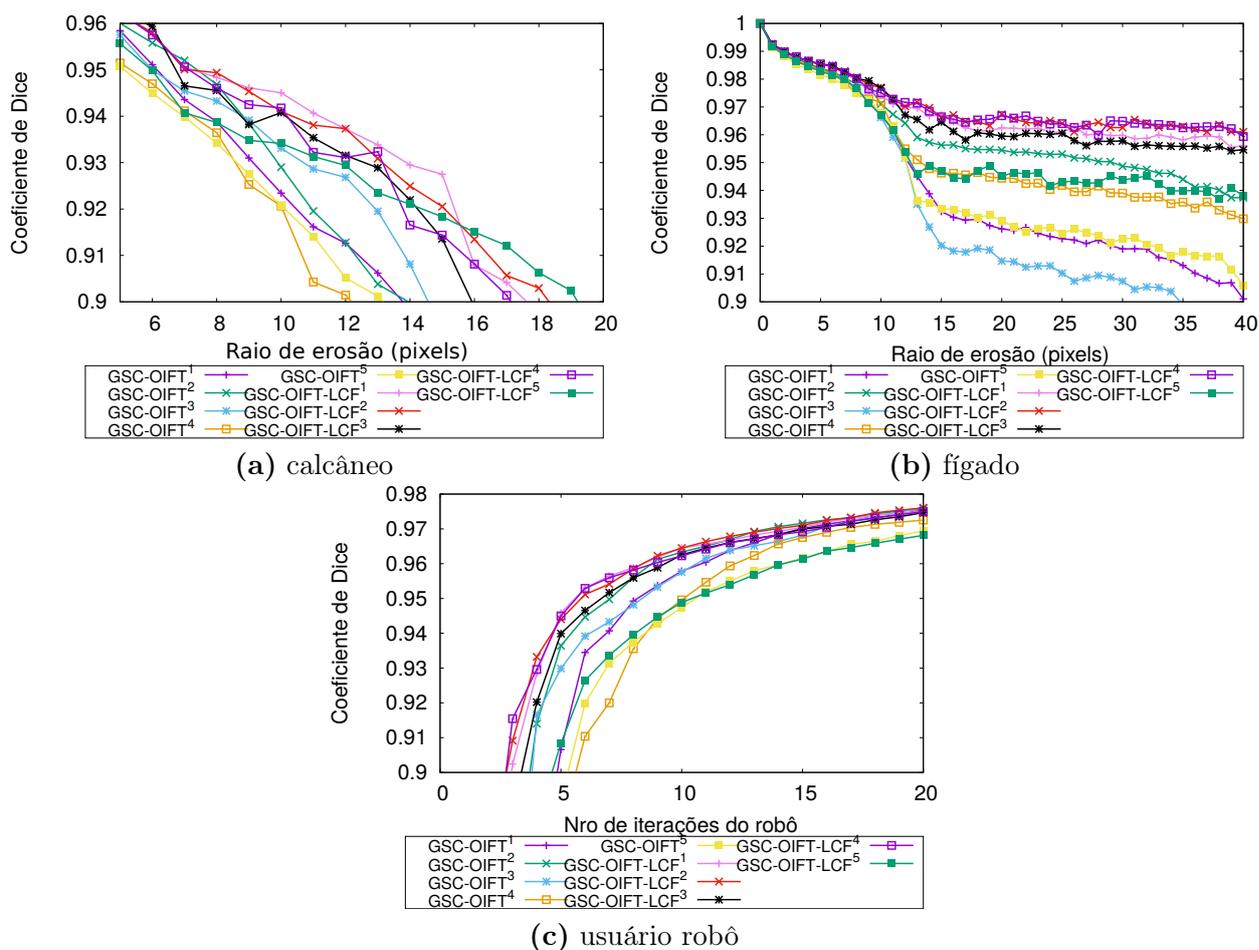


Figura 4.8: As curvas de acurácia média (Dice) usando diferentes gradientes com e sem o uso das características de curvatura local: (a-b) Primeiro experimento para as segmentações do (a) osso calcâneo e (b) fígado. (c) Curva de acurácia média do calcâneo por usuário robô.

Capítulo 5

Segmentação Não Supervisionada de Imagens por Cortes Ótimos em Grafos

Segmentação não supervisionada corresponde a um importante problema em visão computacional, visto que o agrupamento perceptivo desempenha um papel poderoso na percepção visual humana [Wu e Leahy (1993)]. Neste contexto, o método deve decidir quais regiões da imagem são relevantes sem a orientação do usuário, com base na similaridade de cor e textura ou contraste local.

Conforme discutido anteriormente na Seção 2.7.2, regiões compactas e conexas de pixels semelhantes, obtidas por um método de super-segmentação não supervisionada, são comumente chamadas de *superpixels* [Achanta *et al.* (2012); Vargas-Muñoz *et al.* (2018)]. Este particionamento pode reduzir bastante o tempo computacional de algoritmos de visão computacional, substituindo a estrutura rígida de uma grade de pixels [Achanta *et al.* (2012)]. No contexto de métodos baseados em grafos, ela permite a rápida criação de um Grafo de Regiões Adjacentes (RAG - *Region Adjacency Graph*), reduzindo drasticamente o número de elementos do grafo comparado ao grafo no nível de pixels (Figuras 5.1a-b).

Vários métodos baseados em grafos têm sido propostos para segmentação não-supervisionada, incluindo Bacias Hidrográficas (*Watershed*) [Audigier e Lotufo (2007); Cousty *et al.* (2007); Lotufo *et al.* (2002)], Corte Médio Mínimo (*Minimum Mean Cut*) [Wang e Siskind (2003); Wang e Siskind (2001)], Corte Normalizado (*Normalized Cut*) [Carballido-Gamio *et al.* (2004); Shi e Malik (2000)] e métodos baseados na Árvore Geradora Mínima (MST - *Minimum Spanning Tree*) [Cousty *et al.* (2018); Feng *et al.* (2011); Guimarães *et al.* (2017); Guimarães *et al.* (2012); Zhang e Alhajj (2006)].

Por exemplo, Felzenszwalb e Huttenlocher (2004) propuseram um algoritmo de segmentação eficiente que avalia um predicado para medir a evidência de borda entre duas regiões, que produz segmentações satisfazendo propriedades globais, embora com base em decisões gulosas [Felzenszwalb e Huttenlocher (2004)]. Outros métodos incluem o uso de árvores de componentes [Silva e Lotufo (2011); Souza *et al.* (2017)], que podem ser combinadas com a abordagem das Bacias Hidrográficas, permitindo a seleção de bacias de acordo com seus valores de extinção.

Certamente os métodos baseados em sementes para segmentação de imagens por região são conhecidos por fornecer resultados satisfatórios em várias aplicações, sendo geralmente fáceis de estender para imagens multidimensionais. Por exemplo, temos o método da Transformada Imagem-Floresta Orientada (OIFT) [Mansilla *et al.* (2013); Miranda e Mansilla (2014)], tal como discutido previamente no capítulo de introdução (Capítulo 1). A OIFT tem demonstrado possuir um bom balanço entre acurácia e robustez [Tavares *et al.* (2017b)] para a segmentação de um determinado objeto alvo com base em sementes fornecidas pelo usuário, permitindo a incorporação de várias restrições de alto nível, incluindo restrições de forma [de Moraes Braz e Miranda (2014); de Moraes Braz *et al.* (2019, 2020); Mansilla e Miranda (2013b)], restrições de conectividade [Mansilla e Miranda (2016); Mansilla *et al.* (2016)], entre outras [Condori e Miranda (2022); Leon *et al.* (2021b)].

O objetivo deste capítulo é incorporar restrições de alto nível dos objetos no contexto de segmen-

tação não supervisionada. Mais especificamente, neste capítulo estendemos a OIFT para realizar a segmentação não supervisionada de imagens gerando uma partição hierárquica da imagem por sucessivos cortes ótimos em grafos, levando a um novo método, chamado OIFT Não Supervisionada (UOIFT - *Unsupervised OIFT*), que pode ser customizado para diferentes objetos, de acordo com suas polaridades de borda, o que pode ser útil na segmentação de objetos que tenham propriedades de contorno bem definidas, como geralmente ocorre em muitas aplicações de imagens médicas. Como mostrado em [Cousty *et al.* \(2018\)](#), uma hierarquia de partições pode ser representada como um mapa de saliência [[Cousty e Najman \(2011\)](#)]. A partir dessa hierarquia, as segmentações podem ser facilmente obtidas por cortes nas representações a fim de fornecer as regiões conexas.

A Figura 5.1 apresenta a ideia central deste capítulo, que é explorar a polaridade de borda em segmentação não supervisionada de imagens baseada em grafos dirigidos. A polaridade da borda nos permite incluir o conhecimento prévio dos objetos de interesse na segmentação, favorecendo a seleção de objetos com transições de fronteira que interligam pixels com propriedades absolutas e/ou relativas conhecidas (por exemplo, sinal de variação de intensidade). Por exemplo, a Figura 5.1a mostra uma imagem sintética, contendo regiões escuras e claras, a ser segmentada em cinco regiões diferentes. Entretanto, métodos tradicionais não supervisionados, baseados em grafos não dirigidos, tais como Bacias Hidrográficas (Watershed), Cortes Normalizado (NCut) e o algoritmo de *Single-Linkage*, não conseguem distinguir os diferentes tipos de polaridade de borda, dando como saída uma mistura de regiões claras e escuras, tal como apresentado na Figura 5.1c. Nosso método proposto pode favorecer uma polaridade específica (isto é, do claro para escuro ou o oposto). As hierarquias calculadas a partir da Figura 5.1b, que corresponde aos *superpixels* da Figura 5.1a, são representadas pelos mapas de saliência mostrados nas Figuras 5.1e,h; e as segmentações associadas obtidas pelos cortes horizontais nas hierarquias são ilustradas correspondentemente nas Figuras 5.1f,i. Finalmente, nas Figuras 5.1d,g,j são representadas as energias das sementes de cada objeto em forma de um ranqueamento, conforme será explicado posteriormente.

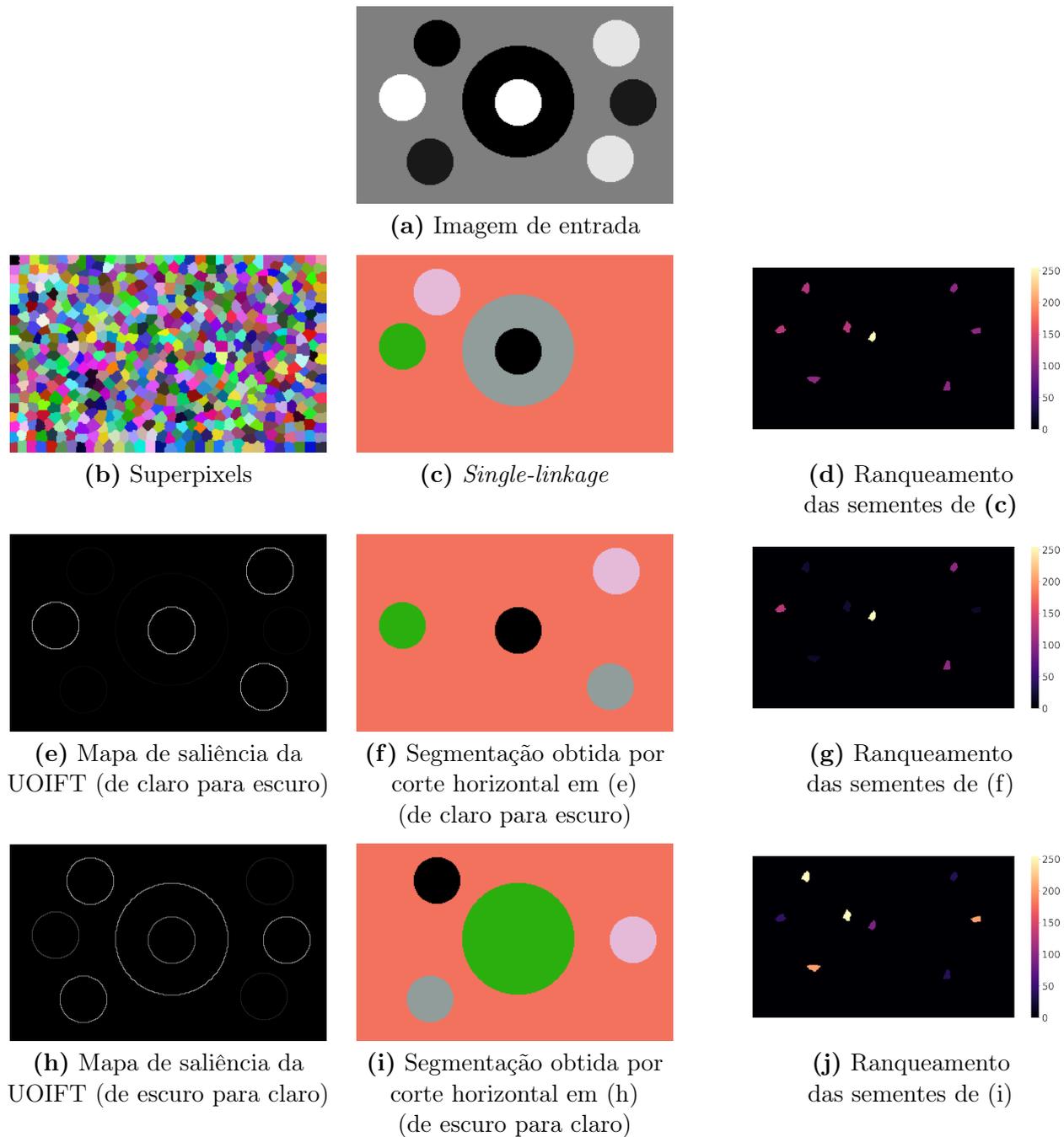


Figura 5.1: Exemplo da ideia central da UOIFT: (a) Imagem de entrada com 320×200 pixels. (b) Imagem dividida em 640 superpixels pela IFT-SLIC [Alexandre et al. (2015)]. (c) A segmentação em cinco regiões pelo algoritmo de Single-Linkage via uma árvore geradora mínima (MST) do Grafo de Regiões Adjacentes (RAG). (d) Sementes candidatas ranqueadas, segundo suas energias pela UOIFT sem polaridade de borda, levam ao mesmo resultado descrito em (c). (e,h) Mapas de saliência [Cousty et al. (2018)] das segmentações hierárquicas resultantes pelo método proposto para pixels de claro para escuro e seu oposto. (f,i) Os resultados de segmentação da UOIFT em cinco regiões, com polaridade favorecendo transições: (f) de pixels claros para escuros e (i) de pixels escuros para claros. (g,j) Ranqueamentos de sementes segundo suas energias pela UOIFT, resultantes para pixels de claro para escuro e seu oposto.

Embora métodos por decomposição por limiarização da intensidade de uma imagem via árvores de componentes [Silva e Lotufo (2011); Souza et al. (2017)] possam diferenciar objetos claros e escuros, como domos e bacias, respectivamente, eles são sensíveis a problemas de inomogeneidade (Figuras 5.2a-b). Suponha que estamos interessados em segmentar o retângulo central claro da Figura 5.2a. Espera-se que fosse um domo na *max-tree*, mas não pode ser obtido via decomposição por limiarização, uma vez que possui uma ampla gama de intensidades (de 118 a 185) que se so-

brepõem ao fundo devido ao efeito de inomogeneidade (Figura 5.2b). Por outro lado, os métodos que exploram o contraste local na forma de pesos de arestas são menos afetados pelo efeito de inomogeneidade, no entanto podem falhar ao misturar polaridades inconsistentes (Figura 5.2c-d). A UOIFT explora o contraste local das regiões da imagem ao lado da polaridade de borda, sendo robusto em relação às variações de iluminação e efeitos de inomogeneidade (Figuras 5.2e-f), além de não se restringir a imagens de um único canal.

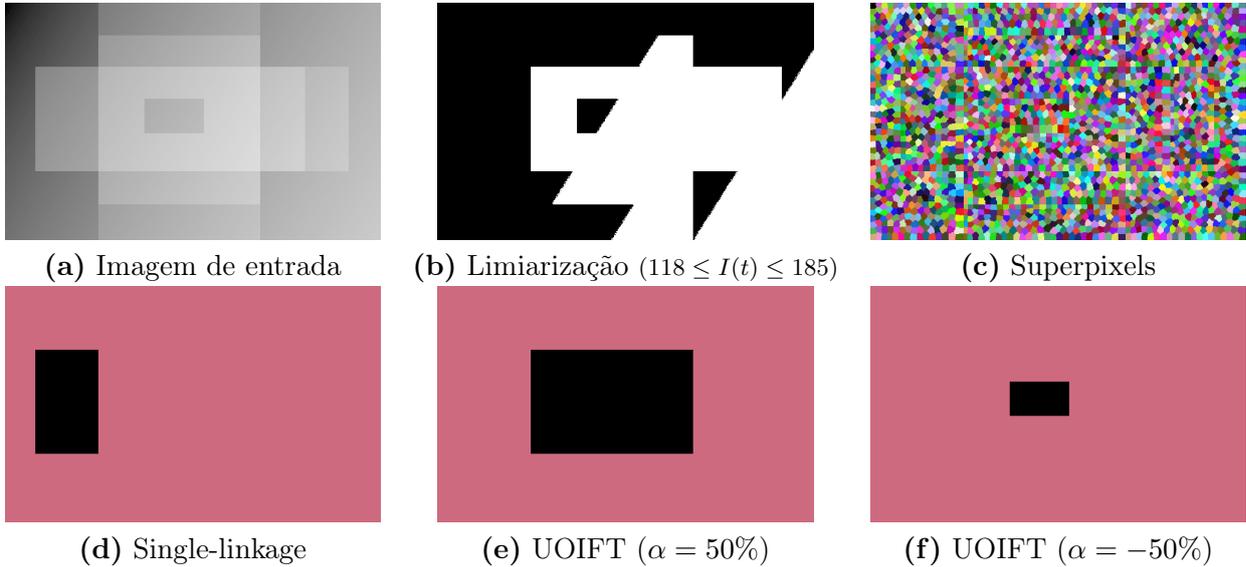


Figura 5.2: Exemplo da UOIFT em uma imagem sintética com inomogeneidade: (a) Imagem de entrada com 620×393 pixels. (b) Uma tentativa de obter o retângulo mais claro via limiarização falha devido ao problema de inomogeneidade. (c) Imagem dividida em 2418 superpixels via IFT-SLIC [Alexandre et al. (2015)]. (d) A segmentação em duas regiões via o algoritmo Single-Linkage usando MST do grafo de regiões adjacentes combina transições de pixels de claro para escuro e de escuro para claro. UOIFT com polaridade favorecendo as transições: (e) de pixels claros para escuros e (f) de pixels escuros para claros.

O método proposto é baseado no algoritmo da Transformada Imagem-Floresta (IFT) [Falcão et al. (2004)], que possui implementações com complexidade linearítmica $\mathcal{O}(n \log n)$, sendo portanto, muito mais rápido comparado a outros métodos baseados em cortes em grafos [Carballido-Gamio et al. (2004); Shi e Malik (2000); Wang e Sinkind (2003); Wang e Siskind (2001)].

Diferentemente de [Krähenbühl e Koltun (2014)], nosso método explora funções de conectividade não monotonicamente incrementais em grafos direcionados, conforme corroborado por vários estudos [Condori e Miranda (2022); Condori et al. (2017)]. O método proposto teoricamente abrange como caso particular o algoritmo de *Single Linkage* via uma árvore geradora mínima (MST) e usualmente requer um menor número de partições da imagem para isolar com precisão as regiões de interesse desejadas com polaridade conhecida, em comparação com outras abordagens comumente usadas na literatura.

A seguir é descrito o método proposto de segmentação não supervisionada da UOIFT.

5.1 Transforma Imagem-Floresta Orientada Não Supervisionada (UOIFT)

Nesta seção, será demonstrado o uso de uma função de custo *Não Monotonicamente Incremental* (isto é, uma função de conectividade que não satisfaz a definição apresentada na Seção 3.1.2), no arcabouço da IFT, para calcular cortes ótimos para segmentação não supervisionada. A OIFT não supervisionada é então obtida, mediante divisões binárias sucessivas do domínio da imagem, via minimização da energia $\varepsilon_\infty(\mathcal{O})$ em grafos direcionados, através de múltiplas execuções da IFT,

resultando em uma hierarquia de partições, que conserva a consistência da polaridade de borda de suas regiões. Portanto, pretendemos avançar no estado da arte em hierarquia de partições, considerando relações assimétricas na composição de hierarquias.

5.1.1 Definições

A seguir apresentamos as definições utilizadas na UOIFT:

Definição 5.1.1 (Conjunto de arcos de corte). *Para uma dada partição dos nós do grafo $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$ em dois conjuntos \mathbf{X} e $\mathcal{V} \setminus \mathbf{X}$, temos que o conjunto de arcos em seu corte de \mathbf{X} para $\mathcal{V} \setminus \mathbf{X}$ é definido da seguinte forma:*

$$\mathcal{C}(\mathbf{X}) = \{ \langle s, t \rangle \in \mathcal{A} : s \in \mathbf{X} \wedge t \notin \mathbf{X} \} \quad (5.1)$$

Esta definição é análoga a Equação 3.8 (página 32) apresentada na Seção 3.3. Cada conjunto de corte $\mathcal{C}(\mathbf{X})$ possui em valor de energia correspondente, aqui denotado por $E(\mathbf{X})$, tal como na versão dual da OIFT apresentada na Seção 3.3, em que esta energia havia sido designada por $\bar{\epsilon}_\infty(\mathbf{X})$ (Equação 3.10 da página 32).

Definição 5.1.2 (Energia de corte). *A energia de um objeto $\mathbf{X} \subset \mathcal{V}$ é o menor peso de arco do conjunto de corte $\mathcal{C}(\mathbf{X})$:*

$$E(\mathbf{X}) = \bar{\epsilon}_\infty(\mathbf{X}) = \min_{\langle s, t \rangle \in \mathcal{C}(\mathbf{X})} \omega(\langle s, t \rangle) \quad (5.2)$$

De maneira similar a definição de universo de objetos que separam dois conjuntos de sementes dada pela Equação 3.7 da Seção 3.3 (página 32), aqui consideramos a seguinte definição:

Definição 5.1.3 (Universo de objetos). *Seja $\mathcal{U}(x, y)$ o universo de todos os objetos possíveis que separam os nós x e y , de modo que x representa um ponto interior do objeto e y representa o fundo, tal como definido da seguinte forma:*

$$\mathcal{U}(x, y) = \{ \mathbf{X} \subset \mathcal{V} : x \in \mathbf{X} \wedge y \in \mathcal{V} \setminus \mathbf{X} \} \quad (5.3)$$

Definição 5.1.4 (Objeto de energia ótima). *Um objeto \mathbf{X}_{opt} possui energia ótima, satisfazendo as restrições x e y , se e somente se:*

$$E(\mathbf{X}_{opt}) = \max_{\mathbf{X} \in \mathcal{U}(x, y)} E(\mathbf{X}) \quad (5.4)$$

O algoritmo da OIFT [Miranda e Mansilla (2014)] (ver Seção 3.3), usando x e y como sementes internas e externas, respectivamente, calcula um objeto ótimo $\mathbf{X}_{opt} \in \mathcal{U}(x, y)$ maximizando a energia definida na Equação 5.2 em um grafo dirigido simétrico, de acordo com o Teorema 2 de Mansilla e Miranda (2013a).

Conforme visto na seção 3.3, a OIFT é construída sobre o arcabouço da IFT considerando a função de conexidade f^σ (Equação 3.12) para caminhos em um dígrafo simétrico, sendo que, neste trabalho, usamos $\mathcal{S}_o = \{x\}$ e $\mathcal{S}_b = \{y\}$ na Equação 3.12.

Para o propósito da segmentação não supervisionada, para um dado ponto de referência r no fundo, pretendemos encontrar um nó $t' \in \mathcal{V} \setminus \{r\}$, resultando em um objeto de energia máxima $\mathbf{X}_{opt} \in \mathcal{U}(t', r)$ pela OIFT com $\mathcal{S}_o = \{t'\}$ e $\mathcal{S}_b = \{r\}$, que também seja de energia máxima entre todos os resultados em $\mathcal{U}(r) = \bigcup_{t \in \mathcal{V} \setminus \{r\}} \mathcal{U}(t, r)$, isto é, $E(\mathbf{X}_{opt}) = \max_{\mathbf{X} \in \mathcal{U}(r)} E(\mathbf{X})$.

Felizmente, t' pode ser obtido eficientemente tomando $t' = \arg \max_{t \in \mathcal{V} \setminus \{r\}} V(t)$, onde V é o mapa de custo da IFT usando a função de conexidade f_{\max} (Equação 3.1), com $\mathcal{S} = \{r\}$ na definição de custo de caminhos triviais (Equação 3.5), calculada sobre o grafo transposto G^T , de acordo com o Lema 1 de Bejar e Miranda (2015).

Para obter o próximo melhor corte, por argumentos análogos, poderíamos pensar em repetir o processo e selecionar $t'' = \arg \max_{t \in \mathcal{V} \setminus \{r, t'\}} V(t)$, mas visto que todos os pixels dentro da região de

robustez de t' possuem o mesmo valor de energia [Tavares *et al.* (2017a)], isto poderia resultar na seleção de um segundo pixel equivalente ao primeiro e possivelmente sendo seu vizinho.

Uma outra ideia seria, portanto, considerar para t'' apenas pixels **fora do núcleo** de t' , mas neste caso a segmentação futura de uma borda aninhada interna poderia se tornar impossível. Uma forma mais apropriada é considerar:

$$t'' = \arg \max_{t \in \mathcal{V} \setminus \{r, t'\}} V'(t) \quad (5.5)$$

tal que V' é o mapa de custos via IFT usando f_{max} com $\mathcal{S} = \{r, t'\}$ no grafo transposto G^T e assim por diante.

No entanto, observe que isso requer o cálculo de um novo mapa de custos para cada corte consecutivo da hierarquia, aumentando o custo computacional. Então, a seguir analisamos outra solução alternativa mais adequada, conforme explicado na seguinte seção.

5.1.2 Próximo objeto de energia máxima com a função de conectividade f_ω

Uma abordagem alternativa para calcular o próximo objeto de energia máxima é considerar:

$$t' = \arg \max_{t \in \mathcal{V} \setminus \{r\}} V^*(t) \quad (5.6)$$

tal que V^* é o mapa de custo via IFT usando a função de conectividade f_ω (Equação 3.3) com $\mathcal{S} = \{r\}$ no grafo transposto G^T , mas esta última abordagem tem a vantagem de nos permitir ranquear os nós de acordo com sua ordem não crescente de valores no mapa V^* , de modo que o próximo corte com energia máxima possa ser facilmente selecionado como:

$$t'' = \arg \max_{t \in \mathcal{V} \setminus \{r, t'\}} V^*(t) \quad (5.7)$$

assim, não exigindo a re-computação do mapa de custos para tratar objetos aninhados e/ou vários objetos separados rodeados por regiões de baixo contraste (Figuras 5.1), porque para alcançar um pixel em um desses objetos $\mathcal{O} \subset \mathcal{V}$, qualquer caminho calculado pelo Algoritmo 1 (Página 30), usando a função de conectividade f_ω no grafo transposto G^T a partir de r terá que atravessar seu arco de saída mais fraco $\langle a, b \rangle \in \mathcal{C}(\mathcal{O})$, isto é, $\omega(\langle a, b \rangle) = E(\mathcal{O})$, e conseqüentemente temos que $V^*(a) = \omega(\langle a, b \rangle) = E(\mathcal{O})$.

Assim, a hierarquia de partições pode ser criada, de acordo com o seguinte algoritmo proposto.

5.2 Algoritmo da UOIFT

Nesta seção apresentamos o algoritmo proposto da UOIFT (Algoritmo 2), que é composto por três etapas.

Algoritmo 2. – ALGORITMO DA OIFT NÃO SUPERVISIONADA (UOIFT)

ENTRADA: Grafo da imagem $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, um nó r de referência para o fundo e o número desejado de regiões k .
SAÍDA: Partição do grafo em k regiões.
AUXILIARES: Conjuntos \mathcal{S}_o e \mathcal{S}_b de sementes de objeto e fundo, respectivamente.

1. **Primeira etapa:**

Calcular $V^* : \mathcal{V} \rightarrow \mathbb{R}$ pela IFT com f_ω (Equação 3.3) e $\mathcal{S} = \{r\}$ no grafo transposto G^T .

2. **Segunda etapa:**

Ordenar os nós de $\mathcal{V} \setminus \{r\}$ em uma sequência em ordem não crescente de seus custos em V^* , obtendo $\{t_1, t_2, \dots, t_n\}$, tal que $V^*(t_i) \geq V^*(t_{i+1})$, $i = 1, \dots, n-1$, sendo que $n = |\mathcal{V}| - 1$.

3. **Terceira etapa:**

$\mathcal{S}_b \leftarrow \{r\}$.

Para Cada t_i , $i = 1, \dots, k-1$, **Faça**

$\mathcal{S}_o \leftarrow \{t_i\}$.

Calcular rótulos L via OIFT em G com sementes \mathcal{S}_o e \mathcal{S}_b , de objeto e fundo, respectivamente.

$\mathcal{C} \leftarrow \{ \langle a, b \rangle \in \mathcal{A} : L(a) \neq L(b) \}$.

Remova de G todos os arcos em \mathcal{C} .

$\mathcal{S}_b \leftarrow \mathcal{S}_b \cup \{t_i\}$.

Devolva todos componentes fortemente conexos do grafo G resultante como k regiões.

O Algoritmo 2 gera uma segmentação hierárquica por divisões binárias sucessivas, levando no final para uma partição com k regiões. A ordenação de nós na Etapa 2 do Algoritmo 2 (Figuras 5.1g e j) pode ser realizada em tempo linear $O(n)$, quando os pesos são números inteiros pequenos, caso contrário, se tem um tempo linearítmico $O(n \log n)$ usando um *heap* binário. Cada execução da OIFT na Etapa 3 tem complexidade linearítmica em relação ao número dos nós n envolvidos. Uma vez que a UOIFT é baseada em múltiplas execuções da OIFT (sendo aplicada em componentes fortemente conexos menores a cada iteração), consideramos um Grafo de Regiões Adjacentes (RAG), no qual as regiões são os *superpixels* calculados pela IFT-SLIC [Alexandre *et al.* (2015); Vargas-Muñoz *et al.* (2018)], em vez de usar os pixels diretamente (Figura 5.1b). O nó de referência inicial para o fundo foi considerado como sendo o primeiro *superpixel* no canto superior/esquerdo da imagem.

A fim de explorar a polaridade de borda, para imagens em escala de cinza consideramos a seguinte atribuição de peso de arco:

$$\omega(\langle s, t \rangle) = \begin{cases} |I(t) - I(s)| \times (1 + \alpha) & \text{se } I(s) > I(t) \\ |I(t) - I(s)| \times (1 - \alpha) & \text{caso contrário} \end{cases} \quad (5.8)$$

tal que os pesos $\omega(\langle s, t \rangle)$ são uma combinação de uma medida não direcionada de dissimilaridade $|I(t) - I(s)|$ entre os *superpixels* vizinhos s e t , multiplicada por um fator de orientação $\alpha \in [-1, 1]$, de tal modo que $\alpha > 0$ favorece a segmentação de objetos claros em um fundo escuro (Figura 5.1f) e $\alpha < 0$ favorece a orientação oposta (Figura 5.1i), e $I(t)$ indica a intensidade média dentro do *superpixel* t .

A seguir, é discutida a atribuição de pesos no caso envolvendo imagens coloridas.

5.3 UOIFT para Imagens Coloridas

No caso de imagens coloridas, são apresentadas duas opções para o cálculo dos pesos do digrafo. Nas duas soluções, iniciamos computando a magnitude do vetor diferença entre as cores médias dos *superpixels*, utilizando o espaço de cores do CIELAB, dado que as diferenças de cores neste espaço melhor refletem nossa percepção visual das variações de cores¹. As diferenças entre as opções se

¹O CIELAB foi concebido como um espaço perceptualmente uniforme, onde uma dada mudança numérica corresponde a uma mudança similar percebida na cor.

dão por meio da forma como é empregada a polaridade de borda. Na Equação 5.9, a polaridade é especificada de modo similar à Equação 5.8, a fim de caracterizar transições de claro para escuro ($\alpha > 0$) ou de escuro para claro ($\alpha < 0$), comparando os valores do canal de luminosidade.

$$\omega(\langle s, t \rangle) = \begin{cases} \|\vec{I}(t) - \vec{I}(s)\| \times (1 + \alpha) & \text{se } l_s > l_t \\ \|\vec{I}(t) - \vec{I}(s)\| \times (1 - \alpha) & \text{se } l_s < l_t \\ \|\vec{I}(t) - \vec{I}(s)\| & \text{caso contrário} \end{cases} \quad (5.9)$$

sendo $\vec{I}(s) = \langle l_s, a_s, b_s \rangle$ e $\vec{I}(t) = \langle l_t, a_t, b_t \rangle$ as cores médias dos *superpixels* s e t no espaço de cores CIELAB² e o símbolo $\|\cdot\|$ denota a norma de um vetor.

Para imagens coloridas, podemos considerar também a seguinte atribuição alternativa de pesos nos arcos em que são especificadas duas cores de referência que caracterizam a transição de borda:

$$\omega(\langle s, t \rangle) = \begin{cases} \|\vec{I}(t) - \vec{I}(s)\| \times (1 + \alpha) & \text{se } d_s + d_t < d \\ \|\vec{I}(t) - \vec{I}(s)\| \times (1 - \alpha) & \text{caso contrário} \end{cases} \quad (5.10)$$

sendo $\vec{I}(s) = \langle l_s, a_s, b_s \rangle$ e $\vec{I}(t) = \langle l_t, a_t, b_t \rangle$ as cores médias dos *superpixels* s e t no espaço de cores CIELAB, o símbolo $\|\cdot\|$ denota a norma de um vetor e

$$d = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}, \quad d_s = \sqrt{(a_s - a_1)^2 + (b_s - b_1)^2}, \quad d_t = \sqrt{(a_t - a_2)^2 + (b_t - b_2)^2}$$

para duas cores de referência

$$C_1 = \langle l_1, a_1, b_1 \rangle \text{ e } C_2 = \langle l_2, a_2, b_2 \rangle$$

no espaço de cores CIELAB, como ilustrado na Figura 5.3. Observe que a luminosidade não é considerada no cálculo de d_s , d_t e d para ter menos sensibilidade a mudanças na iluminação. O fator de orientação $\alpha \in [-1, 1]$, favorece a segmentação de objetos com transições de borda de C_1 para C_2 no fundo circundante para $\alpha > 0$.

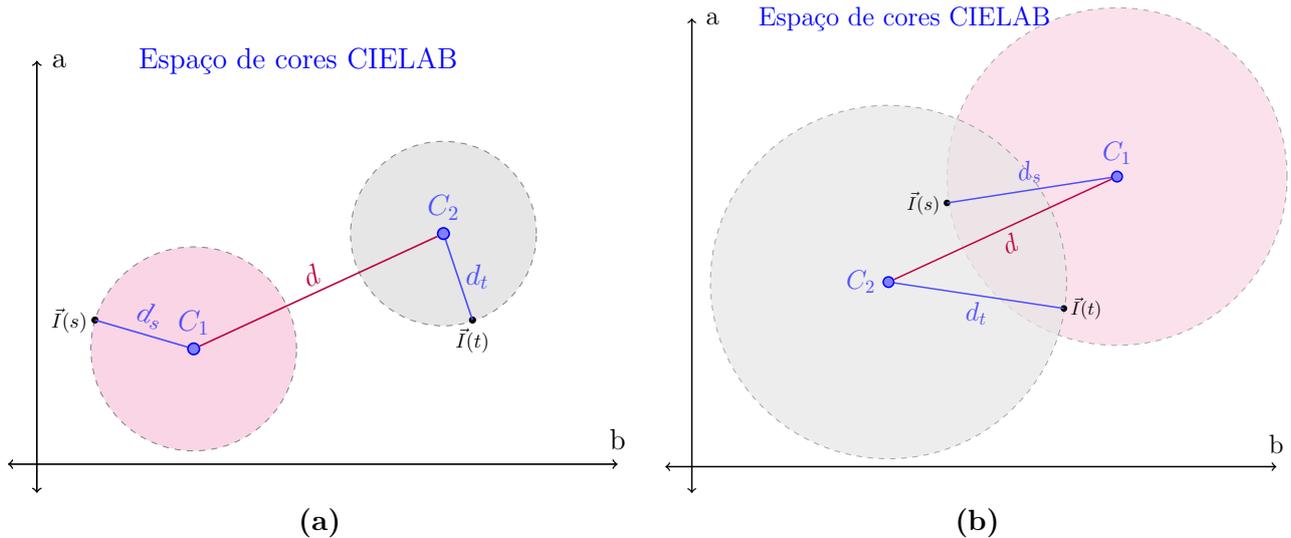


Figura 5.3: Exemplo do cálculo de d_s , d_t e d , para as cores $\vec{I}(s)$ e $\vec{I}(t)$ dos superpixels s e t , bem como as cores de referência C_1 e C_2 , no espaço de cores CIELAB: Condição (a) favorável e (b) desfavorável na Equação 5.10

Para simplificar, estamos considerando apenas cores aqui, mas observe que, em teoria, esse

²Espaço de cores CIELAB $\langle l, a, b \rangle$: Representa a relação quantitativa entre as cores nos três eixos, l é a luminosidade com os valores de 0 (preto) à 100 (branco), e as cores a (componente vermelho/verde) e b (componente verde/azul).

modelo poderia ser estendido para cenários mais gerais envolvendo atributos de textura, tais como atributos baseados em frequência e direcionalidade.

5.3.1 Explicação do algoritmo UOIFT com imagens sintéticas

A fim de ajudar a esclarecer o funcionamento do Algoritmo 2, na Figura 5.4 são apresentados três exemplos detalhados da execução da UOIFT para o cálculo da região com energia máxima, para três diferentes valores do fator de orientação ($\alpha \in \{-50\%, 0, +50\%\}$).

Cada exemplo começa com a inicialização do nó de referência \mathbf{r} na posição superior esquerda, a partir do qual é calculado o mapa de custos V^* pela IFT com a função de conectividade f_ω no grafo transposto G^T , o que permite o ranqueamento das sementes de acordo com seus valores de energia. O nó com maior valor em V^* é selecionado como semente de objeto e a primeira partição do grafo é obtida mediante a OIFT. Este processo deve então ser repetido até atingir o número desejado de k regiões da partição, na Figura 5.4 apenas a primeira partição é ilustrada.

No primeiro caso (Figura 5.4a-b), quando $\alpha = 0$, ou seja sem polaridade de borda, devemos notar que a orientação das transições de intensidade ao longo da borda de corte da partição obtida não é uniforme, misturando transições dos tipos claro para escuro e do tipo escuro para claro (Figuras 5.4a-b). Já nos outros casos, quando $\alpha = 50\%$ ou $\alpha = -50\%$, temos como resultado partições que conservam uma polaridade de borda consistente, com variações de intensidade de claro para escuro (Figuras 5.4c-d) ou vice-versa (Figuras 5.4e, f), ao longo de todo o contorno.

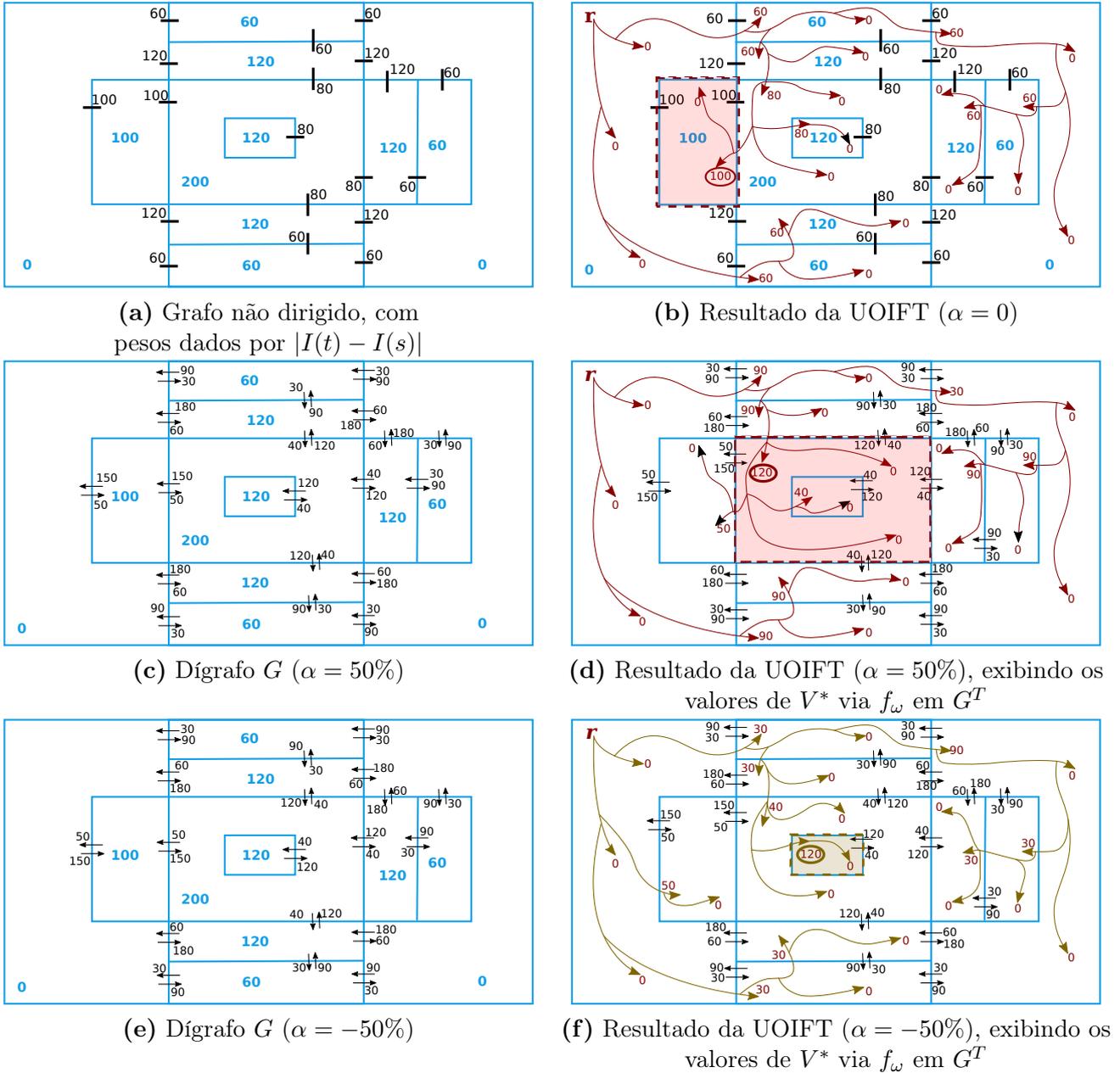


Figura 5.4: Exemplos da primeira partição do Algoritmo da UOIFT para diferentes valores de α : (a) Grafo não direcionado com pesos dados por $|I(t) - I(s)|$ e valores de intensidade de brilho indicados nas regiões homogêneas. (b) Resultado da UOIFT com $\alpha = 0$, mostrando também os caminhos pela IFT e seus valores no mapa de conexidade V^* usando a função de conexidade f_w no grafo não dirigido a partir do ponto de referência \mathbf{r} no fundo. (c) Grafo dirigido obtido a partir do grafo em (a) com $\alpha = 50\%$. (d) Resultado da UOIFT com $\alpha = 50\%$, indicando os caminhos e o mapa de conexidade V^* via função f_w no grafo transposto de (c). (e) Grafo dirigido obtido a partir do grafo (a) com $\alpha = -50\%$. (f) Resultado da UOIFT com $\alpha = -50\%$, indicando os caminhos e o mapa de conexidade V^* via função f_w no grafo transposto de (e).

O segundo exemplo na Figura 5.5, com vários objetos, visa exemplificar a sequência de partições geradas. O mapa de conexidade calculado das energias das regiões a partir do ponto de referência \mathbf{r} , usando a função de conexidade f_w é ilustrado na Figura 5.5b. A sequência de partições geradas segue uma ordem não crescente das energias calculadas, isto é, o ranqueamento 12, 8, 6, 4, 0, considerando $k = 5$.

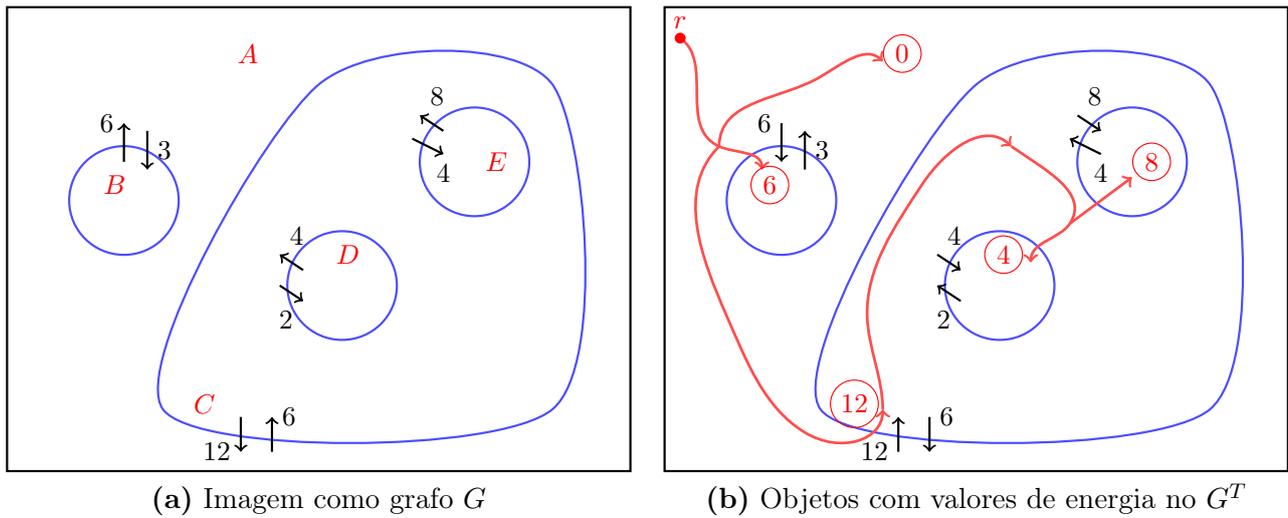


Figura 5.5: Exemplo do mapa de conexidade com a função f_w : (a) Digrafo G , (b) valores do mapa de conexidade para os diversos objetos, com caminhos iniciando a partir do ponto de referência r no grafo transposto da imagem. Para todos os objetos, são exibidas as suas respectivas energias.

A seguir, a Figura 5.6 mostra as divisões binárias sucessivas do exemplo da Figura 5.5, resultando em uma hierarquia de partições, que é exibida com suas respectivas energias.

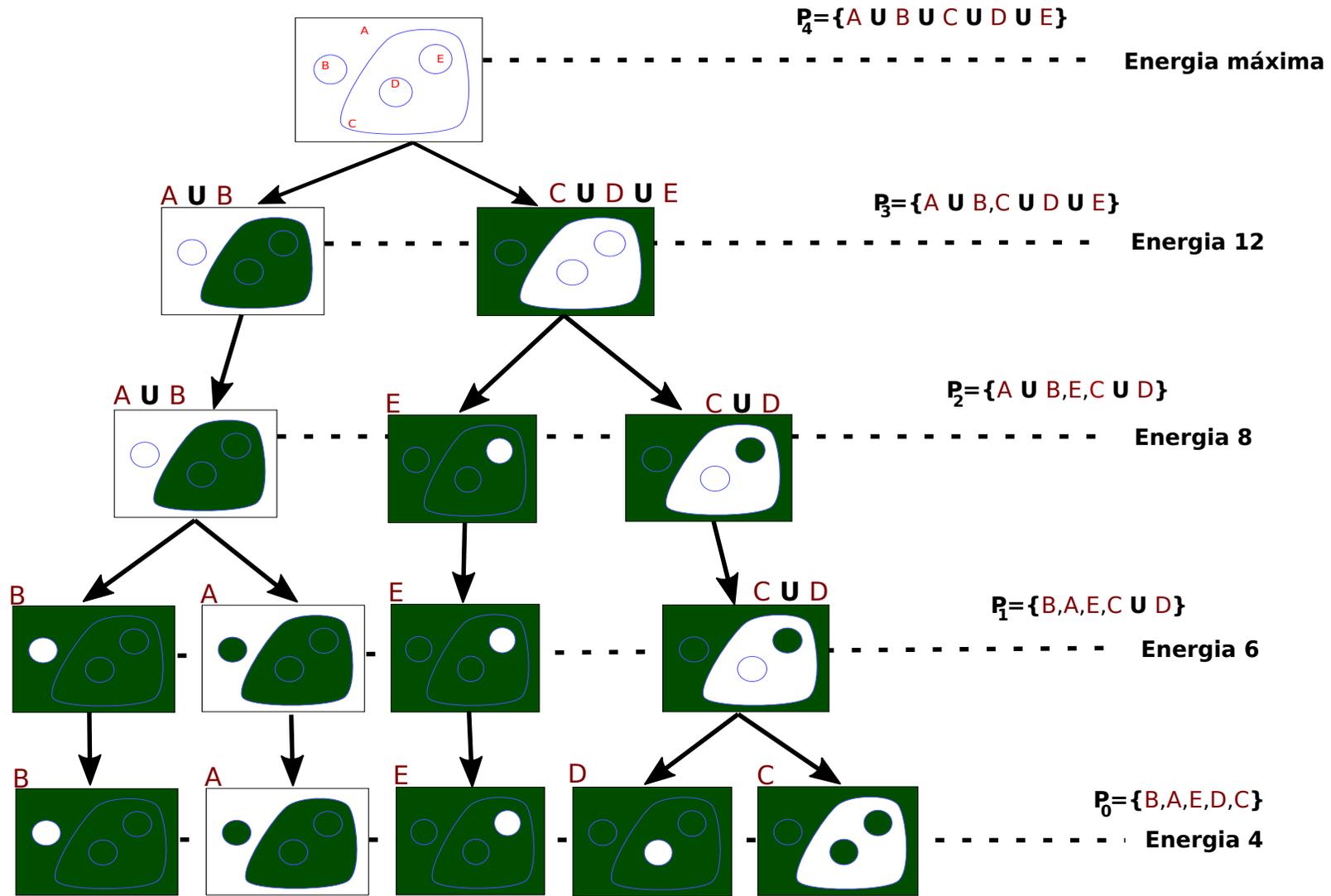


Figura 5.6: Exemplo da hierarquia $\mathcal{H} = (P_0, P_1, P_2, P_3, P_4)$ computada pela UOIFT sobre o grafo da Figura 5.5a.

5.3.2 Relação da UOIFT com uma Árvore de Espalhamento Mínimo (MST)

Para $\alpha = 0$, o Algoritmo 2 (Página 57) abrange como um caso particular a versão *top-down* do algoritmo de vizinhos mais próximos (*Single-Linkage*), obtido por remoções sucessivas de arestas de máximo peso da Árvore de Espalhamento Mínimo (MST- *Minimum Spanning Tree*) do grafo de regiões adjacentes (RAG), anteriormente descrito.

Para verificar isso, observe que a primeira etapa do Algoritmo 2 corresponde ao cálculo da MST para $\alpha = 0$, já que neste caso temos $G = G^T$ (isto é, o grafo passa a ser equivalente a um grafo não direcionado) e a IFT com f_ω torna-se o algoritmo de *Prim*.

Além disso, o maior valor $V^*(t_i)$ na segunda etapa corresponde a uma aresta $\langle Pr(t_i), t_i \rangle$ de máximo peso na MST, na qual $Pr(t_i)$ é o nó predecessor de t_i , calculado pelo Algoritmo 1 da IFT (Página 30) com a função de conexidade f_ω .

Finalmente, na terceira etapa, temos que a OIFT se torna uma segmentação por *Watershed* de marcadores para $\alpha = 0$, que pode ser obtida igualmente pela remoção da aresta de peso máximo da MST.

5.4 Tratamento de Empate no Algoritmo UOIFT

A fim de tratar possíveis ambiguidades que possam ocorrer durante a execução do Algoritmo 2, em função de empates na função de custo da IFT ou no ranqueamento da sequência de nós ordenados pela energia, nesta seção discutimos uma possível política de tratamento de empates.

5.4.1 Problema de Empate

O problema de empate no algoritmo da UOIFT pode acontecer em duas situações:

1. O empate ocorre na fila de prioridade \mathcal{Q} da IFT, durante a execução da primeira etapa do Algoritmo 2.
2. Empates podem também ocorrer durante o ranqueamento dos nós, na segunda etapa do Algoritmo 2, em função de empates no mapa V^* .

O primeiro problema é ilustrado na Figura 5.7. Observe que ambas regiões A e B podem ser alcançadas a partir do nó r , com custo 5 na função f_ω no grafo transposto G^T , logo tendo seus nós de fronteira inseridos na fila \mathcal{Q} com custo 5. Dependendo de qual elemento é removido primeiro da fila de prioridade (um pixel de A ou de B), podemos ter uma dentre as duas soluções distintas apresentadas à direita na Figura 5.7 (primeiro caso ou segundo caso).

Observe que cada caso da Figura 5.7 resulta em uma mapa de conexidade distinto para V^* . Observe também que no primeiro caso, passamos a ter um empate no mapa V^* entre as regiões A e B , dada que ambas apresentam um elemento com energia igual a 5, o que resultará em uma ambiguidade durante o ranqueamento que será realizado na segunda etapa do Algoritmo 2, levando a outros dois subcasos para o primeiro caso, aqui denotados por opção 1 e opção 2.

O impacto dos diferentes casos (e subcasos) sobre a hierarquia de partições resultantes é apresentado nas Figuras 5.8 e 5.9, que ilustram as possíveis divisões binárias sucessivas. Observe que, no primeiro caso da Figura 5.8, temos também o empate das regiões A e B , ambas com energia igual a 5, o que induz diferentes hierarquias durante o ranqueamento da segunda etapa do Algoritmo 2.

A fim de reduzir tais ambiguidades, propomos uma política de desempate (Seção 5.4.2) de modo a favorecer que o resultado apresentado pela opção 1 do primeiro caso seja sempre obtido (Figura 5.8a). Esta escolha se deve ao fato de que, em geral, é preferível garantir que os elementos maiores sejam obtidos nas primeiras divisões, sendo seus detalhes internos revelados somente em níveis mais baixos da hierarquia. Observe que na Figura 5.8a, a união das regiões A e B (formando um objeto da elipse completa) é primeiramente obtida, na primeira divisão binária, o que não ocorre nos demais casos.

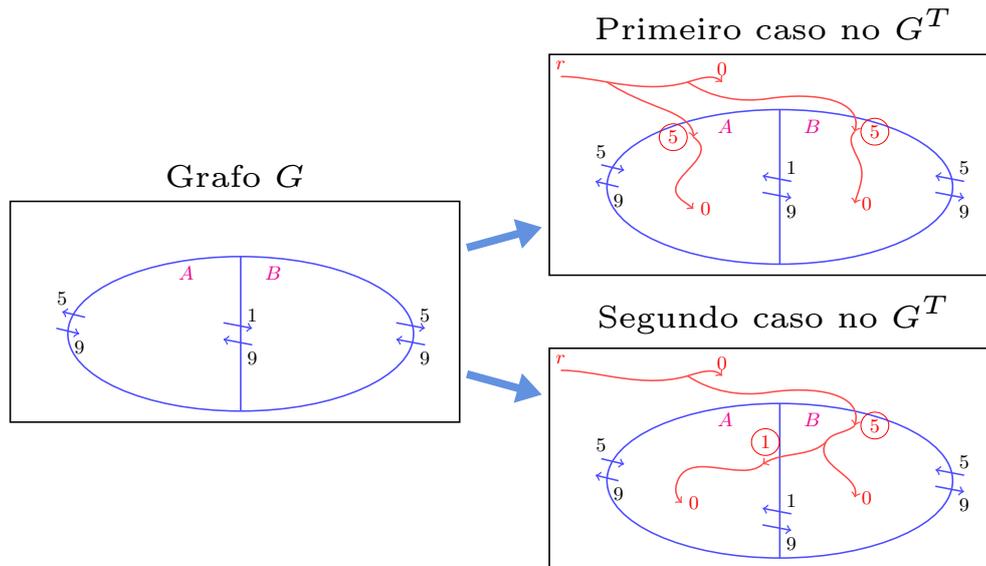


Figura 5.7: Exemplo de dois mapas de conexidade distintos que podem ser obtidos a partir de um mesmo grafo G : Primeiro caso ocorre quando um pixel da região A é removido primeiro da fila de prioridade \mathcal{Q} , com custo 5 na função f_ω da IFT no grafo transposto indicado à direita. O segundo caso ocorre quando um pixel da região B deixa primeiro a fila de prioridade \mathcal{Q} , levando a uma posterior conquista da região A , com custo 1 para o primeiro elemento (quando cruza a fronteira) e 0 para os demais elementos. Por outro lado, observe que o primeiro caso causa um segundo empate, agora no mapa V^* (isto é, tanto A quanto B possuem elementos com energia igual a 5), para efeito do ranqueamento que é realizado na segunda etapa do Algoritmo 2, dando origem a outros dois subcasos, que serão denotados por opção 1 e opção 2.

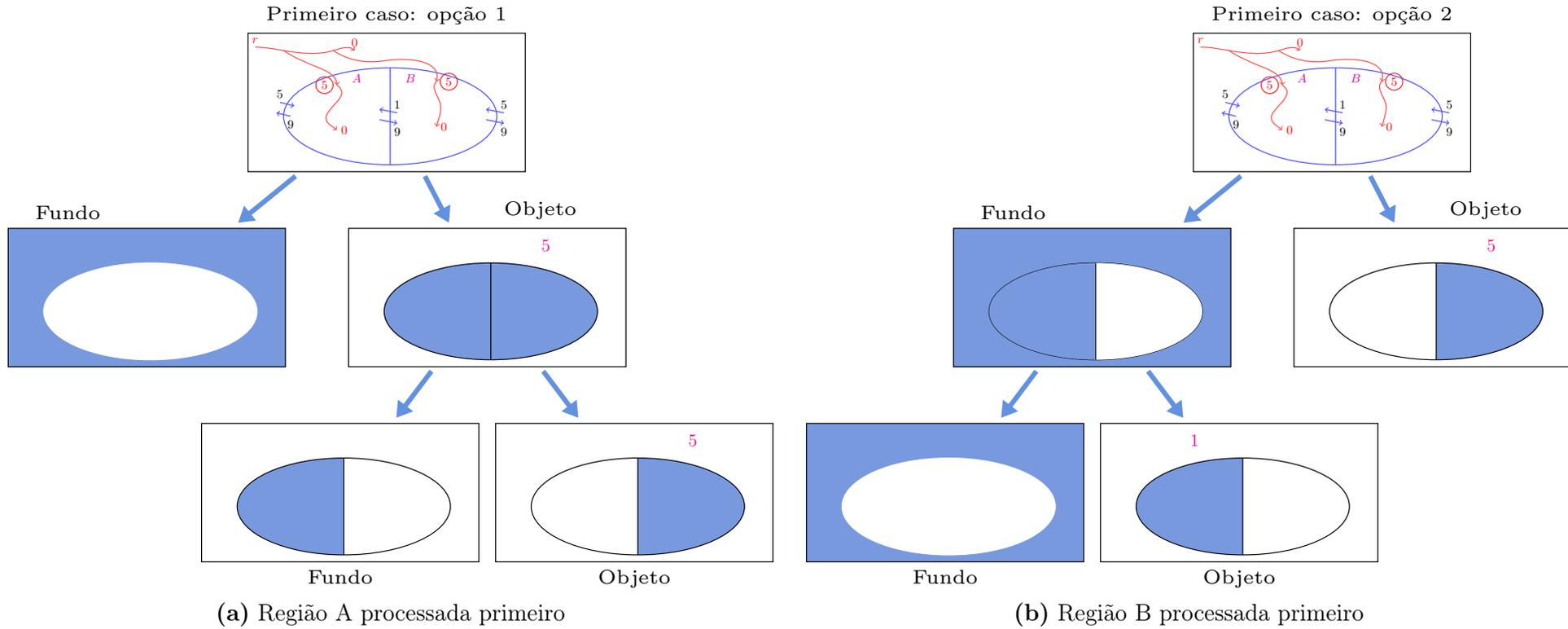


Figura 5.8: Exemplo das possíveis divisões binárias sucessivas resultantes do primeiro caso do mapa de conectividade da Figura 5.7, diferenciando os dois possíveis subcasos (opção 1 e opção 2): (a) Quando a região A é processada primeiro no ranqueamento de energia, e (b) quando a região B é processada primeiro.

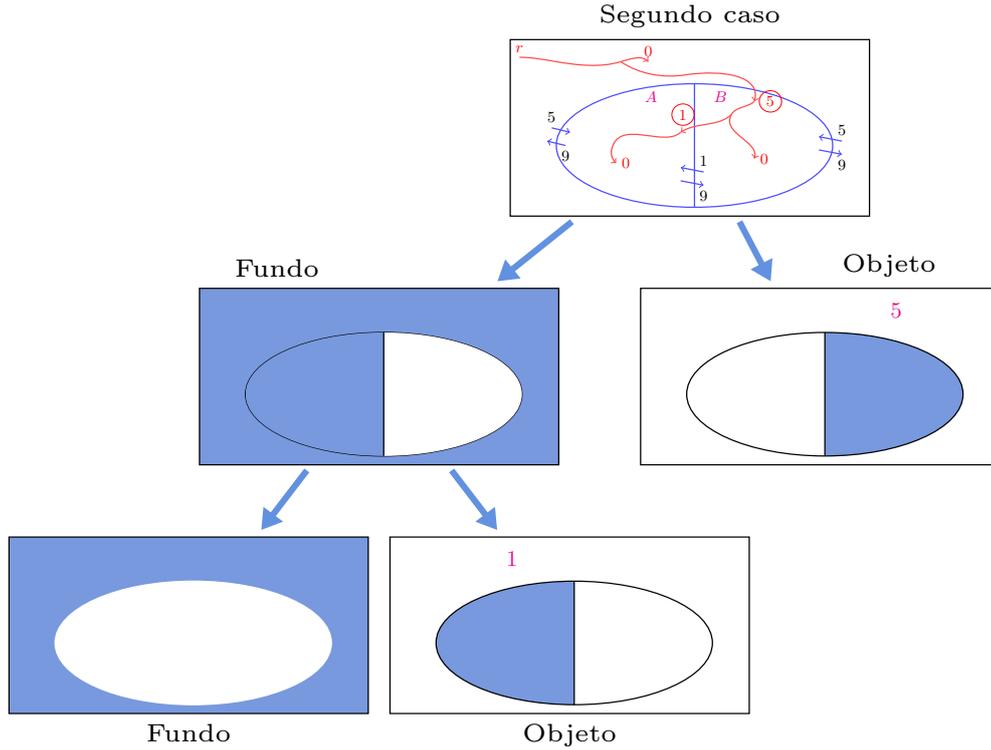


Figura 5.9: Exemplo das divisões binárias sucessivas do segundo caso do mapa de conexidade da Figura 5.7.

5.4.2 Política de Desempate

Para reduzir ambiguidades no Algoritmo 2 da UOIFT, propomos o uso da seguinte política de desempate. Durante ambos os casos, o cálculo do mapa de custos V^* pela IFT usando a função de conexidade f_ω com $\mathcal{S} = \{r\}$ no grafo transposto G^T , e a ordenação de forma não crescente de custos na Etapa 2 do Algoritmo 2, consideramos um segundo componente lexicográfico de desempate para favorecer os nós com maior tamanho de Componente Conexo Direcionado (DCC - *Directed Connected Component*) em um grafo residual apropriado (ver Seção 2.6, página 15). Portanto, consideramos o seguinte custo lexicográfico no cálculo da função de conexidade f_ω :

$$\langle c_1, c_2 \rangle = \left\langle \omega(\langle s, t \rangle), \frac{1}{|DCC_G(t, \omega(\langle s, t \rangle))|} \right\rangle \quad (5.11)$$

no qual o símbolo $|\cdot|$ indica a cardinalidade de um conjunto, e o arco $\langle s, t \rangle$ é obtido do grafo transposto G^T , porém $DCC_G(t, \epsilon)$ é calculado em relação ao grafo G , como:

$$DCC_G(t, \epsilon) = DCC_{G_\geq}(t) \quad (5.12)$$

no qual G_\geq é o grafo residual obtido após a remoção de todos os arcos com pesos $\geq \epsilon$ do grafo G , e $DCC_{G_\geq}(t)$ é calculado pela Equação 2.1, porém usando o grafo residual G_\geq .

De modo similar, na Etapa 2 do Algoritmo 2, consideramos energias lexicográficas $\langle e_1, e_2 \rangle = \langle V^*(t_i), |DCC_G(t_i, V^*(t_i))| \rangle$, enquanto ordenamos os nós para obter $\{t_1, t_2, \dots, t_n\}$.

O motivo para favorecer o DCC com maior tamanho é porque cada DCC corresponde à uma segmentação via ORFC (veja a definição da ORFC - *Oriented Relative Fuzzy Connectedness* apresentada em Bejar e Miranda (2015)), que está sempre contida no resultado da OIFT (Proposição 1 de Tavares *et al.* (2017b)). Como a ORFC define uma grande parte da região segmentada pela OIFT, ao favorecer os DCC de tamanho maior obtemos primeiro as regiões maiores em caso de empate. Da Proposição 1 de Bejar e Miranda (2015), podemos também concluir que:

$$s \in DCC_G(t, \epsilon) \implies |DCC_G(s, \epsilon)| \leq |DCC_G(t, \epsilon)| \quad (5.13)$$

Portanto, nenhum ponto s dentro do DCC de t pode ser um ponto base de um DCC maior. Assim, processamos primeiro as partes maiores e refinamos suas partes constituintes (Figura 5.10b).

Um exemplo de tratamento de ambiguidades, pela política de desempate proposta, é apresentado nas Figuras 5.10 e 5.11. Para o grafo G apresentado na Figura 5.10a, temos dois possíveis mapas de conectividade distintos V^* que podem ser obtidos no grafo transposto G^T , que são dados pelas Figuras 5.10b-c. O primeiro caso (Figura 5.10b) ocorre quando um pixel da região central é removido da fila Q com custo 5, antes de qualquer pixel da região à esquerda. Já o segundo caso (Figura 5.10c) ocorre quando um pixel da região à esquerda é removido primeiro de Q com custo 5.

Na nossa solução proposta, no caso de empates no peso dos arcos, o novo custo lexicográfico no cálculo de f_ω (Equação 5.11) irá favorecer a conquista de um nó em uma região com maior tamanho de componente conexo (DCC). Logo, a solução apresentada em Figura 5.10c será sempre selecionada. Mesmo adotando esta solução (Figura 5.10c), ainda resta um segundo problema de ambiguidade a ser tratado na Etapa 2 do algoritmo da UOIFT, pois note que temos mais de uma região com energia 5 no mapa V^* apresentado na Figura 5.10c. Aqui novamente, tal como explicado anteriormente, resolvemos o empate considerando energias com uma segunda componente de desempate lexicográfica de modo a favorecer o nó com maior tamanho de DCC. Logo, teremos as divisões binárias sucessivas conforme ilustradas na Figura 5.11, dado que o pixel de energia 5 na região esquerda apresenta maior DCC.

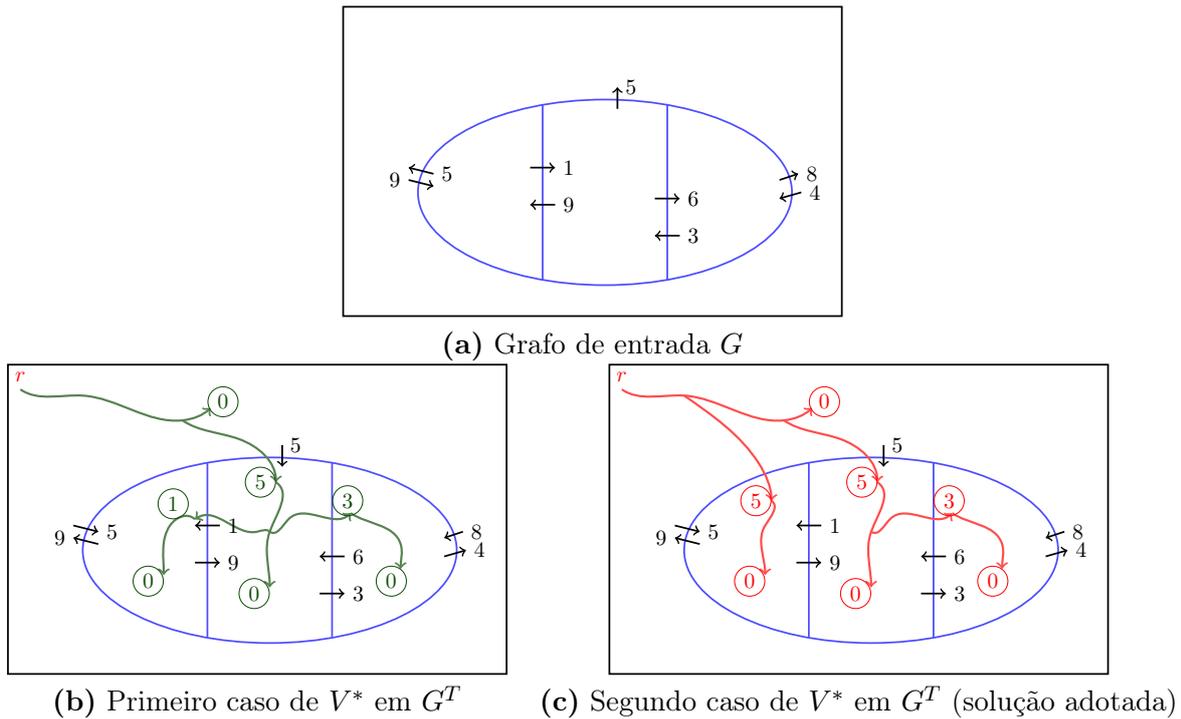


Figura 5.10: Exemplo de tratamento de empate: (a) Grafo de entrada. (b-c) Os dois possíveis mapas de conectividade distintos V^* que podem ser obtidos pela IFT com f_ω em G^T . (c) Solução proposta com quebra de empate favorecendo a região com maior tamanho de componente conexo (DCC).

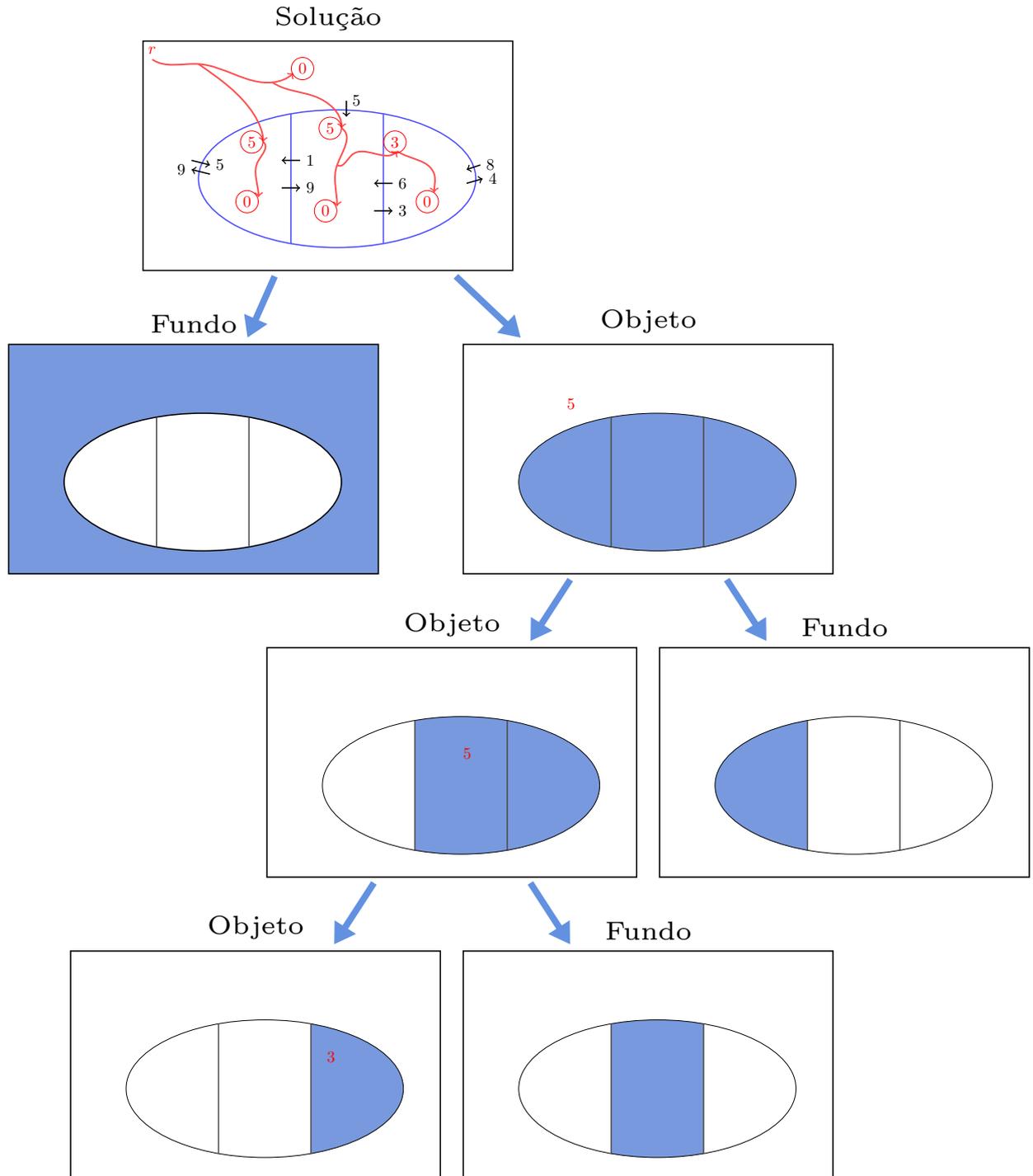


Figura 5.11: Tratamento de empate na Etapa 2 do algoritmo da UOIFT, favorecendo o nó com maior tamanho de DCC, no caso de empate das energias em V^* . No empate de energia 5, a região à esquerda é selecionada para a realização da primeira divisão por possuir maior DCC. A solução completa de divisões binárias sucessivas é apresentada no diagrama.

5.5 Manipulação de Hierarquia pelo valor de Extinção de Volume

Nesta seção, discutimos como favorecer objetos com maior área/volume, via reorganização da hierarquia produzida pela UOIFT, conservando as restrições de polaridade de borda. Ou seja, pretendemos favorecer as regiões com maior extensão espacial, suprimindo ruídos e regiões pequenas, porém garantindo transições de borda com a polaridade correta desejada.

Com base nos resultados apresentados na Seção 2.7.4, sabemos que qualquer hierarquia pode ser igualmente representada na forma de um mapa de saliência de contornos. Logo, dada uma hierarquia computada pela UOIFT, tal como descrita na Seção 5.2, esta pode ser igualmente representada pelo grafo chamado de mapa de saliência. Uma hierarquia pode também ser representada na forma de um histórico de fusões, que equivale a uma árvore que representa uma MST do mapa de saliência.

Por exemplo, considere a hipotética hierarquia de partições $\mathcal{H} = (P_0, P_1, P_2, \dots, P_9)$ da Figura 5.12. Esta hierarquia pode ser representada igualmente pelo grafo chamado de mapa de saliência da Figura 5.13a. Ou seja, a partir da hierarquia de partições \mathcal{H} são calculadas as arestas de corte $\phi_G(\mathcal{H})$, dando como resultado o mapa de saliência $\Phi_G(\mathcal{H})$ (Figura 5.13a), tal como descrito na Seção 2.7.4 (Página 19).

Uma hierarquia pode também ser representada pela árvore de espalhamento mínimo (MST - *Minimum Spanning Tree*) do seu mapa de saliência, como ilustrado na Figura 5.13b. Em geral, consideramos que as regiões da partição P_1 correspondem ao primeiro nível de agrupamento inicial de pixels, dado por um método de geração de *superpixels*. Logo, as regiões de P_1 , circuladas e rotuladas de A a G na Figura 5.14a, formam os *superpixels* do nosso exemplo. Observe que a MST do mapa de saliência pode ser representada alternativamente na forma de um histórico de fusões, entre nós vizinhos com seus respectivos níveis de energia (pesos das arestas da MST), tal como ilustrado na Figura 5.14b.

A hierarquia, tal como representada nas Figuras 5.14a-b, indica que a partição mais relevante é a de energia 8 (divisão mais relevante no topo da hierarquia), que corresponde a P_8 e que possui uma região composta por um único elemento $A = \{a\}$. Geralmente, uma região pequena pode ser fruto de um ruído ou detalhes menores e irrelevantes. Logo, em P_8 estamos dando muita importância para um único nó isolado. Será que é possível reorganizar uma hierarquia, para reestruturá-la de maneira a dar maior importância para regiões maiores e mais significativas da imagem? Uma resposta positiva é obtida pelo emprego dos valores de extinção da *Min-Tree* da MST no nível de arestas (Figuras 5.14c-d).

A *Min-Tree* é uma representação compacta da árvore de componentes, que monitora componentes conexos em um processo de decomposição por limiarização, em que as folhas são os mínimos regionais. Os pixels interconectados por arestas de peso zero (conexões de energia mais baixa) representam bacias (folhas da árvore), em uma interpretação dos valores como uma superfície topográfica. A medida que subimos a energia (isto é, subimos no relevo), bacias vizinhas se fundem, criando os nós internos da árvore. Por exemplo, na Figura 5.14c, quando é subido para o nível de energia 1, a região D se funde com a região E , gerando um novo nó interno da *Min-Tree*. O mesmo ocorre com as regiões B , F e G que se fundem também no nível de energia 1. No nível de energia 2, temos a fusão de C com G , que, por sua vez, já estava fundido com B e F , e assim por diante.

Para reestruturar ou reorganizar o grafo G da Figura 5.13a através dos valores de extinção, primeiramente precisamos calcular atributos crescentes para os nós da *Min-Tree*. Aqui estamos interessados no atributo de volume, pois este permite ponderar de maneira integrada a área juntamente com os desníveis de energia. Observe que, no caso de uma hierarquia computada pela UOIFT, a energia já incorpora em si a informação da polaridade de borda das regiões, logo sua combinação com a informação de área permite o favorecimento de regiões maiores com a polaridade correta.

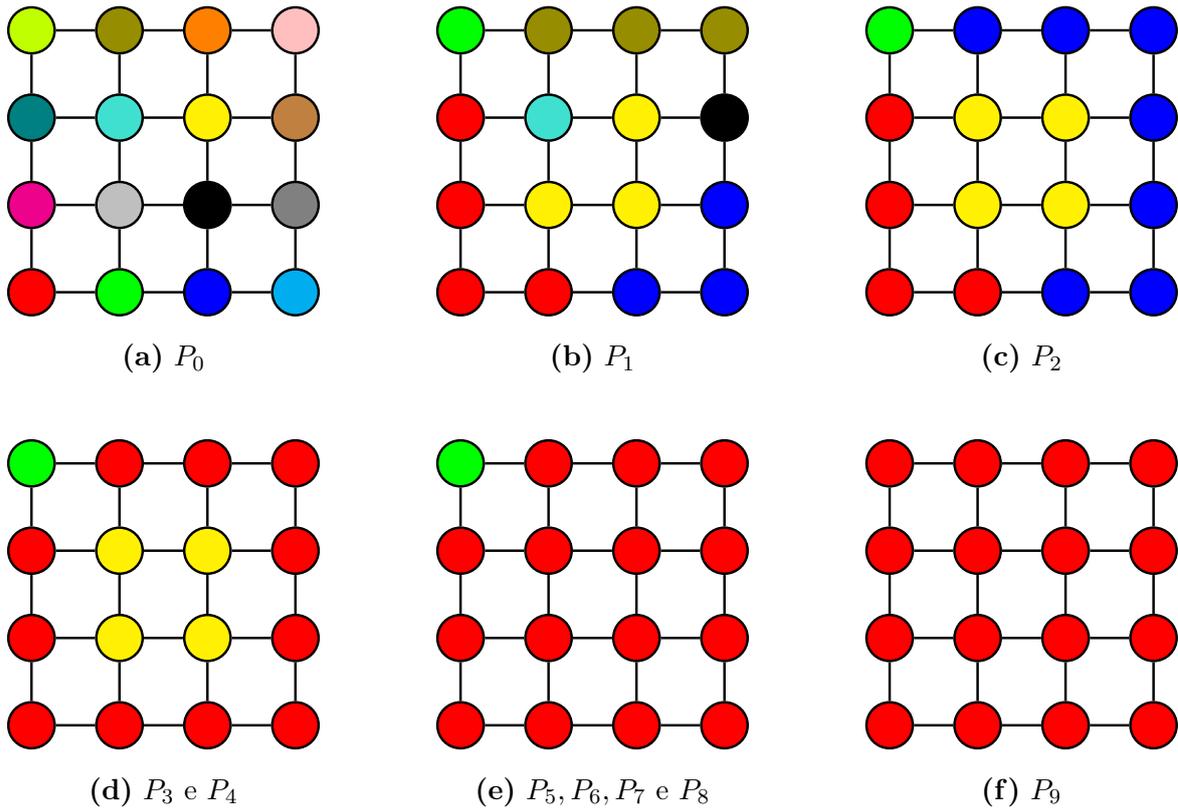


Figura 5.12: Ilustração de uma hierarquia $\mathcal{H} = (P_0, P_1, P_2, \dots, P_9)$. Para cada partição, cada região é representada com uma cor distinta, ou seja, dois nós com a mesma cor pertencem à mesma região.

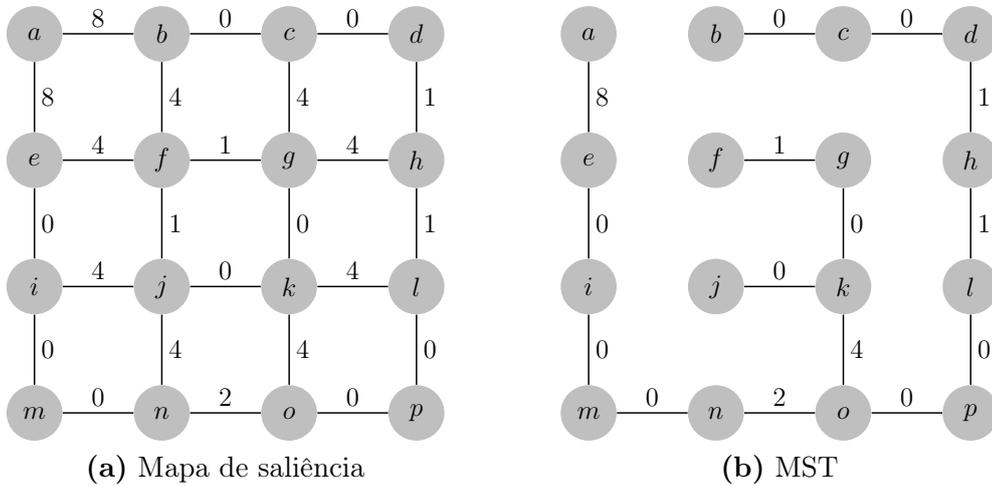


Figura 5.13: Representações alternativas a partir da hierarquia \mathcal{H} da Figura 5.12: (a) Ilustração do mapa de saliência $\Phi_G(\mathcal{H})$. (b) MST do mapa de saliência.

Na Figura 5.14d são exibidos os valores dos atributos de área e volume, em cores azul e vermelho, respectivamente, sobre os nós da *Min-Tree*. O atributo de área é mostrado por auxiliar no processo de cálculo do atributo de volume. Estes atributos são calculados da seguinte forma:

- **Área:** A área de uma folha da *Min-Tree* é o número de elementos do seu componente conexo no grafo da MST, após a remoção de todas as arestas com peso maior do que zero. Ou seja, cada folha corresponde a uma bacia, que é dada por um *superpixel* no nosso caso. Assim, por exemplo, a folha da *Min-Tree* correspondente a região $C = \{e, i, m, n\}$ possui área 4.

Nos níveis superiores, a área de um nó interno é a soma das áreas dos seus nós filhos, por exemplo, para o nó interno, que é pai das folhas B, F e G na *Min-Tree*, temos uma área igual a $7 = 3 + 1 + 3$, pois as áreas dos filhos B, F e G são 3, 1 e 3, respectivamente. O atributo de

área é calculado em todos os nós da árvore até a raiz, sendo a área da raiz igual ao número total de nós do grafo (16 para um grafo de imagem 4×4 , como no exemplo). Figura 5.14d mostra em cor azul o valor do atributo de área sobre cada nó da *Min-Tree*.

- **Volume:** O volume de um nó da *Min-Tree* é a área do nó multiplicada pelo desnível (altura) em relação ao seu pai, mais a soma dos volumes dos seus filhos. Por exemplo, para o nó com volume 16 em vermelho (união de D e E na Figura 5.14d), primeiro multiplicamos sua área que é 4 pelo desnível em relação ao seu pai que é $4 - 1 = 3$, e depois somamos os valores dos volumes dos seus filhos D e E , que são 1 e 3, respectivamente, o que dá como resultado $16 = 4 \times 3 + 1 + 3$. O atributo de volume é calculado em todos os nós da árvore até a raiz, sendo seus valores exibidos em vermelho sobre os nós da *Min-Tree* na Figura 5.14d.

O valor de extinção do atributo de volume é calculado apenas nas folhas da *Min-Tree*, tal como mostrado na Figura 5.14d. O valor de extinção de um mínimo regional para qualquer atributo crescente (altura, área, volume, etc) é o tamanho máximo de um filtro de atributo de modo que esse extremo ainda exista após a filtragem [Silva e Lotufo (2011)].

Por exemplo, na Figura 5.14d, observe que a bacia A seria eliminada após a aplicação de um filtro de volume, preenchendo todas bacias com volume menor ou igual a 8, logo seu valor de extinção de volume é 8. Note que A é menos relevante em comparação com seu irmão na *Min-Tree*, dado que este possui valor de volume 120, o que implica na sua não extinção para o mesmo filtro aplicado. O nó D é extinguido com valor 1, por ser menos relevante do que seu irmão E , que possui valor de volume 3, enquanto D possui volume 1. De modo análogo, o nó F também se extingui com valor 1. Já o nó C possui valor de extinção de 8, pois seu irmão é mais relevante tendo volume de 14, enquanto C possui volume de apenas 8. No caso dos nós B e G , temos que a sua fusão ocorre no nível de energia 1, sendo que eles apresentam um empate, pois ambos possuem volume 3. No exemplo, desempatamos em favor de G , logo B fica com valor de extinção de 3. O nó E tem valor de extinção 16, porque no caminho de volta até a raiz, ele possui um ancestral com valor 16, que é menor do que o volume de 44 de seu irmão. Finalmente, o nó G por ser o último e mais relevante é atribuído com o valor máximo (infinito).

Dado que o valor de extinção indica o quanto uma bacia resiste à aplicação de um filtro de atributo, ele nos fornece uma medida do grau de relevância de uma bacia em relação ao atributo considerado. Portanto, podemos ordenar as folhas da *Min-Tree*, segundo seus valores de extinção, em ordem não crescente, a fim de ranquear quais são as bacias mais relevantes nas primeiras posições.

A Figura 5.14e mostra a reorganização da hierarquia, mediante edição dos pesos das arestas da MST, usando os valores de extinção. Primeiro são comparadas as duas folhas mais relevantes da *Min-Tree*, e o menor valor de extinção entre elas substitui o valor da maior aresta no caminho interligando suas bacias correspondentes no grafo da MST, a fim de promover sua separação a um nível mais alto na nova hierarquia. A seguir, é escolhida a próxima folha mais relevante da *Min-Tree*, até trocar todas as arestas da MST, entre as respectivas bacias, pelos valores de extinção.

Por exemplo, na Figura 5.14e, as bacias mais relevantes do ponto de vista do atributo de volume são a G e a E , com valores de extinção ∞ e 16, respectivamente. Na MST, existe um único caminho entre as regiões G e E . Pegamos a maior aresta neste caminho, no caso a aresta $\langle k, o \rangle$ de peso 4, e trocamos seu valor pelo menor valor de extinção das duas folhas. Logo, no lugar do 4 colocamos o valor $16 = \min(16, \infty)$.

Agora para continuar, pegamos a próxima folha da *Min-Tree* com maior valor de extinção de volume. Por exemplo, podemos pegar C com valor 8. Agora, tomamos o caminho na MST interligando C , com alguma das bacias já vistas (G e E). Encontramos então a maior aresta neste caminho, ou seja tomamos a aresta $\langle n, o \rangle$ de peso 2, e trocamos seu peso para $8 = \min(\infty, 16, 8)$.

Depois pegamos A e trocamos a aresta $\langle a, e \rangle$ de peso 8 para $8 = \min(\infty, 16, 8, 8)$. e assim por diante, resultando em uma MST com novos valores, conseqüentemente, remodelando a hierarquia. Observe que na nova hierarquia (Figura 5.14e), a divisão mais relevante ocorre no nível 16, resultando em uma região central de 4 pixels. Logo, esta primeira quebra foi promovida para o topo da nova hierarquia, gerando uma divisão mais balanceada da imagem.

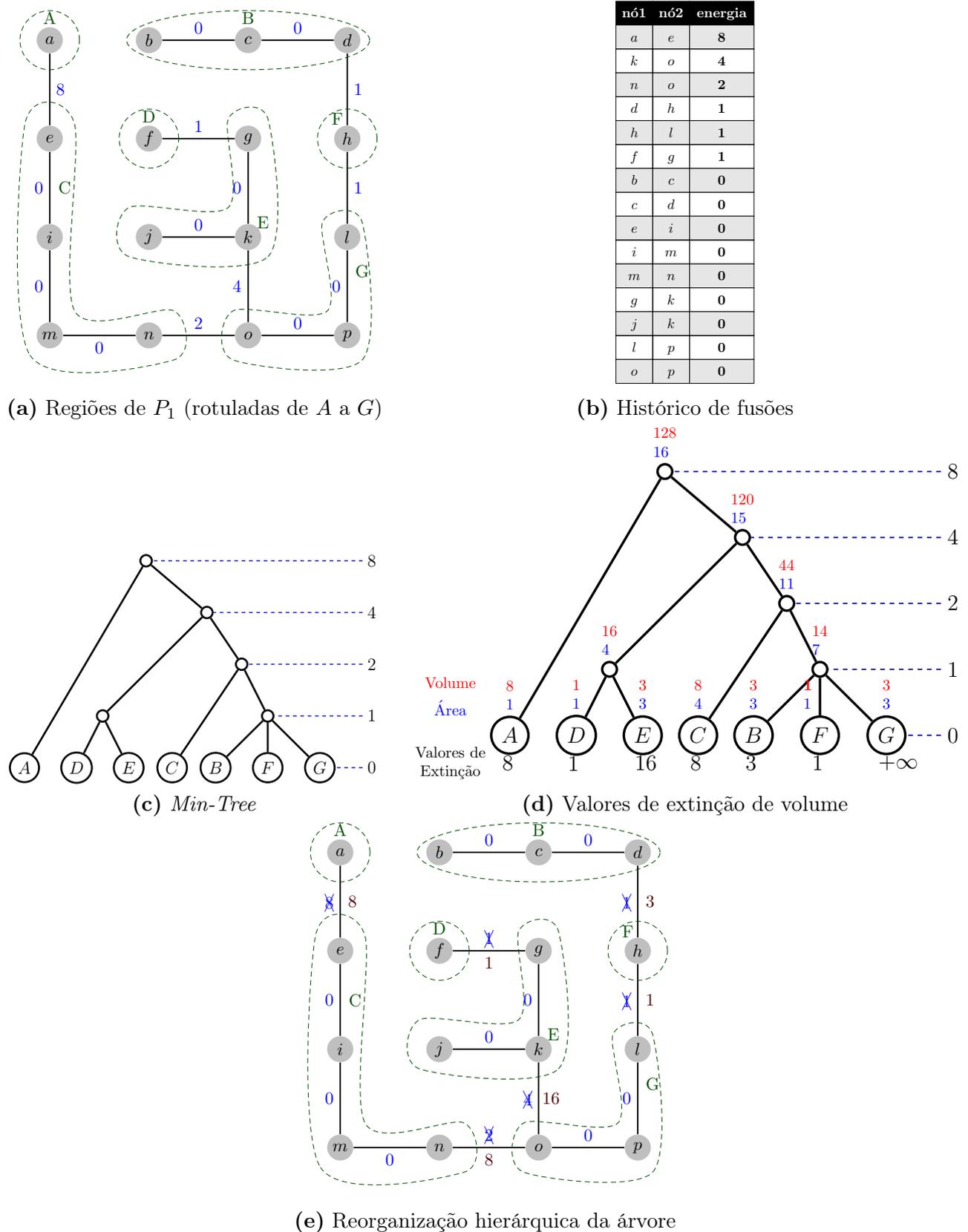


Figura 5.14: Processo de reorganização hierárquica pelo cálculo dos valores de extinção do atributo de volume na *Min-Tree*. (a) Regiões de P_1 circuladas na *MST* e rotuladas de A a G. (b) Representação da árvore na forma de um histórico de fusões, com os respectivos valores de energia. (c) *Min-Tree* da *MST*, a partir dos componentes conexos de P_1 . (d) Valores dos atributos de área (azul) e volume (vermelho) são indicados sobre os nós da *Min-Tree*. Já os valores de extinção de volume são exibidos abaixo dos nós folha. (e) Hierarquia final reorganizada na forma de uma árvore com novos pesos.

Na Figura 5.15 são apresentados os mapas de saliência das hierarquias geradas pela UOIFT, sem e com a reorganização pelo valor de extinção de volume aqui proposta. Nos exemplos, foi utilizada a UOIFT para segmentar objetos com transições de brilho de escuro para claro ($\alpha < 0$ na Equação 5.9), para favorecer a segmentação do padrão *Aruco* [Garrido-Jurado *et al.* (2014); Romero-Ramirez *et al.* (2018)]. Observe que após a reorganização da hierarquia pelo valor de extinção de volume, regiões pequenas, tais como ruídos, são suprimidas, dando preferência para regiões mais extensas da imagem.

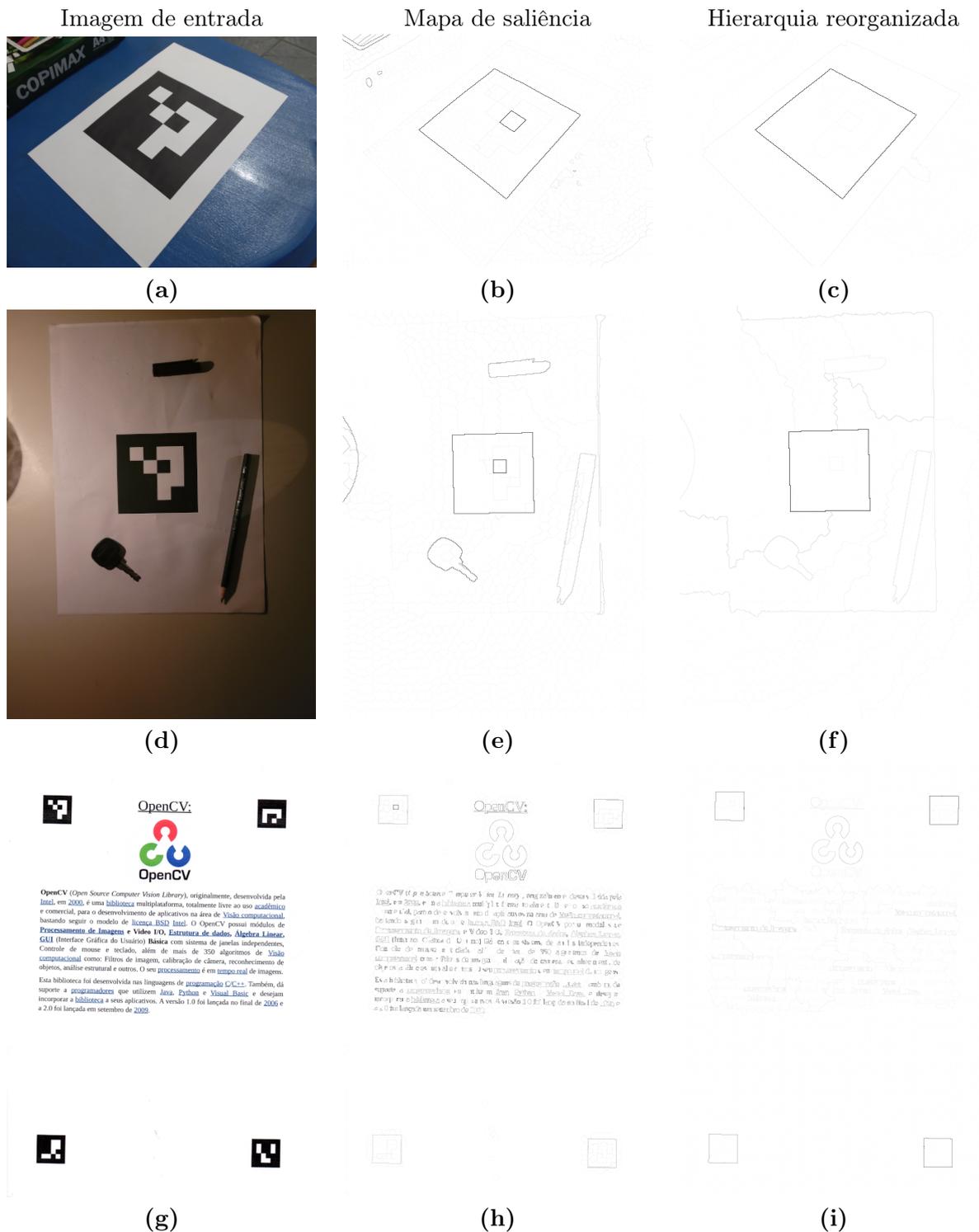


Figura 5.15: Exemplos de reorganização das hierarquias geradas pela UOIFT pelo filtro de extinção de volume para realce dos padrões Aruco: Primeira coluna: Imagens coloridas com padrão Aruco (a,d,g). Segunda coluna: (b,e,h) Mapa de saliência (na forma de uma imagem) da hierarquia gerada pela UOIFT (com $\alpha < 0$ na Equação 5.9, favorecendo transições de escuro para claro sobre o canal de brilho). Terceira coluna: Hierarquia reorganizada pelo filtro de extinção de volume. Observe que os quadrados os padrões Aruco são agora melhor realçados.

Este procedimento para reorganização de uma hierarquia pela UOIFT, visando regiões maiores nos níveis superiores, porém conservando a polaridade de borda adequada, tal como proposto na presente tese, resultou em uma monografia de trabalho de conclusão de curso (TCC) do estudante

Gabriel Miranda de Araújo³.

5.6 Resultados Experimentais

Foram realizados experimentos para comparar a segmentação proposta pela UOIFT com outros métodos baseados em grafos, a seguir:

- **MST** (*Minimum Spanning Tree*): Denota o agrupamento dos nós do Grafo de Regiões Adjacentes (RAG), obtido por remoções sucessivas de arestas de peso máximo da árvore geradora mínima, com pesos $\omega(\langle s, t \rangle) = |I(t) - I(s)|$, o que corresponde a uma versão *top-down* do algoritmo de *Single-Linkage*, também conhecido como agrupamento de vizinho mais próximo (*nearest neighbour clustering*). O algoritmo MST tem complexidade linearítmica $O(n \log n)$ em relação ao número de nós envolvidos no RAG.
- **FH**: Denota a abordagem não supervisionada por Felzenszwalb e Huttenlocher (2004), que calcula um predicado para medir a evidência de borda entre duas regiões, baseada na árvore geradora mínima calculada a partir do RAG. O algoritmo FH tem complexidade linearítmica $O(n \log n)$ em relação ao número de nós n do RAG.
- **HFH**: Denota um método posterior obtido por Hierarquização da segmentação de Felzenszwalb e Huttenlocher conforme proposto por Guimarães *et al.* [Guimarães *et al.* (2017)], com um pós-processamento de filtragem de área para eliminar pequenos componentes com o tamanho do *superpixel*.
- **EF+WS**: Denota a transformada de *Watershed* baseada na IFT [Audigier e Lotufo (2007)], após a aplicação de um filtro de extinção de volume (*EF - Extinction Filter*) [Silva e Lotufo (2011)] definido para preservar k folhas da *min-tree*, a fim de considerar apenas as bacias hidrográficas mais relevantes de um gradiente morfológico por um disco circular de raio 1. Para o filtro de extinção foi utilizado o código disponível na ferramenta iamxt [Souza *et al.* (2017)]. Sua *Min-tree* pode ser calculada em $O(N \times h + M)$, tal que N é o número de pixels, h é o número de níveis da imagem e $M = k \times N$, sendo k o número de vizinhos de cada pixel [Silva e Lotufo (2011)].
- **NCut**: Denota o Corte Normalizado [Shi e Malik (2000)], disponível na ferramenta *scikit-image*, calculado a partir do RAG pela IFT-SLIC [Alexandre *et al.* (2015); Vargas-Muñoz *et al.* (2019)]. A complexidade computacional do NCut é tipicamente $O(n^{3/2})$, tal que n é o número de nós no RAG usando o solucionador iterativo de autovalores pelo método de Lanczos. No entanto, o segundo autovalor é frequentemente mal condicionado, levando a uma lenta convergência na prática.

Note que o Algoritmo 2 da UOIFT (Página 57) engloba como caso particular o algoritmo *top-down* do *Single-Linkage* (MST), já que seu primeiro passo corresponde a um cálculo de MST para $\alpha = 0$, porque cada $V(t_i)$ selecionado no segundo passo do algoritmo corresponde a uma aresta de peso máximo da MST, conforme discutido na Seção 5.3.2.

5.6.1 Análise Qualitativa

Na Figura 5.16 é apresentado um exemplo em 3D usando supervoxels de tamanho $10 \times 10 \times 10$ *voxels* em uma imagem de ressonância magnética (MRI - *Magnetic Resonance Image*) do cérebro com $240 \times 240 \times 180$ *voxels*. Utilizando uma partição do volume em dez regiões ($k = 10$), já é possível obter os ventrículos laterais pelo algoritmo da UOIFT com polaridade $\alpha = -90\%$. O mesmo já não ocorre para a versão *top-down* do algoritmo de *Single-Linkage* via MST, utilizando o mesmo grafo de supervoxels.

³Manipulation of hierarchical segmentation of images based on saliency maps, IME-USP, 2020. url: <https://linux.ime.usp.br/gabriel/mac0499/monografia.pdf>

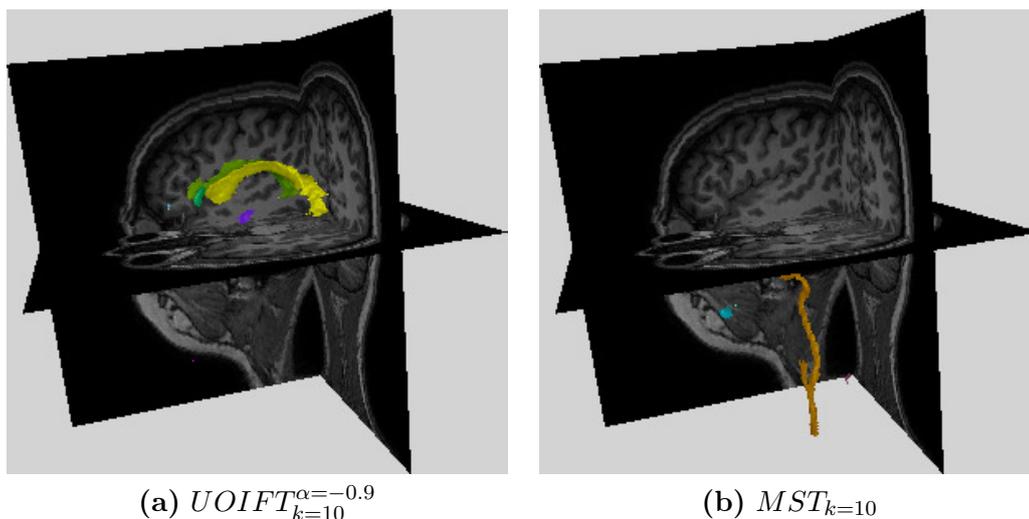


Figura 5.16: Exemplo com imagem médica 3D: (a) Segmentação 3D dos ventrículos laterais de um volume de ressonância magnética do cérebro pelo algoritmo da UOIFT é obtida após uma partição em dez regiões ($k = 10$). (b) MST falha em gerar a segmentação dos ventrículos para uma partição com o mesmo número de regiões.

As Figuras 5.17, 5.18 e 5.19 apresentam vários resultados de segmentação para imagens em escala de cinza comparando diferentes métodos. Na Figura 5.17 temos uma imagem de tomografia computadorizada (CT - *Computerized Tomography*) para segmentação do fígado. A segmentação das letras e números de uma placa de carro é mostrada na Figura 5.18. A seguir a Figura 5.19 mostra a segmentação do osso tálus em uma imagem de ressonância magnética do pé. Foi utilizado $\alpha = 0,9$ para os experimentos com imagens do fígado, devido as suas transições predominantes de claro para escuro, e $\alpha = -0,9$ para as letras da placa de carro e o osso tálus, devido as suas transições características de escuro para claro. A partir das figuras é possível observar que a UOIFT é o método que requer o menor valor de k (número de regiões da partição) para obter as regiões segmentadas desejadas.

Com relação ao tempo computacional, para uma imagem de 256×256 pixels, para calcular 625 *superpixels* pela IFT-SLIC leva 203,4 ms e o agrupamento final em 300 regiões pela UOIFT no grafo de regiões adjacentes (RAG) leva apenas 13,15 ms, em um Intel Core i3-5005U CPU @ 2,00 GHz×4.

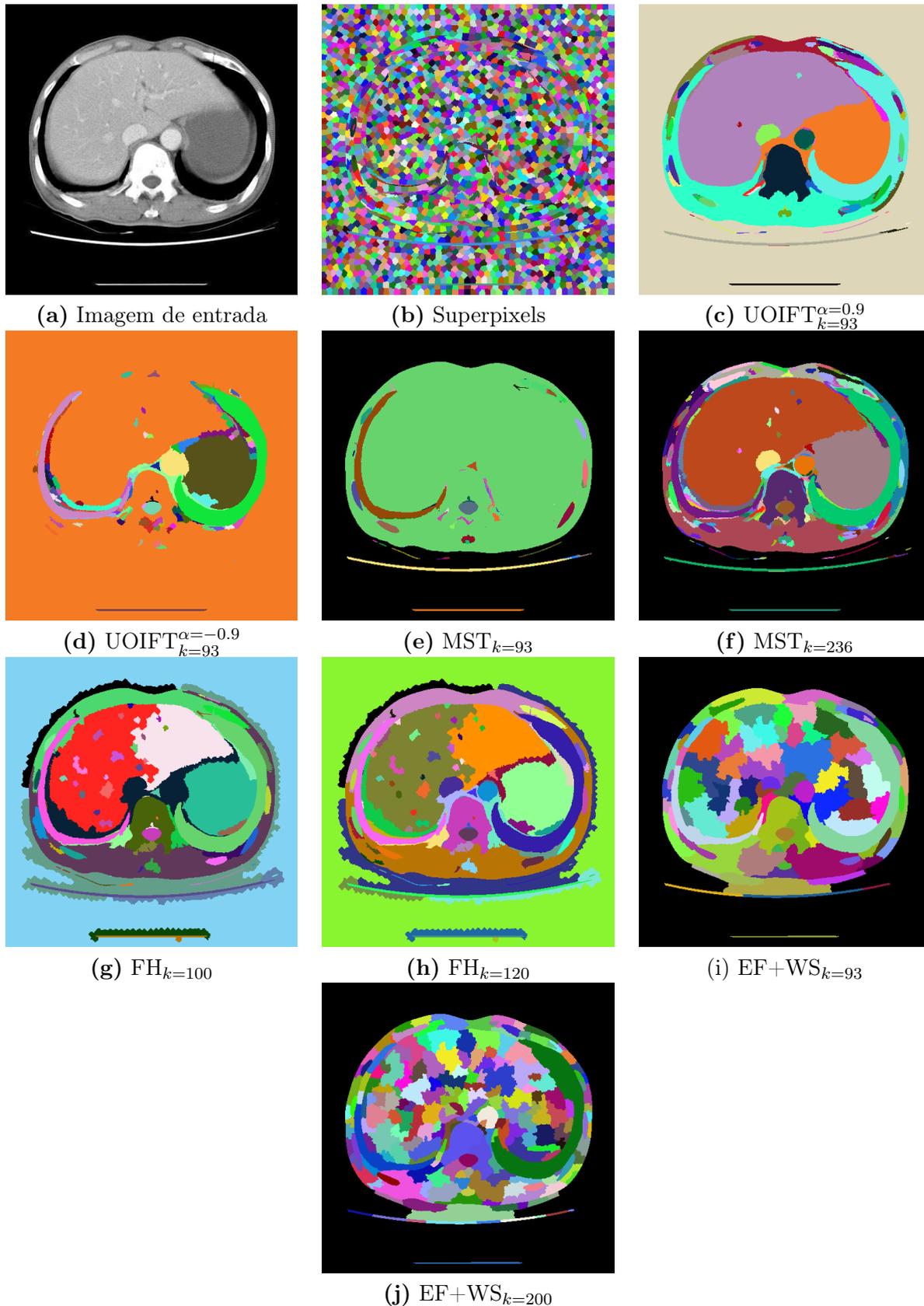


Figura 5.17: Exemplo da segmentação não supervisionada de uma fatia de um estudo torácico por tomografia computadorizada para segmentar o fígado com diferentes métodos e valores de k . Em (d), o resultado da UOIFT com polaridade de borda errada (α negativo) é mostrado que favorece a segmentação de buracos no interior das estruturas, enquanto em (c) temos sua segmentação com a polaridade correta (α positivo), favorecendo a segmentação do fígado e costelas. A fim de segmentar adequadamente o fígado, MST requer $k = 236$, ao passo que FH requer $k = 120$ e EF+WS requer $k = 200$, enquanto, neste caso, o método proposto UOIFT pode obtê-lo utilizando $k = 93$ apenas.

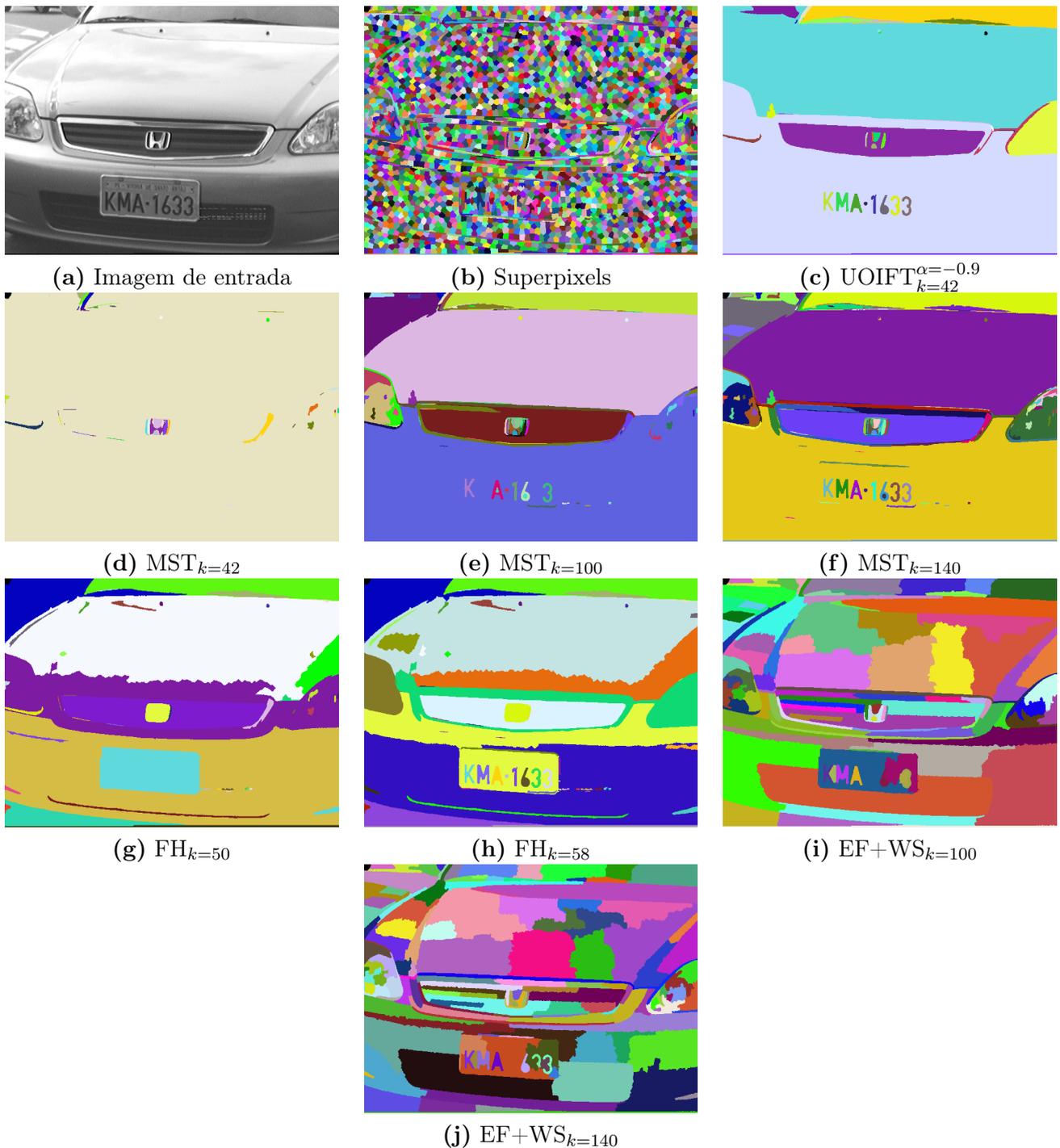


Figura 5.18: Exemplo da segmentação não supervisionada das letras e números de uma placa de carro para diferentes métodos e valores de k . A fim de segmentar adequadamente os números da placa, MST requer $k = 140$, FH $k = 58$ e $EF+WS$ requer valores maiores que $k = 140$, enquanto, neste caso, o método proposto $UOIFT$ pode obtê-lo utilizando $k = 42$ apenas.

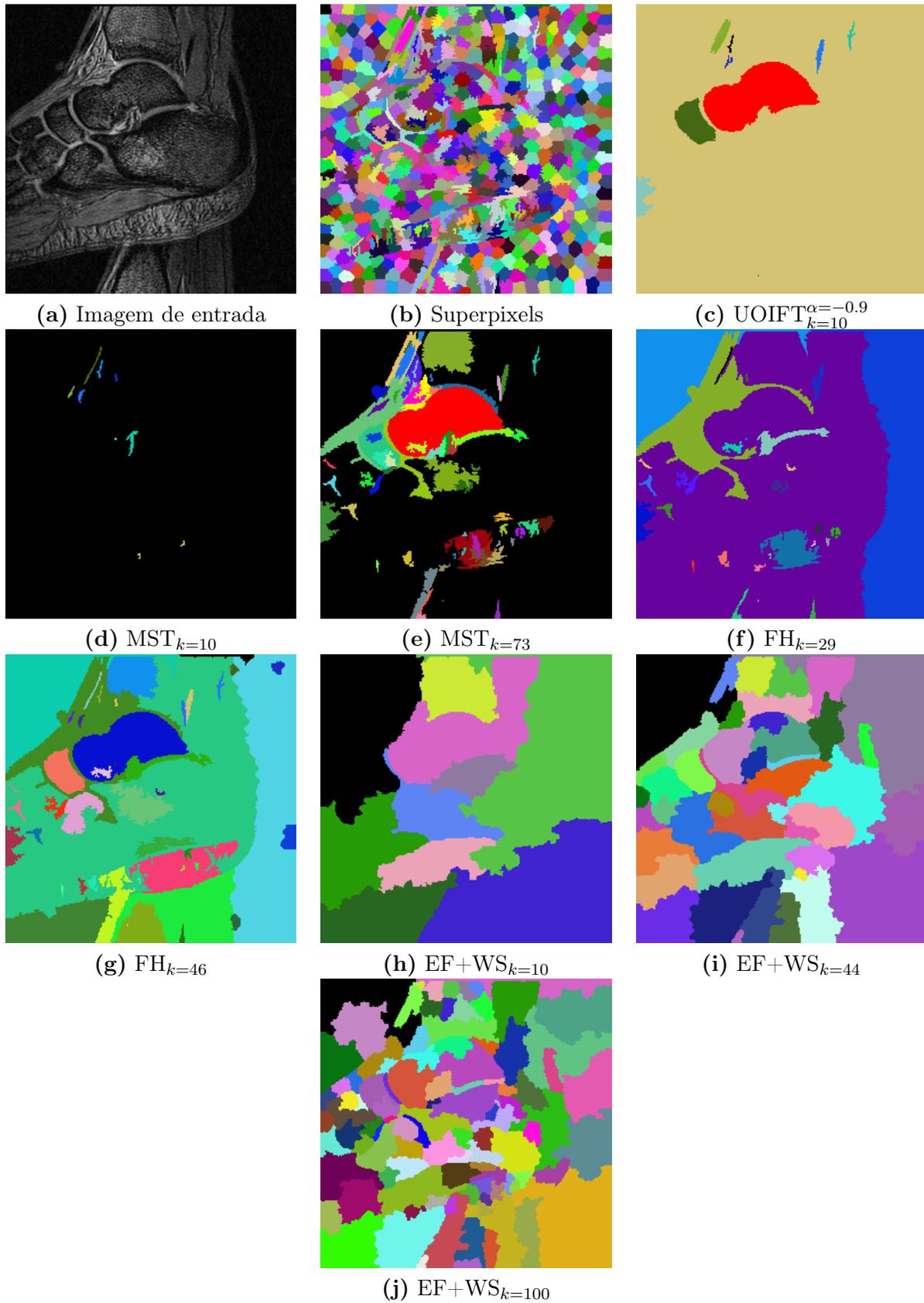


Figura 5.19: Exemplo da segmentação não supervisionada de uma imagem real de ressonância magnética do pé para diferentes métodos e valores de k . A fim de segmentar adequadamente o osso tálus, MST requer $k = 73$, FH requer $k = 46$ e $EF+WS$ requer $k = 44$, enquanto, neste caso, o método proposto $UOIFT$ pode obtê-lo utilizando $k = 10$ apenas.

5.6.2 Análise Quantitativa

Os experimentos de avaliação foram realizados utilizando 40 fatias de imagens de ressonância magnética (MRI - *Magnetic Resonance Imaging*) reais do pé para segmentar o osso tálus (Figura 5.20a) e 40 fatias de imagens de 512×512 pixels de tomografia computadorizada (CT - *Computerized Tomography*) de estudos da coluna cervical de 10 indivíduos para segmentar a vértebra espinal (Figura 5.20b).

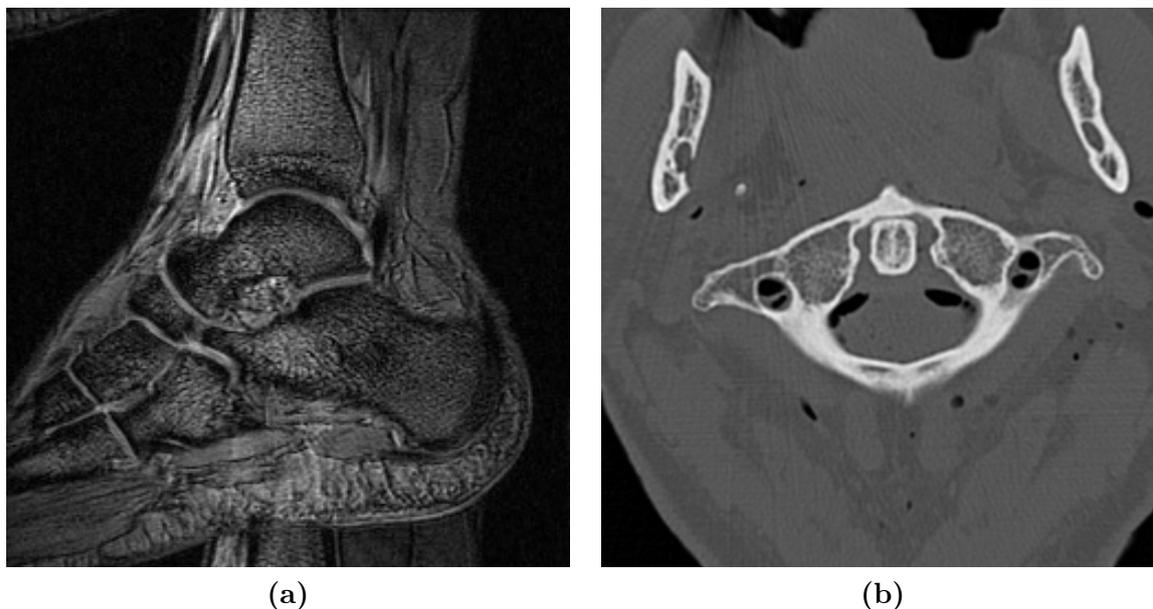


Figura 5.20: Exemplos de imagens para os experimentos quantitativos: (a) do osso tálus em imagem de ressonância magnética e (b) da vértebra espinal em imagem de tomografia computadorizada.

Foram calculadas as curvas de acurácia média para todos os métodos com diferentes valores de k (Figuras 5.21 e 5.22). Para cada valor de k , foi calculado o coeficiente de similaridade de Dice entre o gabarito e a melhor união das regiões segmentadas que gera o objeto. Também foi calculada a acurácia média utilizando o coeficiente Geral Balanceado [Cappabianco *et al.* (2019)] da Equação 2.6 (Página 24), com $\delta = 1$. Note que as diferenças entre a UOIFT e alguns outros métodos foram enfatizadas e se tornaram mais pronunciadas com o coeficiente Geral Balanceado (Figuras 5.21 e 5.22).

Visto que o método de Felzenszwalb e Huttenlocher fornece apenas um controle indireto sobre o número de regiões geradas, então no gráfico, o algoritmo de FH mostra o número médio de regiões obtidas para cada valor do seu parâmetro de entrada. Consideramos *superpixels* de tamanho $10 \times 10 = 100$ pixels para o osso tálus, enquanto *superpixels* menores de tamanho 50 pixels foram usados para a vértebra-espinal devido às suas partes finas, exceto para o NCut que requer *superpixels* maiores, por ser extremamente lento.

Embora o método $EF+WS$ possa gerar uma aproximação grosseira para os objetos no início das curvas, UOIFT é o primeiro a gerar um resultado de verdadeira acurácia elevada para o osso tálus e a vértebra-espinal. Comparado ao seu caso particular sem polaridade, dado pela MST, o método da UOIFT produz resultados consideravelmente melhores em grande parte das curvas, demonstrando a utilidade da informação de polaridade de borda. Ou seja, o método proposto requer um valor menor de k comparado com as outras abordagens para segmentar o osso do tálus e a vértebra espinal para diferentes valores de α , devido à sua informação de polaridade de borda, demonstrando a robustez da UOIFT.

Em todos os resultados quantitativos apresentados para imagens em escala de cinza, consideramos o tratamento de empates da Seção 5.4.2 (Página 66), porém é importante afirmar que os resultados sem qualquer tratamento de empate (não mostrados) foram apenas ligeiramente piores.

Figura 5.23 mostra o efeito de reduzir o tamanho dos *superpixels* para $5 \times 5 = 25$ pixels nos re-

sultados dos métodos baseados em grafo de regiões adjacentes (com exceção do NCut, que se tornou excessivamente lento nesse cenário, tornando-se inviável) nas imagens do osso tálus. Observe que para todos os métodos, foi necessário um número maior de regiões nas partições (k) para segmentar adequadamente os ossos, diferente da Figura 5.21. *Superpixels* menores fazem com que os detalhes locais da imagem sejam tratados como relevantes, consequentemente ampliando radicalmente o universo de possíveis cortes no grafo de regiões adjacentes (RAG - *Region Adjacency Graph*). Por outro lado, *superpixels* maiores do que 10×10 podem acabar misturando regiões de diferentes objetos incorretamente em um mesmo *superpixel*, o que não pode ser corrigido posteriormente pelos métodos com RAG. Portanto, um ponto de equilíbrio deve ser procurado. A Figura 5.23 também salienta o efeito da alteração de α na Equação 5.8 (Página 57) para o valor (-50%) mais próximo de zero. Observe que, nesse caso, a UOIFT se torna mais semelhante ao método MST.

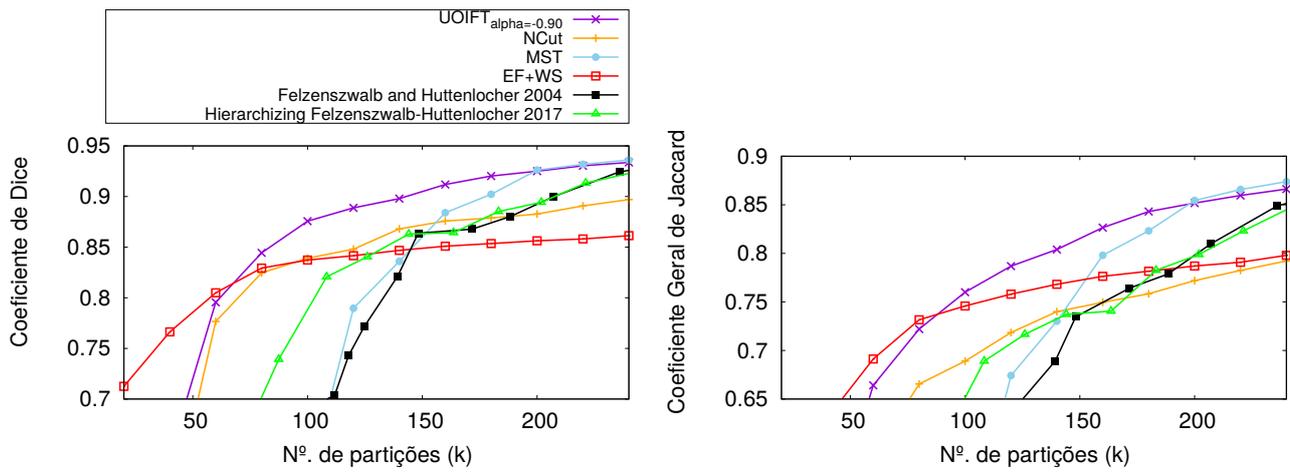


Figura 5.21: As curvas de acurácia média, pelo coeficiente de similaridade de Dice e o coeficiente generalizado de Jaccard (BG com $\delta = 1$), da melhor união das regiões produzidas para diferentes métodos e valores de k , para segmentar o osso tálus.

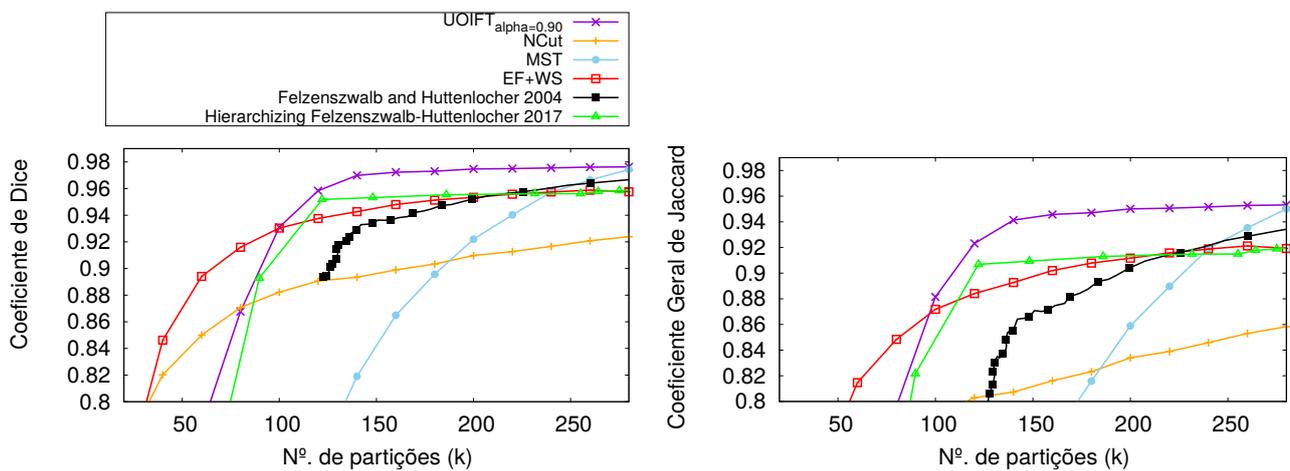


Figura 5.22: As curvas de acurácia média, pelo coeficiente de similaridade de Dice e o coeficiente generalizado de Jaccard (BG com $\delta = 1$), da melhor união das regiões produzidas para diferentes métodos e valores de k , para segmentar a vértebra espinhal.

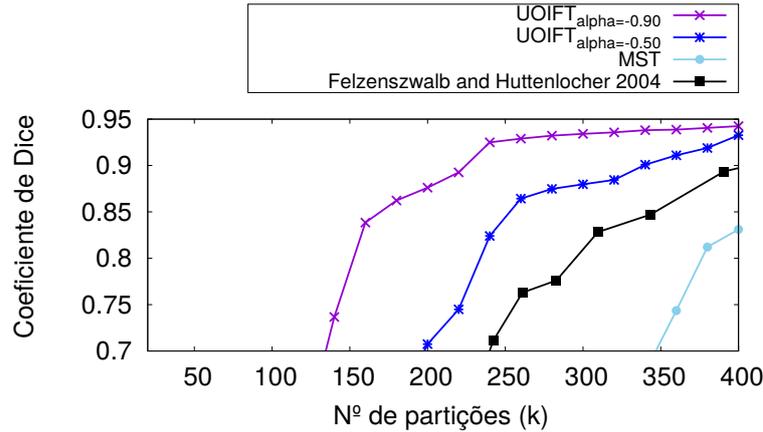


Figura 5.23: O efeito de reduzir o tamanho dos superpixels para 5×5 pixels. A curva de acurácia média, pelo coeficiente de similaridade de Dice, da melhor união das regiões produzidas para diferentes métodos e valores de k , para segmentar o osso tálus.

5.6.3 Experimentos em Imagens Coloridas

Também foram realizados experimentos qualitativos para imagens coloridas. Usamos 40 imagens naturais de um folheto de guia turístico, sujeito a diferentes condições de iluminação, para segmentar um texto em particular (Figura 5.24). As cores de referência foram selecionadas inspecionando um ponto arbitrário dentro de uma das letras e outro ponto em seu exterior na primeira imagem, de modo que as cores de referência foram fixadas nos códigos RGB hexadecimal $C1 = \#6d98ba$ e $C2 = \#b2c071$ para atribuição de peso de arco pela Equação 5.10. Além disso, foram usados *superpixels* de tamanho $10 \times 10 = 100$ pixels.

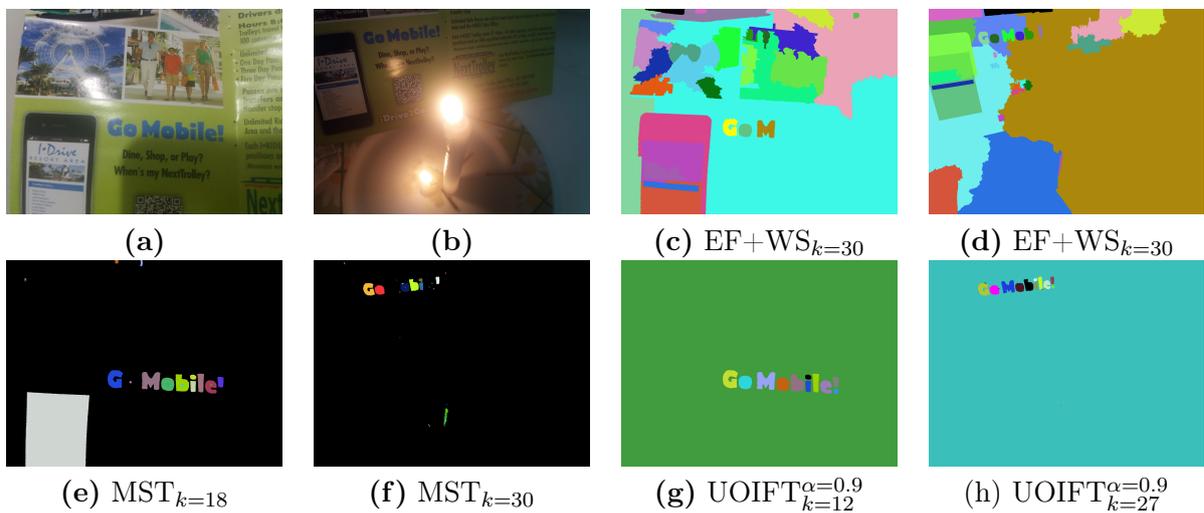


Figura 5.24: Experimentos quantitativos para imagens coloridas: (a-b) Amostra de imagens coloridas de um folheto do guia turístico sujeito a diferentes condições de iluminação para segmentar o texto “Go Mobile!”. (c-d) Resultados de EF+WS com $k = 30$, para as duas imagens. (e-f) Resultados MST com $k = 18$ e $k = 30$, respectivamente, das duas imagens. (g-h) Resultados melhorados pela UOIFT com $k = 12$ e $k = 27$, respectivamente, exigindo partições com menor número de regiões.

As curvas de acurácia média pelo coeficiente de Dice e pelo coeficiente geral de Jaccard para diferentes valores de k são apresentadas na Figura 5.25.

A UOIFT demonstrou ser robusta em relação às variações nas condições de iluminação e sombras, uma vez que se baseia principalmente no contraste das regiões. Note também que o tratamento de empates da Seção 5.4.2 (Página 66) melhorou consideravelmente os resultados.

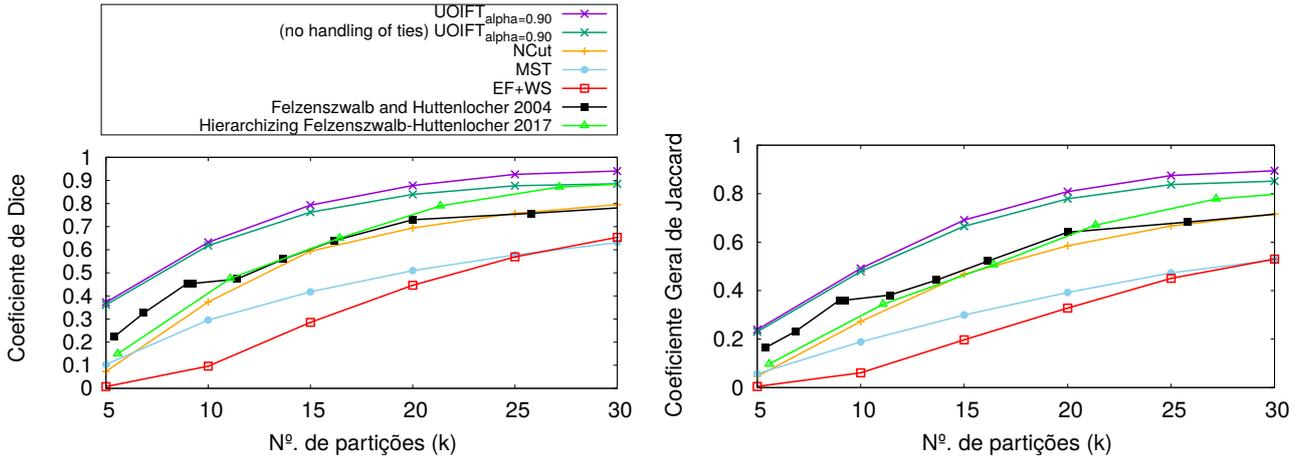


Figura 5.25: As curvas de acurácia média, pelo coeficiente de similaridade de Dice e o coeficiente geral de Jaccard (BG com $\delta = 1$), da melhor união das regiões produzidas para diferentes métodos e valores de k , para segmentar um texto específico em uma imagem colorida sujeita a diferentes condições de iluminação.

5.7 Considerações parciais

Observe que na Etapa 1 do Algoritmo 2, temos uma execução da IFT para calcular o mapa V^* . Já na Etapa 3, realizamos $k - 1$ execução da OIFT para gerar as divisões binárias sucessivas do grafo. Portanto, ao final temos um total de k execuções da IFT, sendo k o número de regiões da partição P_0 com maior nível de refinamento da hierarquia gerada $\mathcal{H} = (P_0, \dots, P_{k-1})$, sendo $k - 1$ a profundidade da hierarquia.

Como consideramos o grafo de entrada do Algoritmo 2 como um RAG de *superpixels*, no caso de uma hierarquia completa, teremos P_0 como o conjunto de $c = k$ *superpixels* (ver Seção 2.7.2), que é o maior nível de refinamento possível para este modelo de grafo. Porém, se adicionarmos a divisão a nível de pixels como sendo $P_0 = \{\{x\} : x \in \mathcal{I}\}$ e promovermos o conjunto de *superpixels* para P_1 , teremos então uma hierarquia representada por $\mathcal{H} = (P_0, P_1, \dots, P_k)$.

O Algoritmo 2 estende com sucesso a OIFT para o paradigma não supervisionado, pois cortes ótimos, via minimização da energia ε_∞ em grafos direcionados, são empregados nas divisões binárias sucessivas, ordenados pelos valores de energia dados pelo mapa V^* . Logo, temos uma resposta positiva para a Hipótese 2 levantada na Seção 1.2. Observe que a polaridade de borda dos objetos pode ser configurada facilmente pelas Equações 5.8 e 5.10, tanto para imagens tons de cinza, quanto para imagens coloridas.

No próximo capítulo, mostramos como os métodos resultantes das Hipóteses 1 e 2, tal como apresentados nos Capítulos 4 e 5, podem ser integrados em um único método, consideravelmente mais poderoso, que provê a segmentação não supervisionada com restrições de curvatura, diferenciando formas com partes côncavas e convexas, além da restrição de polaridade de borda.

Capítulo 6

Restrições de Curvatura na Transformada Imagem-Floresta Orientada Não Supervisionada

Neste capítulo, almejamos incorporar outras restrições de alto nível na geração de hierarquias, além da restrição de polaridade de borda já suportada pela UOIFT, a fim de facilitar a seleção de um objeto alvo de interesse com base na especificação das suas propriedades esperadas. Mais especificamente, mostramos como os métodos resultantes das Hipóteses 1 e 2 (página 5), tal como apresentados nos Capítulos 4 e 5, podem ser integrados em um único método, aqui denotado por UOIFT+LCF (*UOIFT with Localized Curvilinear Features*), que provê a geração de uma hierarquia com penalização de contornos com elevada curvatura, favorecendo a segmentação não supervisionada de objetos com contornos mais regulares. Adicionalmente, também mostramos que o uso de grafos direcionados, como feito pela UOIFT, nos permite distinguir partes côncavas e convexas da forma, assim viabilizando a filtragem de formas indesejadas.

A Figura 6.1 ilustra o problema a ser tratado. Na Figura 6.1a é apresentada uma imagem sintética de um objeto com variações de intensidade de claro para escuro e bordas externas serrilhadas. O algoritmo da UOIFT, com uma partição em duas regiões ($k = 2$), dá como resultado o objeto com bordas serrilhadas, devido ao seu maior contraste em relação ao fundo preto (Figura 6.1b). Mediante o emprego das técnicas aqui discutidas, é possível obter como resultado o círculo interior de menor contraste para uma mesma quantidade de regiões na partição (Figura 6.1c), através da penalização de curvatura do contorno serrilhado. Observe que ambos objetos apresentam transições de claro para escuro em sua fronteira, não sendo, portanto, possível distingui-los via a restrição de polaridade de borda.

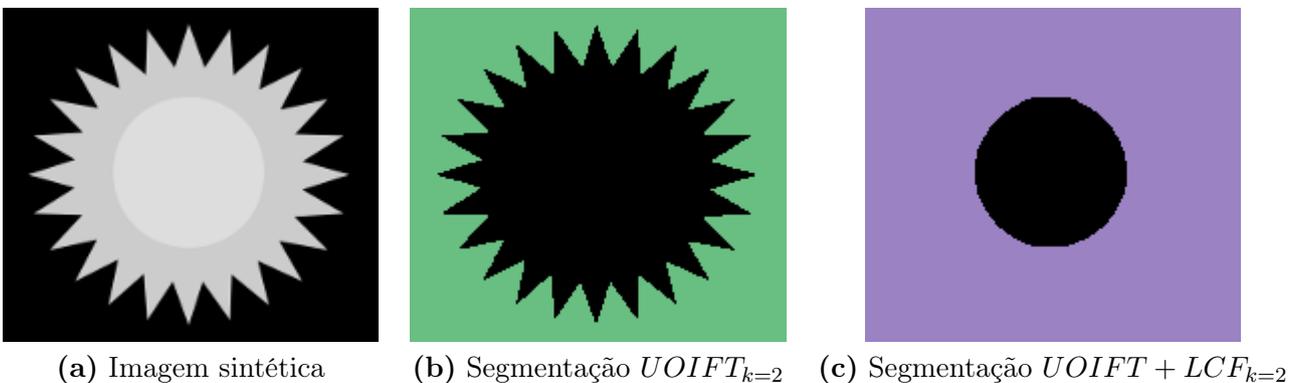


Figura 6.1: Exemplo de penalização da curvatura na segmentação não supervisionada. (a) Imagem sintética com três variações de brilho diferentes, contendo portanto dois objetos candidatos proeminentes. (b) Segmentação pelo algoritmo da UOIFT com $k = 2$. (c) Segmentação desejada com penalização de curvatura, assim filtrando o contorno serrilhado.

A dificuldade inicial encontrada para integrar adequadamente o algoritmo da UOIFT do Capítulo 5, com as técnicas previamente discutidas no Capítulo 4, se deve à diferença de resolução dos grafos empregados. Note que as Características Locais de Curvatura do Capítulo 4 foram aplicadas diretamente sobre um grafo de imagem a nível de pixels, resultando em imagens de magnitude de gradiente complementada na resolução de pixels, tal como dadas pelas Equações 4.3 e 4.4 (página 46). Ao passo que a UOIFT do Capítulo 5 foi empregada em grafos de regiões adjacentes na resolução de *superpixels*.

A seguir discutimos a solução adotada para contornar o problema.

6.1 Características Locais de Curvatura entre *Superpixels*

Os passos adotados na segmentação não supervisionada do Capítulo 5 foram:

1. Cálculo de *superpixels* (por exemplo, via IFT-SLIC).
2. Construção do RAG: Grafo de regiões adjacentes dos *superpixels* calculados no Passo 1.
3. Execução do Algoritmo 2 (UOIFT: Transformada-Imagem Floresta Orientada Não Supervisionada, página 57) sobre o grafo de regiões adjacentes.

Para adicionar as Características Locais de Curvatura (LCF - *Localized Curvilinear Features*) no processo de segmentação, foram analisadas e testadas várias alternativas, com a sua inserção em diferentes estágios do fluxo de execução. Por exemplo, o emprego prévio da LCF, em uma etapa anterior ao cálculo dos *superpixels*, requer o emprego de métodos de geração de *superpixels* que operem diretamente sobre um gradiente de imagem. Foi testado o método do *Waterpixels* [Machairas *et al.* (2014)] sobre o gradiente com penalização de curvatura pela Equação 4.3, no entanto, a qualidade dos *superpixels* obtidos ficou abaixo do esperado e outros tratamentos se faziam necessários para de fato filtrar as regiões de alta curvatura da segmentação final.

Ao final, concluiu-se que o emprego da LCF em uma etapa posterior ao cálculo dos *superpixels* é a solução mais promissora, por permitir o melhor aproveitamento das informações já computadas, melhorando inclusive a própria estimativa da curvatura (Figura 6.2):



Figura 6.2: Sequência de passos para calcular a segmentação automática com penalização de curvatura.

Dessa forma, para conseguir extrair as características de curvatura, foi preciso calcular os caminhos do *Live Wire* ao longo da borda entre cada par de *superpixels* vizinhos (Figura 6.3).

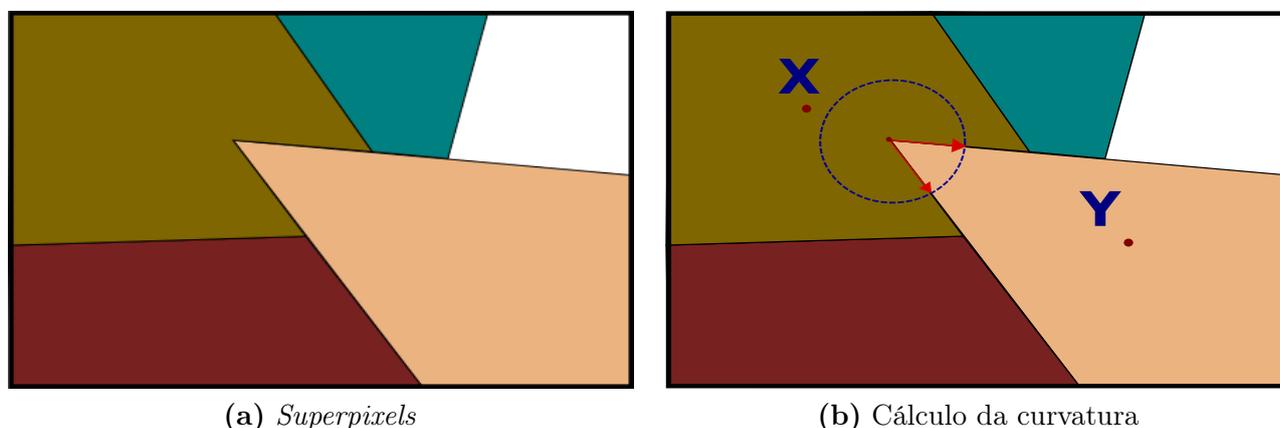


Figura 6.3: Ilustração de curvatura na fronteira de superpixels: (a) Superpixels. (b) Cálculo da curvatura na fronteira dos superpixels X e Y.

6.2 Algoritmo da UOIFT com Características Locais de Curvatura (UOIFT+LCF)

A especificação detalhada da solução proposta é apresentada a seguir com base nas seguintes definições. Primeiramente, apresentamos as definições pertinentes para a identificação da borda entre *superpixels*.

Definição 6.2.1 (Borda de um *superpixel*). A borda de um *superpixel* X é definida como:

$$B(X) = \{p \in X : \exists q \in \mathcal{I}, \text{ tal que } q \notin X \wedge \|q - p\| \leq 1\} \quad (6.1)$$

Definição 6.2.2 (Borda entre um par ordenado de *superpixels* vizinhos). A borda de um *superpixel* X em relação a um *superpixel* vizinho Y , denotada por $B_{X \rightarrow Y}$, é um subconjunto de $B(X)$, ou seja $B_{X \rightarrow Y} \subseteq B(X)$, definida por:

$$B_{X \rightarrow Y} = \{p \in X : \exists q \in Y, \text{ tal que } \|q - p\| \leq 1\} \quad (6.2)$$

Definição 6.2.3 (Borda entre um par não ordenado de *superpixels* vizinhos). A borda entre um par de *superpixels* X e Y , denotada por $B_{X,Y}$ ou $B_{Y,X}$, é a união de suas bordas ordenadas nas duas direções. Isto é:

$$B_{X,Y} = B_{Y,X} = B_{X \rightarrow Y} \cup B_{Y \rightarrow X} \quad (6.3)$$

Definição 6.2.4 (Bordas totais de *superpixels* da imagem). Dado um conjunto de *superpixels* X_0, \dots, X_{n-1} , composto por n *superpixels*, temos que o conjunto total de bordas de *superpixels* da imagem é dado por:

$$\mathcal{B} = \bigcup_{j=0}^{n-1} B(X_j) \quad (6.4)$$

sendo $B(X_j)$ a borda de um *superpixel* dada pela Definição 6.2.1.

Uma vez identificada a borda $B_{X,Y}$ entre um par de *superpixels* vizinhos X e Y , precisamos de um meio para estimar o nível de curvatura existente. Para isto, são computados caminhos pelo método do *Live Wire*, tal como explicado na Seção 4.1 (página 44), tomando como centro cada pixel c em $B_{X,Y}$. Cada caminho computado $LW(c, R)$ interliga dois pontos na borda do disco de raio R (centrado em $c \in B_{X,Y}$) e passa pelo seu centro. Porém, ao contrário da Seção 4.1, aqui a IFT empregada no cálculo dos caminhos é configurada de modo a dar preferência a caminhos que se deslocam apenas por pixels pertencentes ao conjunto \mathcal{B} (Definição 6.2.4). Caminhos passando por pixels não pertencentes a \mathcal{B} somente são computados caso não haja outra possibilidade. Esta medida melhora o desempenho nos cálculos, além de melhorar a aderência às bordas reais presentes na imagem.

Após o cálculo dos caminhos, a medida de curvatura é então estimada conforme a seguinte definição.

Definição 6.2.5 (Curvatura média entre *superpixels* vizinhos). A curvatura média entre dois *superpixels* X e Y é estimada pela média de potência (Power Mean), também conhecida como média de Hölder, pela seguinte fórmula:

$$Curv(X, Y) = \sqrt[\rho]{\frac{\sum_{c \in B_{X,Y}} [Curv(LW(c, R))]^\rho}{|B_{X,Y}|}} \quad (6.5)$$

sendo que, $Curv(LW(c, R))$ foi definida na Equação 4.2 (Página 45), ρ é o parâmetro da média de potência e $|B_{X,Y}|$ é a cardinalidade do conjunto de pixels da borda entre X e Y .

Uma vez computadas as estimativas de curvatura média entre os pares de *superpixels* vizinhos, procede-se então com a inclusão da restrição de curvatura no RAG. Assim como ocorre com a polaridade de borda, a penalização da curvatura é embutida diretamente na definição dos pesos dos arcos do RAG, viabilizando assim uma implementação eficiente do método, tal como definido abaixo.

Definição 6.2.6 (Função de custo de arco com penalização de curvatura). *Seja o grafo $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$ de regiões adjacentes (RAG), entre os superpixels vizinhos, com a função de custo de arco ω tal como definida pela Equação 5.8 (ou 5.10, página 58). Consideramos, um novo grafo $G' = \langle \mathcal{V}, \mathcal{A}, \omega' \rangle$, com uma nova função de custo de arco ω' , com penalização de curvatura, que é calculada da seguinte forma:*

$$\omega'(\langle s, t \rangle) = \omega(\langle s, t \rangle) \cdot (1 - f_{LS}(\text{Curv}(X, Y)))^\psi \quad (6.6)$$

tal que $\text{Curv}(X, Y)$ é dada pela Equação 6.5, o expoente ψ é um parâmetro usado para realçar a penalização e f_{LS} é a função de aumento linear de contraste (Linear Stretch) dada pela Equação 6.7.

$$f_{LS}(x) = \begin{cases} nmin & \text{se } 0 \leq x \leq omin \\ \frac{nmax - nmin}{omax - omin}(x - omin) + nmin & \text{se } omin < x \leq omax \\ nmax & \text{caso contrário} \end{cases} \quad (6.7)$$

Tal que, f_{LS} é usado com os seguintes parâmetros, $omin = 0,05$, $nmin = 0$, $omax = nmax = 1$, e a inclinação é $\frac{dy}{dx} = 1,0526$. Foi considerado o valor de $omin = 0,05$, a fim de converter para zero valores pequenos de curvatura gerados pela Equação 6.5, pois na prática, em função do domínio discreto da imagem, os segmentos de *Live Wire* computados sofrem de efeitos de serrilhado em direções que não sejam na horizontal ou na vertical (isto é, efeito de escada para desenhar retas nas diagonais). Logo, mesmo em regiões supostamente retas, a Equação 6.5 gera valores de curvatura não nulos, em consequência dos pequenos erros de discretização dos segmentos. Portanto, valores baixos de $\text{Curv}(X, Y)$ devem ser suprimidos, a fim de eliminar ruídos indesejados nos resultados. O valor de $omin$ foi definido com base na análise empírica de retas em diferentes orientações, medindo o valor de curvatura residual dado pela Equação 6.5, que deveria ser nulo caso o domínio da imagem fosse contínuo. Observou-se empiricamente que valores de curvatura menores ou iguais a 0,05 não devem ser levados em consideração, por serem resultado de efeitos da discretização da imagem (ruído), portanto, esses valores são estabelecidos como sendo zero (Figura 6.4).

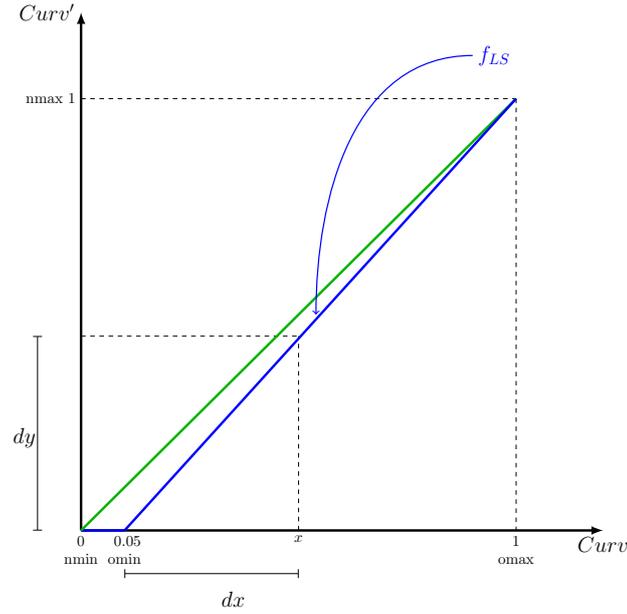


Figura 6.4: Ilustração da função de aumento linear de contraste (Linear Stretch) f_{LS} (reta azul) do valor da curvatura $Curv(X, Y)$ pela Equação 6.5 (eixo x). A função identidade (reta verde) é também mostrada para fins comparativos. Os valores x menores ou iguais a $0,05$ (o_{min}) são convertidos para zero (n_{min}) a fim de suprimir pequenos valores de curvatura gerados pela Equação 6.5, pois estes se devem a pequenos erros de discretização dos segmentos de Live Wire computados devido ao domínio discreto da imagem, não representando, portanto, curvaturas reais. Consideramos os valores máximos $o_{max} = 1$ e $n_{max} = 1$.

Por fim, temos o seguinte algoritmo para o cálculo de uma partição da imagem em k regiões, que além de suportar a polaridade de borda dos objetos, é capaz de filtrar formas indesejadas com elevada curvatura, favorecendo a seleção de objetos com contornos mais regulares.

Algoritmo 3. – ALGORITMO UOIFT+LCF

ENTRADA: Imagem $\hat{I} = (\mathcal{I}, \vec{I})$, parâmetros para estimativa de curvatura R (raio do disco circular) e ρ (Equação 6.5), parâmetro ψ para penalização da curvatura (Equação 6.6) e o número desejado de regiões k .
SAÍDA: Partição do grafo em k regiões.

1. Calcular uma imagem de *superpixels* $\hat{L} = (\mathcal{I}, L)$, tal como definido na Seção 2.7.2 (página 16), por algum método do estado da arte (e.g., IFT-SLIC [Alexandre *et al.* (2015)] ou ISF [Vargas-Muñoz *et al.* (2018)]).
2. Calcular o grafo de regiões adjacentes $G = \langle \mathcal{V}, \mathcal{A} \rangle$ a partir da imagem \hat{L} , tal como descrito na Seção 2.7.2.
3. Computar um digrafo ponderado $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$ com polaridade de borda (Equação 5.8 ou 5.10, página 58).
4. Calcular a função de peso ω' de modo a obter um novo digrafo ponderado nos arcos $G' = \langle \mathcal{V}, \mathcal{A}, \omega' \rangle$ (RAG+LCF¹), tal como descrito no Algoritmo 4.
5. Calcular a UOIFT no grafo de regiões adjacentes $G' = \langle \mathcal{V}, \mathcal{A}, \omega' \rangle$, como descrito no Algoritmo 2 (página 57), para partição do grafo em k regiões.

O Algoritmo 3 gera uma segmentação hierárquica que leva para uma partição em k regiões. A seguir é descrito o algoritmo auxiliar para calcular o grafo de regiões adjacentes com características

¹RAG+LCF - Region Adjacency Graph with Localized Curvilinear Features

locais de curvatura embutidas (RAG+LCF - *Region Adjacency Graph with Localized Curvilinear Features*).

Algoritmo 4. – ALGORITMO RAG+LCF

ENTRADA: Imagem $\hat{I} = (\mathcal{I}, \vec{I})$, imagem de *superpixels* \hat{L} , grafo de regiões adjacentes baseado nos *superpixels* $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, parâmetros para estimativa de curvatura R (raio do disco circular) e ρ (Equação 6.5), e parâmetro ψ para penalização da curvatura (Equação 6.6).
 SAÍDA: Grafo de regiões adjacentes G' com penalização de curvatura.

1. Calcular o conjunto \mathcal{B} de bordas totais dos *superpixels* da imagem (Definição 6.2.4).
2. Para todo $c \in \mathcal{B}$, calcular as características locais de curvatura $Curv(LW(c, R))$, tal que os caminhos $LW(c, R)$ são computados de modo a passar preferencialmente apenas por pixels em \mathcal{B} .
3. Para todo par de *superpixels* vizinhos X e Y , computar a curvatura média em sua borda $Curv(X, Y)$ (Definição 6.2.5).
4. Atribuir a função de custo com penalização de curvatura $\omega'(\langle s, t \rangle)$ nos arcos do grafo G' (Equação 6.6).

O Algoritmo 4 gera um grafo de regiões adjacentes com função de custo de arco ω' , que inclui a penalização de curvatura. O cálculo das características locais de curvatura restrito às bordas dos *superpixels* (isto é, os caminhos $LW(c, R)$ computados pelo *Live Wire* são calculados apenas para $c \in \mathcal{B}$) o torna mais eficiente em tempo de execução, dado que $|\mathcal{B}| \ll |\mathcal{I}|$, viabilizando o emprego de um raio R de disco circular maior do que utilizado na abordagem do Capítulo 4.

Uma comparação entre os resultados dos Algoritmos 2 e 3 é apresentada na Figura 6.5, usando como entrada a imagem sintética da Figura 6.1a. Observe que a Figura 6.1a apresenta duas bordas candidatas, ambas com transições de pixels claros para escuros no seu exterior. Logo, não é possível diferenciá-las mediante o emprego da polaridade de borda. Nas Figuras 6.5a-b são apresentados os grafos de regiões adjacentes (RAGs), G e G' , com pesos dos arcos ω e ω' , respectivamente, de modo que os pesos mais elevados são exibidos em cor branca. Os mapas de saliência das hierarquias geradas pela UOIFT e UOIFT+LCF são apresentadas nas Figuras 6.5c-d, respectivamente, enquanto que os respectivos resultados de segmentação em duas regiões ($k = 2$) são apresentados nas Figuras 6.5e-f. Note que via a penalização de curvatura da UOIFT+LCF, é possível selecionar o objeto circular interno de menor contraste, evitando o contorno irregular externo gerado pela UOIFT.

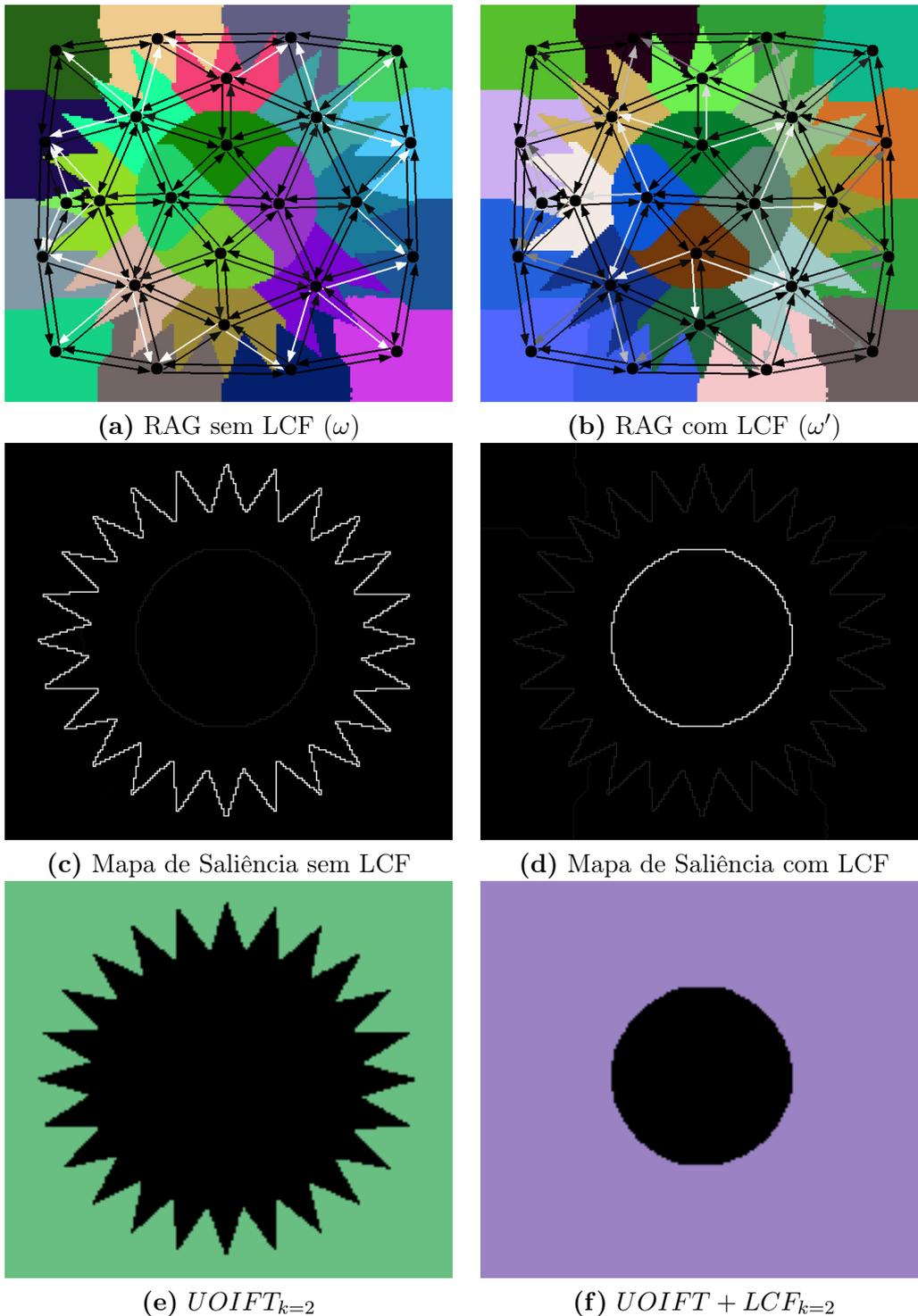


Figura 6.5: Exemplo comparativo entre a UOIFT (Algoritmo 2) e UOIFT+LCF (Algoritmo 3), a partir da imagem sintética da Figura 6.1a. Primeira coluna: UOIFT. Segunda coluna: UOIFT+LCF. (a,b) Grafos de Regiões Adjacentes com arcos entre os superpixels com pesos ω e ω' , respectivamente. (c,d) Mapas de saliência das segmentações hierárquicas resultantes. (e,f) Segmentações resultantes com partição em duas regiões ($k = 2$).

Experimentos comparativos com imagens reais entre os Algoritmos 2 e 3 são apresentados na Figura 6.6. Em todos os casos foi utilizado $\alpha = 90\%$ nas Equações 5.8 e 5.9 (página 58), para imagens tons de cinza e coloridas, respectivamente. Na primeira linha da Figura 6.6 são apresentadas as segmentações pela UOIFT ($k = 26$ regiões) e UOIFT+LCF ($k = 11$ regiões) de uma imagem de tomografia computadorizada para segmentação do fígado. Na segunda linha da Figura 6.6 são apresentadas as segmentações pela UOIFT ($k = 6$ regiões) e UOIFT+LCF ($k = 2$ regiões) de uma

imagem colorida para segmentação da placa de sinal de trânsito. Na terceira linha da Figura 6.6 são apresentadas as segmentações pela UOIFT ($k = 5$ regiões) e UOIFT+LCF ($k = 2$ regiões) de uma imagem de ressonância magnética para segmentação do osso do punho. Observe que, em todos os casos, a UOIFT+LCF obteve o objeto de interesse, utilizando um menor número de regiões na partição da imagem em comparação com a UOIFT. Isso se deve ao fato de que várias bordas indesejadas com fronteiras irregulares foram devidamente filtradas pelo emprego da nova restrição de curvatura.

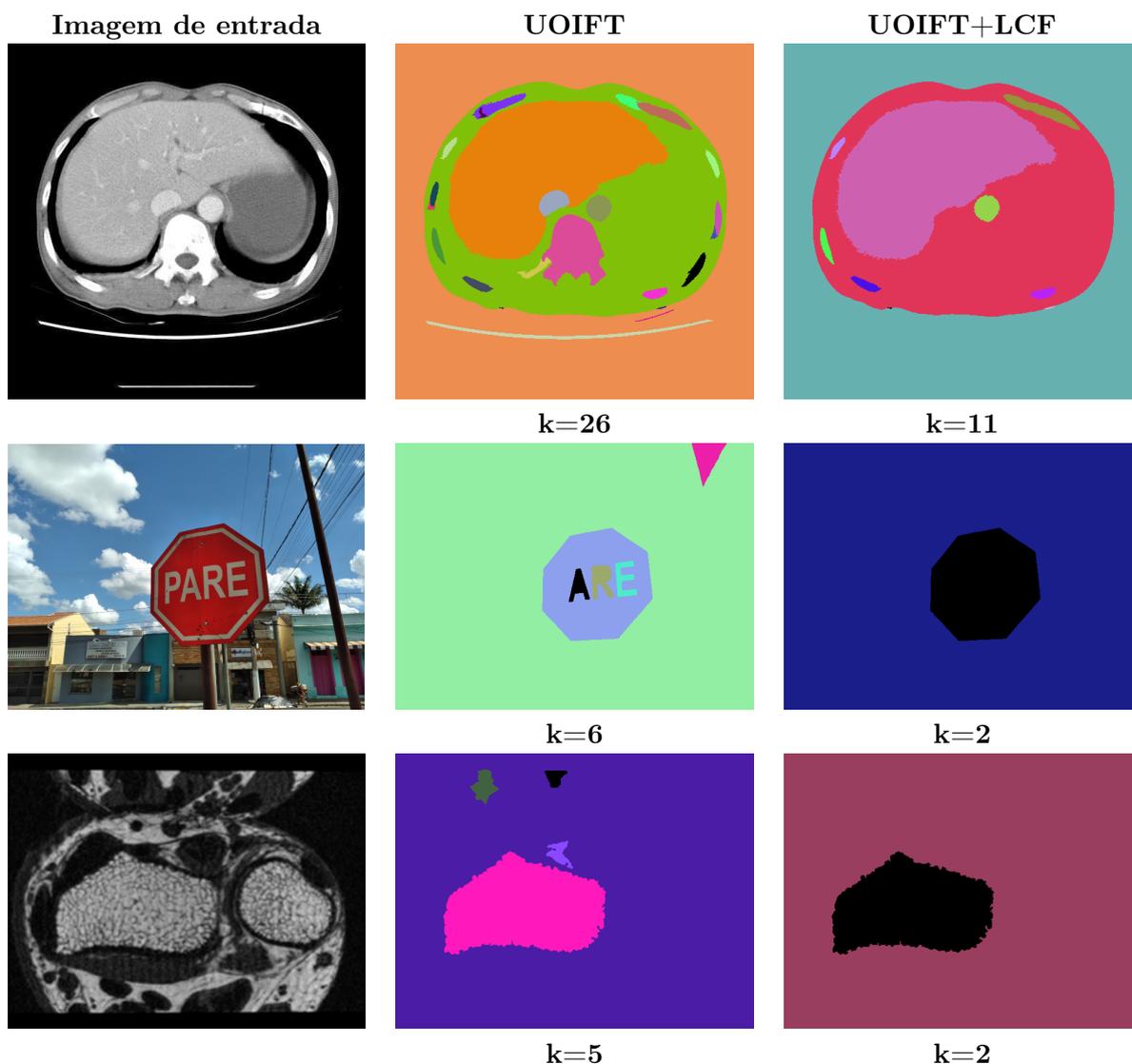


Figura 6.6: Experimentos comparativos entre a UOIFT (Algoritmo 2) e UOIFT+LCF (Algoritmo 3) em imagens reais. Primeira coluna: Imagens de entrada (imagem de tomografia computadorizado do fígado, imagem colorida de placa de sinal de trânsito e ressonância magnética do osso do punho). Segunda coluna: Segmentação via UOIFT. Terceira coluna: Segmentação via UOIFT+LCF com penalização de curvatura utilizando os parâmetros $R = 15$ pixels, $\rho = 2$ e $\psi = 3$. Abaixo das segmentações são indicados os respectivos valores do parâmetro k empregados, que foram os menores valores necessários para obter os objetos de interesse.

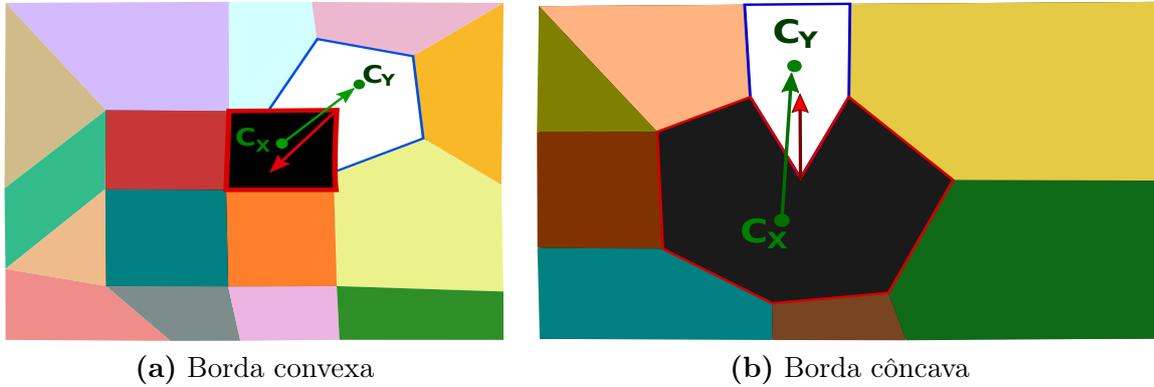
A seguir apresentamos um algoritmo que considera apenas Características Locais de Curvatura Côncava (LCCF - *Localized Concave Curvilinear Features*).

6.3 Algoritmo da UOIFT com Características Locais de Curvatura Cônica (UOIFT+LCCF)

Com relação ao algoritmo da UOIFT+LCF, observe que $Curv(X, Y)$, definido pela Equação 6.5, define uma relação simétrica, dado que $Curv(X, Y) = Curv(Y, X)$. Logo, os pesos $\omega'(\langle s, t \rangle)$, tais como definidos pela Equação 6.6, resultam em uma relação simétrica quando $\omega(\langle s, t \rangle) = \omega(\langle t, s \rangle)$. Ou seja, $\omega(\langle s, t \rangle) = \omega(\langle t, s \rangle) \implies \omega'(\langle s, t \rangle) = \omega'(\langle t, s \rangle)$. Nesta seção, mostramos que é possível diferenciar as partes côncavas e convexas da forma mediante o emprego de uma relação de curvatura assimétrica.

Figura 6.7 explica a relação assimétrica necessária para caracterizar a convexidade e concavidade das formas. Na Figura 6.7a, é apresentada a relação entre um *superpixel* X com centroide em C_X e um *superpixel* Y com centroide em C_Y . Observe que a relação de X (interior) para Y (exterior) define uma curvatura do tipo convexa. Ao passo que a relação inversa de Y (interior) para X (exterior) é do tipo côncava. Já na Figura 6.7b, o contrário se verifica, sendo a relação de X para Y côncava e de Y para X convexa.

Considerando a medida de k -curvatura (Seção 3.4.6, página 40) em sua forma vetorial, tal como apresentada nos vetores resultantes da Figura 3.9 (página 41), e considerando a sua relação com o vetor $\overrightarrow{C_X C_Y}$ que aponta de C_X para C_Y , é possível caracterizar o tipo de curvatura existente de X para Y pelo produto interno dos vetores. Na Figura 6.7a, os vetores referidos apontam em direções opostas (produto interno negativo), caracterizando uma curvatura de X para Y do tipo convexa. Já na Figura 6.7b, os vetores apontam na mesma direção (produto interno positivo), caracterizando uma curvatura de X para Y do tipo côncava. Portanto, para todos pares de *superpixels* vizinhos, o cálculo do produto interno entre o vetor interligando os centroides dos *superpixels* e o vetor de curvatura, nos permite identificar se o segmento de curvatura em questão é convexo ou côncavo.



(a) Borda convexa

(b) Borda côncava

Figura 6.7: Exemplo de dois casos de curvatura na borda entre dois *superpixels* vizinhos X e Y , tal que C_X e C_Y são seus respectivos centroides. O vetor $\overrightarrow{C_X C_Y}$ que aponta de C_X para C_Y é mostrado na cor verde e o vetor curvatura em vermelho. (a) Borda convexa do *superpixel* X para Y . (b) Borda côncava do *superpixel* X para Y .

Nos algoritmos apresentados serão utilizadas as seguintes definições:

Definição 6.3.1 (Centroide do *superpixel*). O centroide C_X de um *superpixel* $X = \{p_1, p_2, \dots, p_n\}$, composto pelos *pixels* $p_i = (x_i, y_i)$, $i = 1, \dots, n$, é definido como:

$$C_X = (\bar{x}_X, \bar{y}_X) = \frac{1}{|X|} \left(\sum_{p_i \in X} x_i, \sum_{p_i \in X} y_i \right) \quad (6.8)$$

tal que $|X| = n$ é a cardinalidade do *superpixel*.

A medida de curvatura da Equação 4.2 (página 45) computa um valor médio das magnitudes dos vetores resultantes de curvatura normalizados dados pela Equação 3.23 (página 41). A fim de diferenciar curvaturas côncavas e convexas, a seguir definimos o vetor médio de curvatura.

Definição 6.3.2 (Vetor médio de curvatura). O vetor médio de curvatura $\overrightarrow{Curv}(LW(c, R)) = (Curv_x, Curv_y)$, é definido por:

$$\overrightarrow{Curv}(LW(c, R)) = \overrightarrow{Curv}(\langle p_1, \dots, p_k = c, \dots, p_m \rangle) = \frac{1}{(L/2)} \cdot \sum_{i=L/2}^{L-1} \frac{p_{k-i} - 2 \cdot c + p_{k+i}}{\|p_{k-i} - c\| + \|c - p_{k+i}\|} \quad (6.9)$$

em que $L = \min\{k - 1, m - k\} + 1$ e $LW(c, R)$ é o caminho computado pelo Live Wire, tal como explicado na Seção 4.1 (página 44), passando pelo pixel central $c \in \mathcal{B}$.

Definição 6.3.3 (Vetor de curvatura média côncava entre superpixels vizinhos). O vetor de curvatura média côncava entre superpixels vizinhos é definido como:

$$\overrightarrow{Curv}(X, Y) = \begin{cases} 0 & \text{se } B_c(X, Y) = \emptyset \\ \frac{\sum_{c \in B_c(X, Y)} \overrightarrow{Curv}(LW(c, R))}{|B_c(X, Y)|} & \text{caso contrário} \end{cases} \quad (6.10)$$

tal que, o conjunto $B_c(X, Y)$ inclui pixels que contribuem para a caracterização de curvatura côncava:

$$B_c(X, Y) = \{c \in B_{X, Y} : \overrightarrow{Curv}(LW(c, R)) \cdot \overrightarrow{XY} > 0\} \quad (6.11)$$

sendo que $\overrightarrow{Curv}(LW(c, R)) \cdot \overrightarrow{XY}$ é o produto interno entre o vetor \overrightarrow{XY} e o vetor curvatura $\overrightarrow{Curv}(LW(c, R))$, e \overrightarrow{XY} é o vetor que interliga os centroides dos superpixels X e Y da seguinte forma:

$$\overrightarrow{XY} = C_Y - C_X = (\bar{x}_Y, \bar{y}_Y) - (\bar{x}_X, \bar{y}_X) = (\bar{x}_Y - \bar{x}_X, \bar{y}_Y - \bar{y}_X) \quad (6.12)$$

tal que $C_X = (\bar{x}_X, \bar{y}_X)$ é o centroide de X e $C_Y = (\bar{x}_Y, \bar{y}_Y)$ é o centroide de Y .

Definição 6.3.4 (Função de custo de arco com penalização de curvatura côncava). Seja o grafo $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$ de regiões adjacentes (RAG), entre os superpixels vizinhos, com a função de custo de arco ω tal como definida pela Equação 5.8 (ou 5.10, página 58). Consideramos, um novo grafo $G'' = \langle \mathcal{V}, \mathcal{A}, \omega'' \rangle$, com uma nova função de custo de arco ω'' , com penalização de curvatura côncava, que é calculada da seguinte forma:

$$\omega''(\langle s, t \rangle) = \omega(\langle s, t \rangle) \cdot \left(1 - f_{LS}(\|\overrightarrow{Curv}(X, Y)\|)\right)^\psi \quad (6.13)$$

tal que, $\|\cdot\|$ é a norma do vetor $\overrightarrow{Curv}(X, Y)$ dada pela Equação 6.10, o expoente ψ é usado para realçar a penalização, e f_{LS} é a função de aumento linear de contraste (Linear Stretch) dada pela Equação 6.7.

Por fim, temos o seguinte algoritmo, aqui denotado por UOIFT+LCCF (*UOIFT with Localized Concave Curvilinear Features*), para o cálculo de uma partição da imagem em k regiões, que além de suportar a polaridade de borda dos objetos, é capaz de filtrar formas indesejadas com curvaturas do tipo côncava elevadas, porém sem penalizar curvaturas do tipo convexa.

Algoritmo 5. – ALGORITMO UOIFT+LCCF

ENTRADA: Imagem $\hat{I} = (\mathcal{I}, \vec{I})$, parâmetro para estimativa de curvatura R (raio do disco circular), parâmetro ψ para penalização da curvatura (Equação 6.13) e o número desejado de regiões k .
SAÍDA: Partição do grafo em k regiões.

1. Calcular uma imagem de *superpixels* $\hat{L} = (\mathcal{I}, L)$, tal como definido na Seção 2.7.2 (página 16), por algum método do estado da arte (e.g., IFT-SLIC [Alexandre *et al.* (2015)] ou ISF [Vargas-Muñoz *et al.* (2018)]).
2. Calcular o grafo de regiões adjacentes $G = \langle \mathcal{V}, \mathcal{A} \rangle$ a partir da imagem \hat{L} , tal como descrito na Seção 2.7.2.
3. Computar um digrafo ponderado $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$ com polaridade de borda (Equação 5.8 ou 5.10, página 58).
4. Calcular a função de peso ω'' de modo a obter um novo digrafo ponderado nos arcos $G'' = \langle \mathcal{V}, \mathcal{A}, \omega'' \rangle$ (RAG+LCCF²), tal como descrito no Algoritmo 6.
5. Calcular a UOIFT no grafo de regiões adjacentes $G'' = \langle \mathcal{V}, \mathcal{A}, \omega'' \rangle$, como descrito no Algoritmo 2 (página 57), para partição do grafo em k regiões.

O Algoritmo 5 gera uma segmentação hierárquica que leva para uma partição em k regiões. A seguir é descrito o algoritmo auxiliar para calcular o grafo de regiões adjacentes com características locais de curvatura côncava embutidas (RAG+LCCF - *Region Adjacency Graph with Localized Concave Curvilinear Features*).

Algoritmo 6. – ALGORITMO RAG+LCCF

ENTRADA: Imagem $\hat{I} = (\mathcal{I}, \vec{I})$, imagem de *superpixels* \hat{L} , grafo de regiões adjacentes baseado nos *superpixels* $G = \langle \mathcal{V}, \mathcal{A}, \omega \rangle$, parâmetro para estimativa de curvatura R (raio do disco circular) e parâmetro ψ para penalização da curvatura (Equação 6.6).
SAÍDA: Grafo de regiões adjacentes G'' com penalização de curvatura côncava.

1. Calcular os centroides C_{X_j} dos *superpixels* X_j , $j = 0, \dots, c - 1$ (ver Seção 2.7.2), tal como definido na Equação 6.8.
2. Calcular o conjunto \mathcal{B} de bordas totais dos *superpixels* da imagem (Definição 6.2.4).
3. Para todo $c \in \mathcal{B}$, calcular as características locais de curvatura na forma vetorial $\overrightarrow{Curv}(LW(c, R))$ (Equação 6.9), tal que os caminhos $LW(c, R)$ são computados de modo a passar preferencialmente apenas por pixels em \mathcal{B} .
4. Para todo par de *superpixels* vizinhos X e Y , computar o vetor de curvatura média côncava em sua borda $\overrightarrow{Curv}(X, Y)$ (Equação 6.10).
5. Atribuir a função de custo com penalização de curvatura côncava $\omega''((s, t))$ nos arcos do grafo G'' (Equação 6.13).

O Algoritmo 6 gera um grafo de regiões adjacentes com função de custo de arco ω'' , que inclui a penalização de curvaturas do tipo côncava. O domínio de execução do *Live Wire* restrito as bordas dos *superpixels* (isto é, $c \in \mathcal{B}$) para calcular as características de curvatura foi reduzido enormemente, dado que $|\mathcal{B}| \ll |\mathcal{I}|$, e o caminho $LW(c, R)$ calculado via *Live Wire* no disco circular

²RAG+LCCF - *Region Adjacency Graph with Localized Concave Curvilinear Features*

foi simplificado, dando preferência a caminhos que se deslocam apenas por pixels pertencentes ao conjunto \mathcal{B} . Com isso, foi possível aumentar o raio do disco circular utilizado (por exemplo, $R = 15$ pixels é utilizado em todos exemplos apresentados), o que permitiu melhorar a estimativa de curvatura em comparação à abordagem do Capítulo 4.

A Figura 6.8a apresenta uma imagem sintética que contém três objetos com diferentes níveis de contraste, sendo o objeto com partes côncavas o de maior contraste (objeto à direita), seguido pelo triângulo e finalmente o círculo. O objetivo é mostrar os efeitos da penalização de curvaturas do tipo côncava. A Figura 6.8b apresenta a imagem de *superpixels* gerada na primeira etapa do Algoritmo 5, aqui empregando a IFT-SLIC [Alexandre *et al.* (2015)].

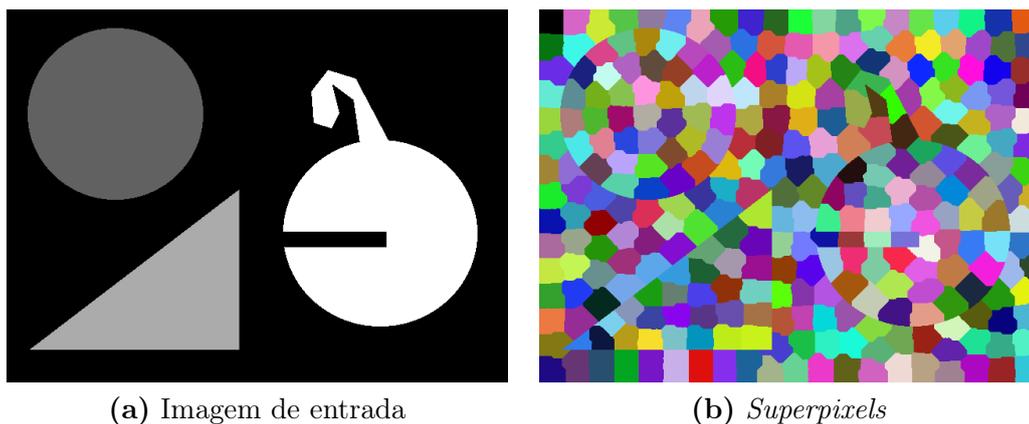


Figura 6.8: Exemplo de objetos com diferentes curvaturas. (a) Imagem sintética. (b) Imagem dividida em superpixels via IFT-SLIC [Alexandre *et al.* (2015)].

Nas Figuras 6.9, 6.10 e 6.11 são ilustrados os grafos de regiões adjacentes, correspondentes aos algoritmos da UOIFT, UOIFT+LCF e UOIFT+LCCF, respectivamente. Foi utilizada a polaridade de borda $\alpha = 90\%$ da Equação 5.8 em todos os casos, porém observe que ela não nos permite diferenciar os objetos, dado que todos apresentam transições do tipo claro para escuro.

Por exemplo, na Figura 6.9 é apresentado o grafo de regiões adjacentes sem penalização de curvatura, em que prevalece o objeto de maior contraste (com formato de Pac-Man). Observe os arcos mais claros em sua borda de corte (Equação 3.8, página 32).

Já na Figura 6.10 é apresentado o grafo de regiões adjacentes com penalização de curvatura, em que a energia de corte (Equação 3.10, página 32) do círculo prevalece. Isso ocorre, pois os outros dois objetos são penalizados por apresentarem curvaturas acentuadas ao longo dos seus contornos, portanto seus arcos nestas regiões de alta curvatura são exibidos em tonalidade mais escura.

Finalmente, na Figura 6.11 é apresentado o grafo de regiões adjacentes com penalização apenas de curvaturas côncavas, tal que a energia de corte (Equação 3.10) do triângulo é a mais elevada.

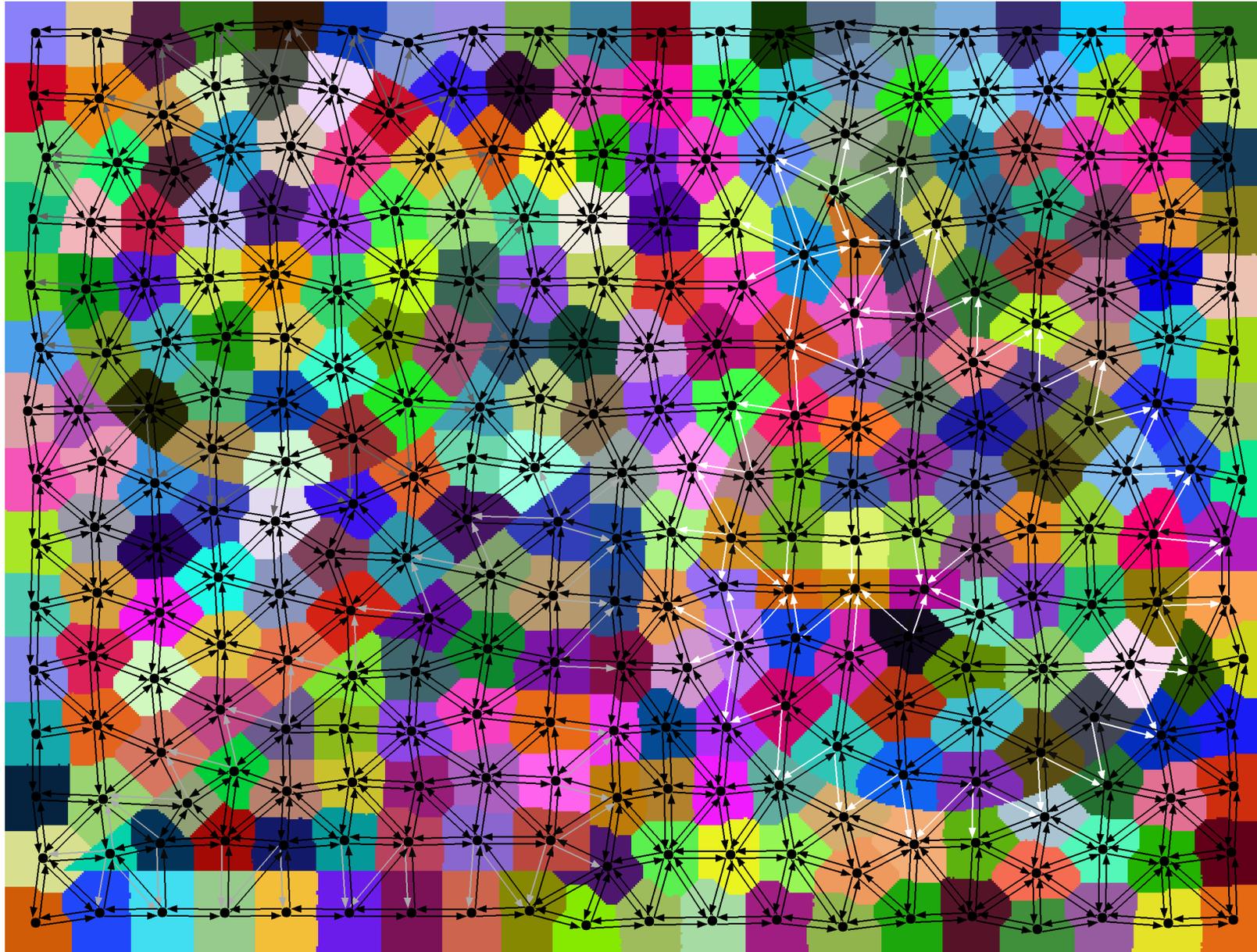


Figura 6.9: Grafo de Regiões Adjacentes (RAG) sem penalização de curvatura (com pesos $\omega((s,t))$ computados pela Equação 5.8 com $\alpha = 90\%$), a partir da imagem sintética da Figura 6.8a, utilizando os superpixels da Figura 6.8b. Os arcos mais claros em sua borda de corte são os do objeto com formato de Pac-Man, por apresentar maior contraste.

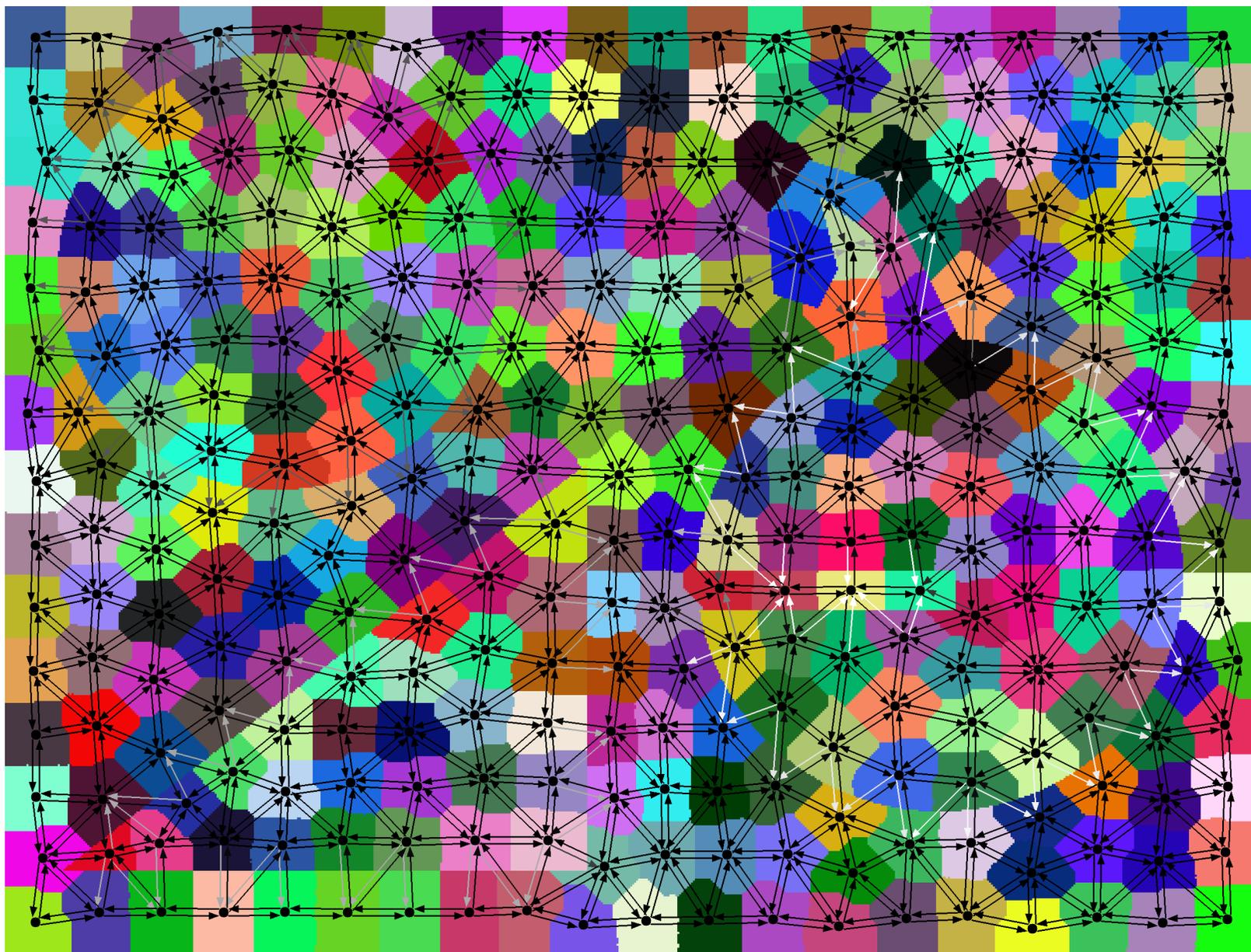


Figura 6.10: *Grafo de Regiões Adjacentes com penalização de curvatura (RAG+LCF), com pesos $\omega'(\langle s, t \rangle)$ computados pela Equação 6.6 (parâmetros $R = 15$ pixels, $\rho = 2$ e $\psi = 3$), a partir da imagem sintética da Figura 6.8a, utilizando os superpixels da Figura 6.8b. Os arcos nas regiões de alta curvatura são exibidos em tonalidade mais escura.*

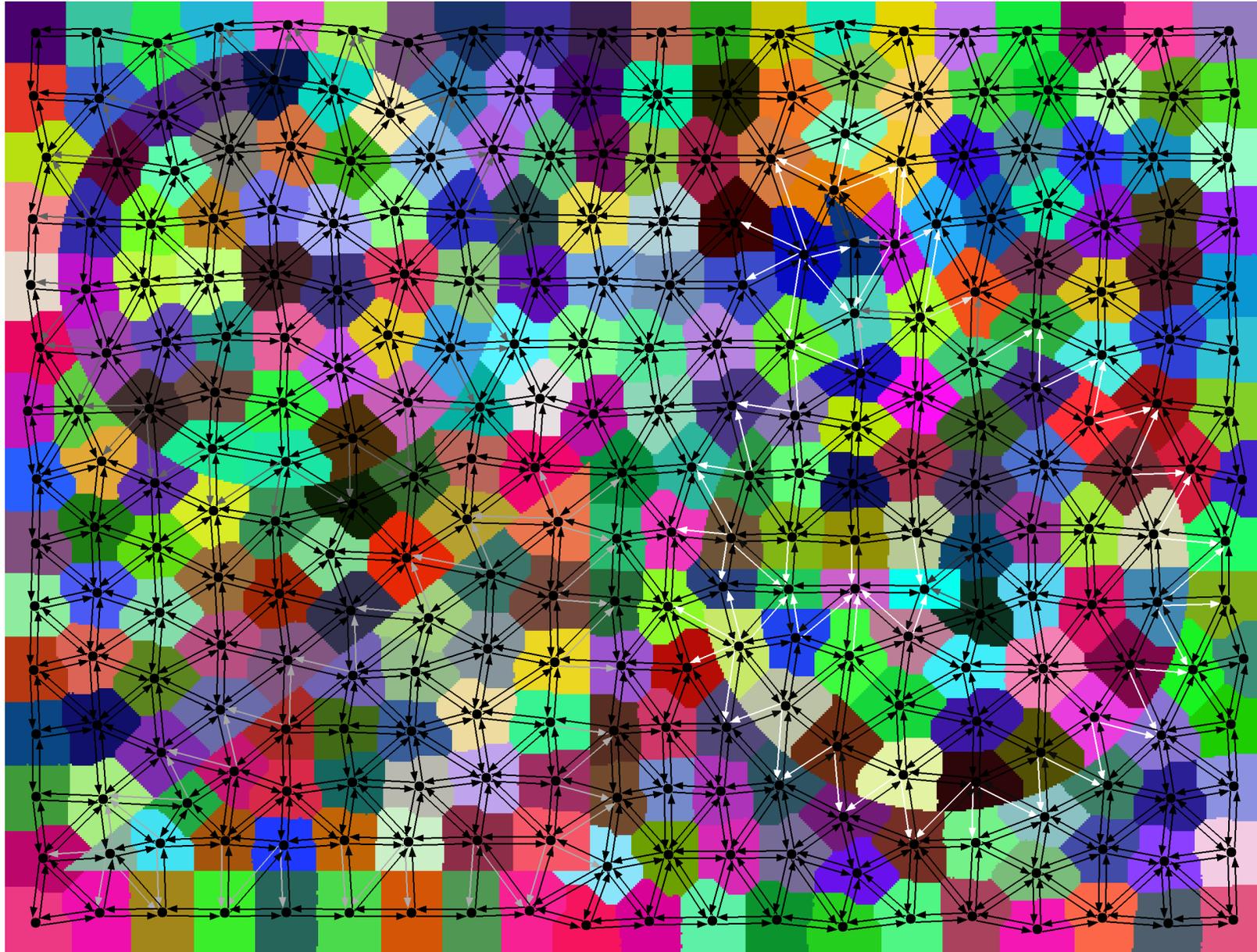


Figura 6.11: Grafo de Regiões Adjacentes com penalização apenas de curvaturas côncavas (RAG+LCCF), com pesos $\omega''(\langle s, t \rangle)$ computados pela Equação 6.13 (parâmetros $R = 15$ pixels, $\rho = 2$ e $\psi = 3$), a partir da imagem sintética da Figura 6.8a, utilizando os superpixels da Figura 6.8b. A energia de corte do triângulo é a mais elevada.

Por fim, mostramos na Figura 6.12, a comparação entre os resultados de segmentação da imagem sintética da Figura 6.8a obtidos pelos Algoritmos 2, 3 e 5. Os mapas de saliência das hierarquias geradas pela UOIFT, UOIFT+LCF e UOIFT+LCCF são apresentados na primeira linha da Figura 6.12, respectivamente, enquanto que os resultados de segmentação em duas regiões ($k = 2$) são apresentados na segunda linha da Figura 6.12. Note que via UOIFT+LCF os objetos com alta curvatura são penalizados (objeto com formato de Pac-Man e o triângulo). Por outro lado, via UOIFT+LCCF apenas o objeto com formato de Pac-Man é penalizado, beneficiando o segundo objeto com maior contraste, que é o triângulo.

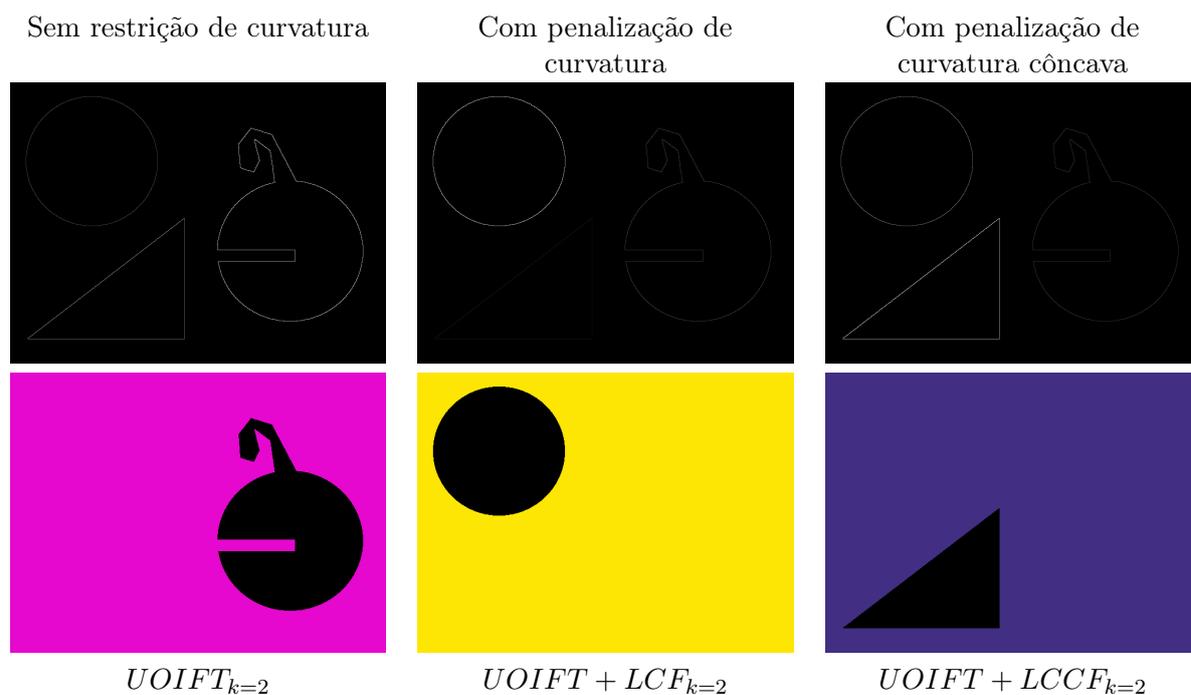


Figura 6.12: Exemplo comparativo entre a UOIFT (Algoritmo 2), UOIFT+LCF (Algoritmo 3) e UOIFT+LCCF (Algoritmo 5), a partir da imagem sintética da Figura 6.8a. Primeira linha: Mapas de saliência das segmentações hierárquicas resultantes. Segunda linha: Segmentações resultantes com partição em duas regiões ($k = 2$).

A fim de favorecer uma melhor compreensão dos resultados apresentados na Figura 6.12, uma análise mais detalhada dos pesos dos arcos nas regiões chaves dos objetos é apresentada na Figura 6.13, comparando os três grafos de regiões adjacentes (RAG, RAG+LCF e RAG+LCCF). Na primeira linha da Figura 6.13, são indicadas as regiões chaves dos objetos que serão ampliadas (zoom). Para as demais linhas, são mostrados os respectivos arcos chaves de corte $\langle p, q \rangle$, nas diferentes regiões, apontando do objeto para o fundo. Na segunda linha da Figura 6.13 (que corresponde ao RAG normal sem penalização de curvatura), o arco $\langle p, q \rangle$ exibido com maior peso é o do objeto com formato de Pac-Man. Na terceira linha da Figura 6.13 (que corresponde ao RAG+LCF), o arco $\langle p, q \rangle$ exibido com maior peso é o do círculo. Já na quarta linha da Figura 6.13 (que corresponde ao RAG+LCCF), o arco $\langle p, q \rangle$ exibido com maior peso é o do triângulo.

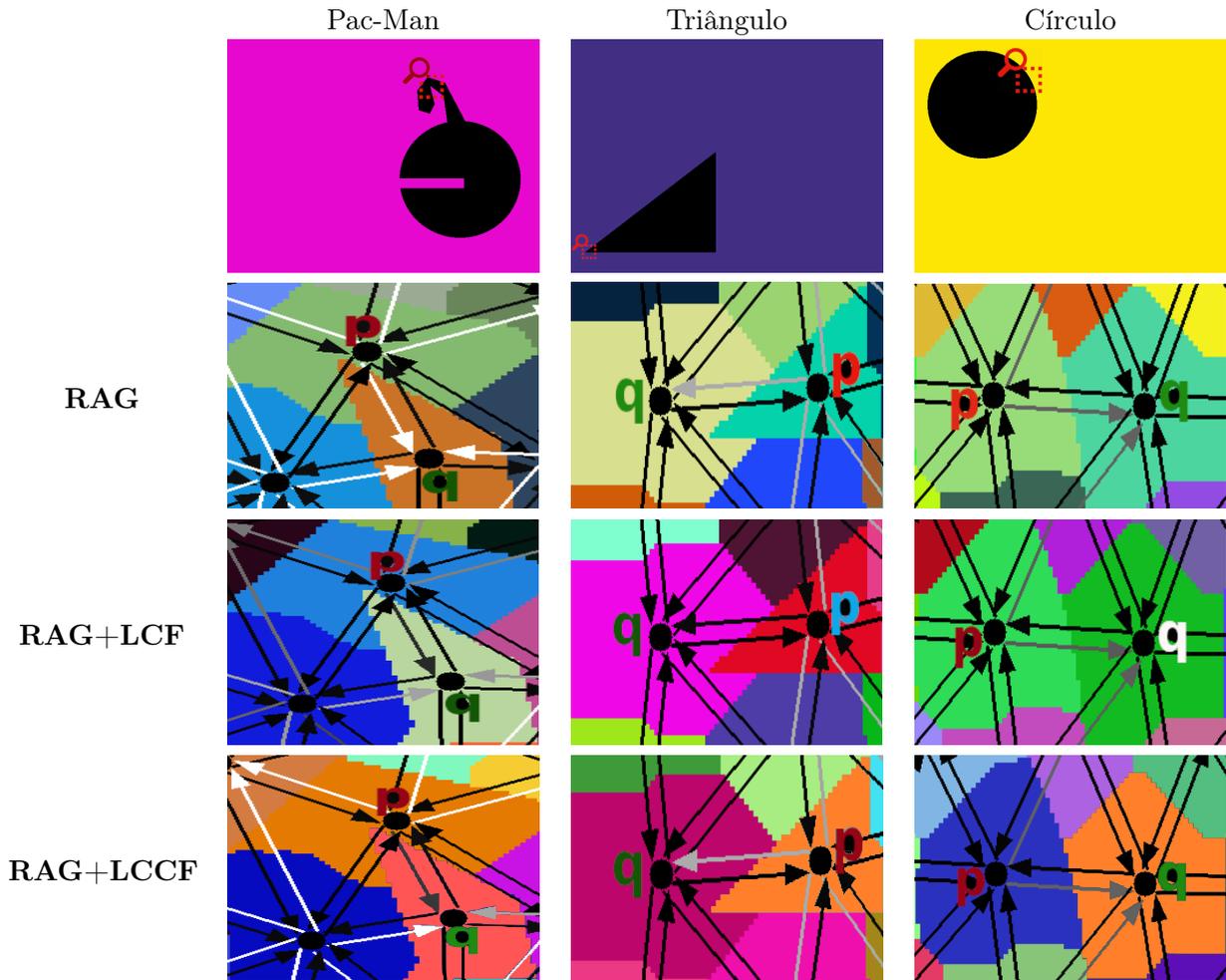


Figura 6.13: Comparação dos pesos dos arcos $\langle p, q \rangle$ nos grafos de regiões adjacentes (RAG, RAG+LCF e RAG+LCCF) dos objetos da imagem sintética da Figura 6.8a, com p no interior e q no exterior dos objetos. Primeira coluna: Variação do peso do arco $\langle p, q \rangle$ no objeto com formato de Pac-Man. Segunda coluna: Variação do peso do arco $\langle p, q \rangle$ no triângulo. Terceira coluna: Variação do peso do arco $\langle p, q \rangle$ no círculo.

Experimentos com imagens reais coloridas são apresentados nas Figuras 6.14 e 6.15. Em todos os casos, foi utilizado $\alpha = 90\%$ na Equação 5.9 (página 58).

No primeiro experimento, particionamos a Figura 6.14a em diferentes números de regiões, com parâmetro $k \in \{2, 5\}$, fazendo uma análise comparativa entre os Algoritmos 2, 3 e 5. Na coluna da UOIFT, ou seja, sem penalização de curvatura, o objeto selecionado é o garfo de brinquedo. Na coluna da UOIFT+LCF, as peças com formas mais regulares são beneficiadas. Já na coluna da UOIFT+LCCF, o objeto giz de cera é selecionado por não ter tido suas partes convexas penalizadas. Observe que, o objeto com maior contraste é o garfo, tal que é segmentado pela UOIFT. Por outro lado, a UOIFT+LCF seleciona os objetos com formas mais regulares. Por fim, a UOIFT+LCCF não seleciona o garfo, dado que este foi penalizado por ter segmentos de curvatura côncavas.

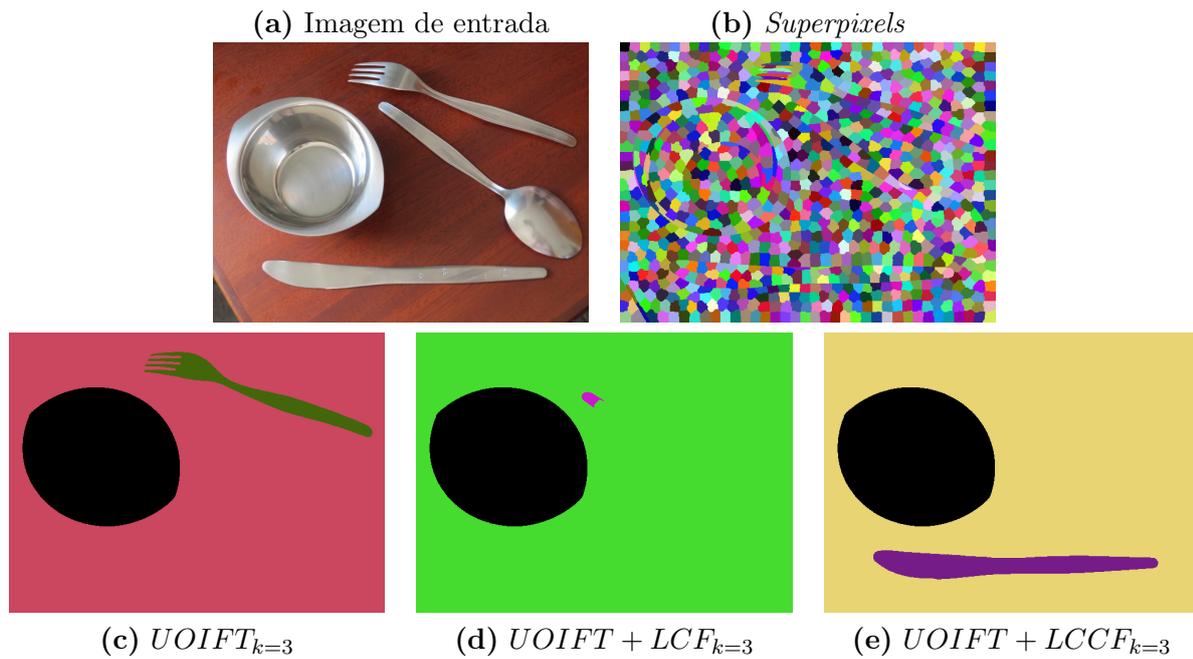


Figura 6.15: Experimento comparativo entre a UOIFT (Algoritmo 2, página 57), UOIFT+LCF (Algoritmo 3) e UOIFT+LCCF (Algoritmo 5) em uma imagem real. (a) Imagem de entrada. (b) Imagem dividida em superpixels (IFT-SLIC [Alexandre et al. (2015)]). (c-e) Resultados de segmentação de talheres e tigela para partição em três regiões ($k = 3$), com os parâmetros $R = 15$ pixels, $\rho = 2$ e $\psi = 3$.

Capítulo 7

Conclusões

Neste capítulo discutimos as considerações finais do nosso trabalho, assim como potenciais trabalhos futuros.

7.1 Considerações finais

Neste trabalho, o algoritmo da OIFT foi estendido com sucesso para realizar a segmentação não supervisionada (UOIFT - *Unsupervised Oriented Image Foresting Transform*) para imagens em tons de cinza, imagens coloridas e volumes 3D. UOIFT também pode ser vista como uma extensão da versão *top-down* do algoritmo *Single-Linkage* pela MST para grafos dirigidos a fim de explorar as informações de polaridade de borda dos objetos.

O Algoritmo da UOIFT foi melhorado para lidar com a problemática relacionada a políticas de tratamento de empate e utilizamos o caso da função não monotonicamente incremental no Algoritmo da UOIFT.

Foram realizadas avaliações de desempenho qualitativas e quantitativas, incluindo volumes 3D e também aplicações com imagens coloridas, demonstrando que a polaridade de borda não está restrita a análise de imagens de tons de cinza de transições claro/escuro ou escuro/claro.

Foi construída uma segmentação hierárquica a partir da UOIFT, e também, foram mostrados resultados de mapas de saliência das hierarquias.

Conseguimos integrar a característica local de curvatura com a informação de intensidade de cor, para propor um novo algoritmo UOIFT com características locais de curvatura (UOIFT+LCF).

O uso de relações assimétricas na análise de curvatura permitiu a diferenciação de objetos côncavos e convexos, na segmentação descrita pelo Algoritmo UOIFT+LCF.

Como contribuição deste trabalho, temos as seguintes publicações:

- Um novo método baseado em múltiplas execuções da IFT, chamado OIFT-LCF (*Oriented Image Foresting Transform with Localized Curvilinear Features*), publicado no *International Conference on Image Analysis and Processing (ICIAP17)*, Catania, Itália [Bejar *et al.* (2017)].
- Um novo método de segmentação não supervisionado, chamado UOIFT (*Unsupervised Oriented Image Foresting Transform*), baseado em cortes ótimos em grafos dirigidos, publicado no *23rd Iberoamerican Congress on Pattern Recognition (CIARP 2018)*, Madrid, Espanha [Bejar *et al.* (2018)].

- Uma publicação na revista *Pattern Recognition Letter* do método UOIFT com tratamento de empates, testes em imagem coloridas e 3D [Bejar *et al.* (2020)].
 - Os materiais suplementares desta publicação, incluindo a disponibilização pública de um *dataset* de 40 imagens (conjuntamente com seus gabaritos de segmentação), sob condições adversas de iluminação, estão disponíveis na página web: http://vision.ime.usp.br/~hans/color_uoift/ (Figura 7.1).

Além disso, durante o doutorado também publicamos os seguintes artigos que não foram incluídos no texto da presente tese:

- Cálculo da região de robustez do método ORFC [Bejar e Miranda (2015)], em continuidade aos trabalhos do mestrado, publicado no SPIE *Medical Imaging* 2017, Orlando, Florida, US. [Tavares *et al.* (2017a)].
- Cálculo da região de robustez do método OIFT, publicado no *International Symposium on Mathematical Morphology* (ISMM17), Fontainebleau, França [Tavares *et al.* (2017b)].

Color UOIFT | Unsupervised Oriented Image-Foresting Transform for Color Image

This dataset was used in this scientific publication:

Original image(1548x2064)	Ground truth	UOIFT segmentation

Dataset

- The dataset is composed of 40 images of a tourist guide brochure with diffuse lighting (including ground truth images), which is available [here](#)

Publications

- [PRLetters'20] Hans H.C. Bejar and Silvio Jamil Ferzoli Guimaraes and Paulo A.V. Miranda, "Efficient hierarchical graph partitioning for image segmentation by optimum oriented cuts" | Pattern Recognition Letters, March 2020.

Sponsors

USP Universidade de São Paulo

INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
UNIVERSIDADE DE SÃO PAULO

CAPES

Figura 7.1: Página web dos materiais suplementares da publicação [Bejar et al. (2020)].

7.2 Trabalhos futuros

Observando as várias restrições de alto nível atualmente suportadas pela OIFT e que ainda não estão disponíveis na UOIFT, um possível tema para trabalhos futuros corresponde ao estudo de como incorporá-las na UOIFT. Assim, temos as seguintes sugestões para trabalhos futuros:

- Estender a UOIFT para considerar predicados mais sofisticados, com base nos trabalhos [Cousty et al. (2018); Felzenszwalb e Huttenlocher (2004); Feng et al. (2011); Guimarães et al. (2017); Zhang e Alhajj (2006)].
- Explorar a polaridade de borda nas transições com diferentes atributos de textura.

- Processamento de vídeo pela UOIFT baseada em *supervoxels* computados pelo SICLE [Belém *et al.* (2022)].
- Investigar a incorporação de restrições hierárquicas de inclusão/exclusão na UOIFT, tal como atualmente empregadas na HLOIFT (*Hierarchical Layered Oriented Image Foresting Transform*) [Leon *et al.* (2021a)].
 - Aplicação em imagens médicas explorando as relações estruturais entre os diferentes órgãos do corpo humano.
 - Identificação de marcadores fiduciais (*landmarks/fiducial markers*) envoltos por molduras com múltiplas transições de cores (Figura 7.2).
- Reorganizar a hierarquia produzida pela UOIFT, por meio de técnicas recentes de aprendizado de máquina, tal como explorado nos recentes trabalhos [Adão *et al.* (2020)]
- Estender as restrições de curvatura, aqui apresentadas, para o caso 3D de superfícies. Neste sentido, algumas ideias podem ser aproveitadas com base no trabalho [Paiva *et al.* (2020)], em que uma nova abordagem baseada na integração da Transformada Hierárquica Watershed com técnicas de análise da curvatura da superfície de nuvem de pontos é apresentada para a segmentação de construções históricas de diferentes estilos e períodos arquitetônicos. Neste trabalho, a principal motivação reside na conjectura de que um alto desvio angular (normal) entre pontos vizinhos pode denotar uma região diferente na superfície e áreas com baixa variação angular são definidas como sementes. Acreditamos que a forma como as curvaturas são estimadas neste ambiente 3D poderiam ser aproveitadas na nossa expansão almejada.

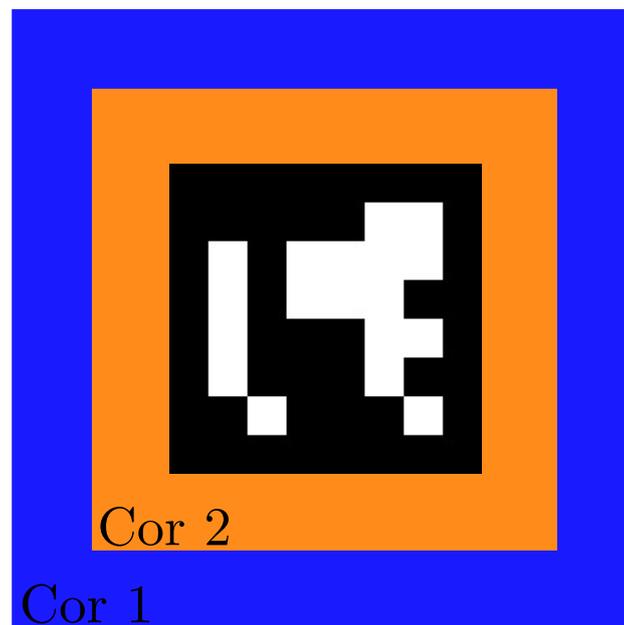


Figura 7.2: Marcadores fiduciais (*landmarks/fiducial markers*) com molduras em múltiplas cores.

Referências Bibliográficas

- Achanta et al. (2010)** Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua e Sabine Süsstrunk. Slic superpixels. página 15. Citado na pág. 6
- Achanta et al. (2012)** Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua e Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282. ISSN 0162-8828. Citado na pág. 51
- Adão et al. (2020)** Milena M. Adão, Silvio Jamil F. Guimarães e Zenilton K.G. Patrocínio Jr. Learning to realign hierarchy for image segmentation. *Pattern Recognition Letters*, 133:287–294. ISSN 0167-8655. doi: <https://doi.org/10.1016/j.patrec.2020.03.010>. URL <https://www.sciencedirect.com/science/article/pii/S016786552030088X>. Citado na pág. 108
- Alexandre et al. (2015)** E. B. Alexandre, A. S. Chowdhury, A. X. Falcão e P. A. V. Miranda. IFT-SLIC: A general framework for superpixel generation based on simple linear iterative clustering and image foresting transform. Em *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, páginas 337–344. Citado na pág. 6, 9, 16, 17, 49, 53, 54, 57, 75, 89, 95, 96, 102, 103
- Andaló et al. (2007)** Fernanda A Andaló, Paulo AV Miranda, Ricardo da Silva Torres e Alexandre X Falcao. A new shape descriptor based on tensor scale. Em *ISMM (1)*, páginas 141–152. Citado na pág. 27
- Antunes et al. (2021)** Daniel Martins Antunes, Jacques-Olivier Lachaud e Hugues Talbot. A maximum-flow model for digital elastica shape optimization. Em Joakim Lindblad, Filip Malmberg e Nataša Sladoje, editors, *Discrete Geometry and Mathematical Morphology*, páginas 429–440, Cham. Springer International Publishing. ISBN 978-3-030-76657-3. Citado na pág. 39
- Arica e Yarman Vural (2003)** Nafiz Arica e Fatos T. Yarman Vural. Bas: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9):1627–1639. ISSN 0167-8655. doi: [https://doi.org/10.1016/S0167-8655\(03\)00002-3](https://doi.org/10.1016/S0167-8655(03)00002-3). URL <https://www.sciencedirect.com/science/article/pii/S0167865503000023>. Citado na pág. 40
- Audigier e Lotufo (2007)** R. Audigier e R.A. Lotufo. Seed-relative segmentation robustness of watershed and fuzzy connectedness approaches. Em *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, páginas 61–68, Belo Horizonte, MG. IEEE CPS. Citado na pág. 2, 51, 75
- Bejar et al. (2017)** Hans H. C. Bejar, Fábio A. M. Cappabianco e Paulo A. V. Miranda. Efficient image segmentation in graphs with localized curvilinear features. Em Sebastiano Battiato, Giovanni Gallo, Raimondo Schettini e Filippo Stanco, editors, *Image Analysis and Processing - ICIAP 2017*, páginas 718–728, Cham. Springer International Publishing. ISBN 978-3-319-68560-1. Citado na pág. 8, 9, 44, 105
- Bejar e Miranda (2015)** Hans H.C. Bejar e Paulo A.V. Miranda. Oriented relative fuzzy connectedness: Theory, algorithms, and its applications in hybrid image segmentation methods. *EURASIP Journal on Image and Video Processing*, 2015(21):21. doi: 10.1186/s13640-015-0067-4. URL <http://dx.doi.org/10.1186/s13640-015-0067-4>. Citado na pág. 3, 4, 9, 55, 66, 106

- Bejar et al. (2018)** Hans H.C. Bejar, Lucy A.C. Mansilla e Paulo A.V. Miranda. Efficient unsupervised image segmentation by optimum cuts in graphs. Em *23rd Iberoamerican Congress on Pattern Recognition (CIARP)*, páginas 359–367, Madrid, Spain. Citado na pág. 9, 105
- Bejar et al. (2020)** Hans H.C. Bejar, Silvio Jamil Ferzoli Guimaraes e Paulo A.V. Miranda. Efficient hierarchical graph partitioning for image segmentation by optimum oriented cuts. *Pattern Recognition Letters*, 131:185 – 192. ISSN 0167-8655. doi: <https://doi.org/10.1016/j.patrec.2020.01.008>. URL <http://www.sciencedirect.com/science/article/pii/S0167865520300088>. Citado na pág. xxi, 9, 106, 107
- Belém et al. (2019)** Felipe Belém, Silvio Jamil F. Guimarães e Alexandre Xavier Falcão. Superpixel segmentation by object-based iterative spanning forest. Em Ruben Vera-Rodriguez, Julian Fierrez e Aythami Morales, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, páginas 334–341, Cham. Springer International Publishing. ISBN 978-3-030-13469-3. Citado na pág. 1
- Belém et al. (2022)** Felipe Belém, Isabela Borlido, Leonardo João, Benjamin Perret, Jean Cousty, Silvio J. F. Guimarães e Alexandre Falcão. Fast and effective superpixel segmentation using accurate saliency estimation. Em Étienne Baudrier, Benoît Naegel, Adrien Krähenbühl e Mohamed Tajine, editors, *Discrete Geometry and Mathematical Morphology*, páginas 261–273, Cham. Springer International Publishing. ISBN 978-3-031-19897-7. Citado na pág. 1, 8
- Belém et al. (2022)** Felipe Belém, Benjamin Perret, Jean Cousty, Silvio J. F. Guimarães e Alexandre Falcão. Efficient multiscale object-based superpixel framework, 2022. URL <https://arxiv.org/abs/2204.03533>. Citado na pág. 6, 8, 9, 108
- Belém et al. (2020)** Felipe C. Belém, Silvio Jamil F. Guimarães e Alexandre X. Falcão. Superpixel segmentation using dynamic and iterative spanning forest. *IEEE Signal Processing Letters*, 27: 1440–1444. doi: 10.1109/LSP.2020.3015433. Citado na pág. 6, 8, 9
- Belém et al. (2021)** Felipe C. Belém, Benjamin Perret, Jean Cousty, Silvio J. F. Guimarães e Alexandre X. Falcão. Towards a simple and efficient object-based superpixel delineation framework. Em *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, páginas 346–353. doi: 10.1109/SIBGRAPI54419.2021.00054. Citado na pág. 1
- Bennett e Mac Donald (1975)** John R. Bennett e John S Mac Donald. On the measurement of curvature in a quantized environment. *IEEE Transactions on computers*, 24(08):803–820. Citado na pág. 34
- Bergo e Falcão (2007)** Felipe P.G. Bergo e Alexandre Falcão. A partitioned algorithm for the image foresting transform. Em *Proceedings of the Eighth International Symposium on Mathematical Morphology*, páginas 425–436, São José dos Campos, SP, Brazil. Gerald J. F. Banon, MCT/INPE. ISBN 978-85-17-00032-4. URL <http://urlib.net/dpi.inpe.br/ismm@80/2007/04.13.17.56>. Citado na pág. 4, 27
- Boykov e Funka-Lea (2006)** Y. Boykov e G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *Intl. Jnl. of Comp. Vision*, 70(2):109–131. ISSN 0920-5691. Citado na pág. 1, 2, 3, 32, 43, 44
- Braz et al. (2022)** Caio de Moraes Braz, Luiz Felipe D. Santos e Paulo A. V. Miranda. Graph-based image segmentation with shape priors and band constraints. Em Étienne Baudrier, Benoît Naegel, Adrien Krähenbühl e Mohamed Tajine, editors, *Discrete Geometry and Mathematical Morphology*, páginas 287–299, Cham. Springer International Publishing. ISBN 978-3-031-19897-7. Citado na pág. 3

- Cappabianco et al. (2019)** F. A. M. Cappabianco, P. F. O. Ribeiro, P. A. V. de Miranda e J. K. Udupa. A general and balanced region-based metric for evaluating medical image segmentation algorithms. Em *2019 IEEE International Conference on Image Processing (ICIP)*, páginas 1525–1529. doi: 10.1109/ICIP.2019.8803083. Citado na pág. 24, 80
- Cappabianco et al. (2007)** F.A.M. Cappabianco, G. Araujo e A.X. Falcão. The image forest transform architecture. Em *Field-Programmable Technology, 2007. ICFPT 2007. International Conference on*, páginas 137–144. doi: 10.1109/FPT.2007.4439242. Citado na pág. 4, 27
- Cappabianco et al. (2017)** Fábio A. M. Cappabianco, Paulo A. V. de Miranda e Jayaram K. Udupa. A critical analysis of the methods of evaluating mri brain segmentation algorithms. Em *2017 IEEE International Conference on Image Processing (ICIP)*, páginas 3894–3898. doi: 10.1109/ICIP.2017.8297012. Citado na pág. 24
- Carballido-Gamio et al. (2004)** J. Carballido-Gamio, S.J. Belongie e S. Majumdar. Normalized cuts in 3D for spinal MRI segmentation. *IEEE Transactions on Medical Imaging*, 23(1):36–44. Citado na pág. 51, 54
- C. Couprie et al. (2010)** C. Couprie, L. Grady, L. Najman e H. Talbot. Power watersheds: A unifying graph-based optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.200. Citado na pág. 3
- Choi et al. (2023)** Jungeui Choi, Edson K. Ueda, Guilherme C. Duran, Paulo A. V. Miranda e Marcos S. G. Tsuzuki. Automatic lung segmentation with seed generation and ROIFT algorithm for the creation of anatomical atlas. Em *ICGG 2022 - Proceedings of the 20th International Conference on Geometry and Graphics*, páginas 636–647, Cham. Springer International Publishing. ISBN 978-3-031-13588-0. Citado na pág. 4
- Ciesielski e Udupa (2010a)** K.C. Ciesielski e J.K. Udupa. Affinity functions in fuzzy connectedness based image segmentation i: Equivalence of affinities. *Computer Vision and Image Understanding*, 114(1):146–154. ISSN 1077-3142. Citado na pág. 32, 43
- Ciesielski e Udupa (2010b)** K.C. Ciesielski e J.K. Udupa. Affinity functions in fuzzy connectedness based image segmentation ii: Defining and recognizing truly novel affinities. *Computer Vision and Image Understanding*, 114(1):155–166. ISSN 1077-3142. Citado na pág. 43
- Ciesielski et al. (2007)** K.C. Ciesielski, J.K. Udupa, P.K. Saha e Y. Zhuge. Iterative relative fuzzy connectedness for multiple objects with multiple seeds. *Computer Vision and Image Understanding*, 107(3):160–182. ISSN 1077-3142. Citado na pág. 1, 2, 3
- Ciesielski et al. (2012a)** K.C. Ciesielski, J.K. Udupa, A.X. Falcão e P.A.V. Miranda. Fuzzy connectedness image segmentation in graph cut formulation: A linear-time algorithm and a comparative analysis. *Journal of Mathematical Imaging and Vision*, 44(3):375–398. Citado na pág. 3, 8, 28
- Ciesielski et al. (2012b)** K.C. Ciesielski, J.K. Udupa, A.X. Falcão e P.A.V. Miranda. A unifying graph-cut image segmentation framework: algorithms it encompasses and equivalences among them. Em *Proc. of SPIE on Medical Imaging: Image Processing*, volume 8314. doi: 10.1117/12.911810. Citado na pág. 3
- Ciesielski et al. (2013)** Krzysztof Chris Ciesielski, P.A.V. Miranda, A.X. Falcão e Jayaram K. Udupa. Joint graph cut and relative fuzzy connectedness image segmentation algorithm. *Medical Image Analysis (MEDIA)*, 17(8):1046–1057. Citado na pág. 3
- Ciesielski et al. (2016)** Krzysztof Chris Ciesielski, Gabor T. Herman e T. Yung Kong. General theory of fuzzy connectedness segmentations. 55(3):304–342. ISSN 1573-7683. doi: 10.1007/s10851-015-0623-7. Citado na pág. 4

- Ciesielski et al. (2018)** Krzysztof Chris Ciesielski, Alexandre Xavier Falcão e Paulo A. V. Miranda. Path-value functions for which dijkstra's algorithm returns optimal mapping. *Journal of Mathematical Imaging and Vision*. ISSN 1573-7683. doi: 10.1007/s10851-018-0793-1. Citado na pág. 27, 30
- Condori e Miranda (2022)** Marcos A. T. Condori e Paulo A. V. Miranda. Differential oriented image foresting transform segmentation by seed competition. Em Étienne Baudrier, Benoît Naegel, Adrien Krähenbühl e Mohamed Tajine, editors, *Discrete Geometry and Mathematical Morphology*, páginas 300–311, Cham. Springer International Publishing. ISBN 978-3-031-19897-7. Citado na pág. 3, 8, 9, 44, 51, 54
- Condori et al. (2017)** Marcos A.T. Condori, Lucy A.C. Mansilla e Paulo A.V. Miranda. Bandeirantes: A graph-based approach for curve tracing and boundary tracking. Em *13th International Symposium on Mathematical Morphology*, volume LNCS 10225, páginas 95–106, Fontainebleau, France. ISBN 978-3-319-57240-6. Citado na pág. 5, 41, 54
- Condori et al. (2020)** Marcos A.T. Condori, Fábio A.M. Cappabianco, Alexandre X. Falcão e Paulo A.V. Miranda. An extension of the differential image foresting transform and its application to superpixel generation. 71:102748. ISSN 1047-3203. Citado na pág. 4, 6
- Costa et al. (2001)** Luciano da Fontoura Costa, A G Campos e Edson Tadeu Monteiro Manoel. An integrated approach to shape analysis: results and perspectives. Em *International Conference on Quality Control by Artificial Vision*. CEPADUES. Citado na pág. 38
- Cousty et al. (2007)** J. Cousty, G. Bertrand, L. Najman e M. Couprie. Watersheds, Minimum Spanning Forests, and the Drop of Water Principle. Relatório Técnico IGM2007-01, Université de Marne-la-Vallée. Citado na pág. 51
- Cousty et al. (2010)** J. Cousty, G. Bertrand, L. Najman e M. Couprie. Watershed cuts: Thinnings, shortest path forests, and topological watersheds. *Trans. on Pattern Analysis and Machine Intelligence*, 32:925–939. ISSN 0162-8828. doi: <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2009.71>. Citado na pág. 1, 3
- Cousty e Najman (2011)** Jean Cousty e Laurent Najman. Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. Em *Mathematical Morphology and Its Applications to Image and Signal Processing - 10th Intl. Symposium, ISMM, Verbania-Intra, Italy, July 6-8*, páginas 272–283. Citado na pág. 52
- Cousty et al. (2008)** Jean Cousty, Gilles Bertrand, Laurent Najman e Michel Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. 31(8):1362–1374. Citado na pág. 1, 3
- Cousty et al. (2018)** Jean Cousty, Laurent Najman, Yukiko Kenmochi e Silvio Guimarães. Hierarchical segmentations with graphs: Quasi-flat zones, minimum spanning trees, and saliency maps. *Journal of Mathematical Imaging and Vision*, 60(4):479–502. ISSN 1573-7683. doi: 10.1007/s10851-017-0768-7. URL <https://doi.org/10.1007/s10851-017-0768-7>. Citado na pág. 5, 16, 19, 20, 51, 52, 53, 107
- da S. Torres e Falcão (2007)** R. da S. Torres e A.X. Falcão. Contour salience descriptors for effective image retrieval and analysis. *Image and Vision Computing*, 25(1):3–13. ISSN 0262-8856. doi: <https://doi.org/10.1016/j.imavis.2005.12.010>. URL <https://www.sciencedirect.com/science/article/pii/S0262885606000266>. SIBGRAPI. Citado na pág. 27
- Danda et al. (2017)** Sravan Danda, Aditya Challa, B. S. Daya Sagar e Laurent Najman. Power tree filter: A theoretical framework linking shortest path filters and minimum spanning tree filters. Em Jesús Angulo, Santiago Velasco-Forero e Fernand Meyer, editors, *Mathematical Morphology and*

Its Applications to Signal and Image Processing, páginas 199–210, Cham. Springer International Publishing. ISBN 978-3-319-57240-6. Citado na pág. 8, 9

de Moraes Braz e Miranda (2014) C. de Moraes Braz e P. A. V. Miranda. Image segmentation by image foresting transform with geodesic band constraints. Em *2014 IEEE International Conference on Image Processing (ICIP)*, páginas 4333–4337. doi: 10.1109/ICIP.2014.7025880. Citado na pág. 3, 27, 44, 51

de Moraes Braz et al. (2019) Caio de Moraes Braz, Paulo A. V. Miranda, Krzysztof Chris Ciesielski e Fábio A. M. Cappabianco. Graph-based segmentation with local band constraints. Em Michel Couprie, Jean Cousty, Yukiko Kenmochi e Nabil Mustafa, editors, *Discrete Geometry for Computer Imagery*, páginas 155–166, Cham. Springer International Publishing. ISBN 978-3-030-14085-4. Citado na pág. 51

de Moraes Braz et al. (2020) Caio de Moraes Braz, Paulo A.V. Miranda, Krzysztof Chris Ciesielski e Fábio A.M. Cappabianco. Optimum cuts in graphs by general fuzzy connectedness with local band constraints. 62:659–672. Citado na pág. 3, 4, 44, 51

Demario e Miranda (2019) Caio L. Demario e Paulo A. V. Miranda. Relaxed oriented image foresting transform for seeded image segmentation. Em *2019 IEEE International Conference on Image Processing (ICIP)*, páginas 1520–1524. doi: 10.1109/ICIP.2019.8803080. Citado na pág. 2, 4, 6

Falcão et al. (2000) A. X. Falcão, J. K. Udupa e F. K. Miyazawa. An ultra-fast user-steered image segmentation paradigm: live wire on the fly. *IEEE Transactions on Medical Imaging*, 19(1):55–62. ISSN 0278-0062. doi: 10.1109/42.832960. Citado na pág. 5, 8, 31, 42, 43, 44

Falcão et al. (1998) Alexandre X. Falcão, Jayaram K. Udupa, Supun Samarasekera, Shoba Sharma, Bruce Elliot Hirsch e Roberto de A. Lotufo. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing*, 60(4):233 – 260. ISSN 1077-3169. doi: <http://dx.doi.org/10.1006/gmip.1998.0475>. URL <http://www.sciencedirect.com/science/article/pii/S1077316998904750>. Citado na pág. 5, 28, 31

Falcão e Bergo (2004) A.X. Falcão e F.P.G. Bergo. Interactive volume segmentation with differential image foresting transforms. *Medical Imaging, IEEE Transactions on*, 23(9):1100–1108. ISSN 0278-0062. doi: 10.1109/TMI.2004.829335. Citado na pág. 4, 6, 27

Falcão et al. (2001) A.X. Falcão, B.S. da Cunha e R.A. Lotufo. Design of connected operators using the image foresting transform. Em *Proceedings of SPIE on Medical Imaging*, volume 4322, páginas 468–479. Citado na pág. 6, 27

Falcão et al. (2004) A.X. Falcão, J. Stolfi e R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29. Citado na pág. 2, 6, 27, 31, 54

Falcão et al. (2002) A.X. Falcão, L. da Fontoura Costa e B.S. da Cunha. Multiscale skeletons by image foresting transform and its application to neuromorphometry. *Pattern Recognition*, 35(7): 1571 – 1582. ISSN 0031-3203. doi: [https://doi.org/10.1016/S0031-3203\(01\)00148-0](https://doi.org/10.1016/S0031-3203(01)00148-0). Citado na pág. 6, 27

Farabet et al. (2012) Clement Farabet, Camille Couprie, Laurent Najman e Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929. Citado na pág. 1

Farabet et al. (2013) Clement Farabet, Camille Couprie, Laurent Najman e Yann LeCun. Learning hierarchical features for scene labeling. *Trans. on Pattern Analysis and Machine Intelligence*, 35: 1915–1929. doi: 10.1109/TPAMI.2012.231. Citado na pág. 1

- Felzenszwalb e Huttenlocher (2004)** Pedro F. Felzenszwalb e Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181. ISSN 1573-1405. doi: 10.1023/B:VISI.0000022288.19776.77. URL <https://doi.org/10.1023/B:VISI.0000022288.19776.77>. Citado na pág. 51, 75, 80, 107
- Feng et al. (2011)** W. Feng, H. Xiang e Y. Zhu. An improved graph-based image segmentation algorithm and its gpu acceleration. Em *2011 Workshop on Digital Media and Digital Content Management*, páginas 237–241. doi: 10.1109/DMDCM.2011.44. Citado na pág. 51, 107
- Förstner e Gülch (1987)** Wolfgang Förstner e Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. Em *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, volume 6, páginas 281–305. Interlaken. Citado na pág. 36
- Garrido-Jurado et al. (2014)** S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas e M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2014.01.005>. URL <https://www.sciencedirect.com/science/article/pii/S0031320314000235>. Citado na pág. 73
- Grady (2006)** Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783. Citado na pág. 1
- Groen e Verbeek (1978)** FCA Groen e PW Verbeek. Freeman-code probabilities of object boundary quantized contours. *Computer Graphics and Image Processing*, 7(3):391–402. Citado na pág. 34
- Guimarães et al. (2017)** Silvio Guimarães, Yukiko Kenmochi, Jean Cousty, Zenilton Patrocínio Jr e Laurent Najman. Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity: the case of the felzenszwalb-huttenlocher method. 2(1):55–75. Citado na pág. 51, 75, 107
- Guimarães et al. (2012)** Silvio Jamil F. Guimarães, Jean Cousty, Yukiko Kenmochi e Laurent Najman. A hierarchical image segmentation algorithm based on an observation scale. Em Georgy Gimel'farb, Edwin Hancock, Atsushi Imiya, Arjan Kuijper, Mineichi Kudo, Shinichiro Omachi, Terry Windeatt e Keiji Yamada, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, páginas 116–125, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN 978-3-642-34166-3. Citado na pág. 51
- Gulshan et al. (2010)** V. Gulshan, C. Rother, A. Criminisi, A. Blake e A. Zisserman. Geodesic star convexity for interactive image segmentation. Em *Proc. of Computer Vision and Pattern Recognition*, páginas 3129–3136. Citado na pág. 17, 24, 49
- Guru e Dinesh (2004)** D.S. Guru e R. Dinesh. Non-parametric adaptive region of support useful for corner detection: a novel approach. *Pattern Recognition*, 37(1):165–168. ISSN 0031-3203. doi: [https://doi.org/10.1016/S0031-3203\(03\)00234-6](https://doi.org/10.1016/S0031-3203(03)00234-6). URL <https://www.sciencedirect.com/science/article/pii/S0031320303002346>. Citado na pág. 34, 37
- Harris e Stephens (1988)** Chris Harris e Mike Stephens. A combined corner and edge detector. Em *In Proc. of Fourth Alvey Vision Conference*, páginas 147–151. Citado na pág. 34, 35, 36
- J.A. et al. (2022)** Romero González J.A., Herrera Navarro M. e H. Jiménez Hernández. Shape descriptor based on curvature. *Open Access Library Journal*, 9(3):1–13. doi: <https://doi.org/10.4236/oalib.1108422>. URL <https://www.scirp.org/journal/paperinformation.aspx?paperid=116205>. Citado na pág. 34

- Jones et al. (2016)** Jonathan-Lee Jones, Xianghua Xie e Ehab Essa. *Biomedical Image Segmentation: Advances and Trends*. CRC Press. Citado na pág. 43
- Kang (2005)** Hyung W. Kang. G-wire: A livewire segmentation algorithm based on a generalized graph formulation. *Pattern Recognition Letters*, 26(13):2042 – 2051. ISSN 0167-8655. doi: <http://dx.doi.org/10.1016/j.patrec.2005.02.011>. URL <http://www.sciencedirect.com/science/article/pii/S0167865505000760>. Citado na pág. 5, 31, 41, 43
- Kitchen e Rosenfeld (1982)** Les Kitchen e Azriel Rosenfeld. Gray-level corner detection. *Pattern recognition letters*, 1(2):95–102. Citado na pág. 34
- Kovalevsky (1989)** V.A Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46(2):141 – 161. ISSN 0734-189X. doi: [https://doi.org/10.1016/0734-189X\(89\)90165-5](https://doi.org/10.1016/0734-189X(89)90165-5). URL <http://www.sciencedirect.com/science/article/pii/0734189X89901655>. Citado na pág. 20
- Krähenbühl e Koltun (2014)** Philipp Krähenbühl e Vladlen Koltun. Geodesic object proposals. Em *Computer Vision – ECCV 2014*, páginas 725–739, Cham. Springer International Publishing. ISBN 978-3-319-10602-1. Citado na pág. 54
- Latecki e Lakamper (2000)** L.J. Latecki e R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190. doi: 10.1109/34.879802. Citado na pág. 34
- Lemes Galvão et al. (2018)** Felipe Lemes Galvão, Alexandre Xavier Falcão e Ananda Shankar Chowdhury. Risf: Recursive iterative spanning forest for superpixel segmentation. Em *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, páginas 408–415. doi: 10.1109/SIBGRAPI.2018.00059. Citado na pág. 6, 7, 9
- Leon e Miranda (2017)** L. M. C. Leon e P. A. V. D. Miranda. Multi-object segmentation by hierarchical layered oriented image foresting transform. Em *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, páginas 79–86. doi: 10.1109/SIBGRAPI.2017.17. Citado na pág. 3
- Leon e de Miranda (2019)** Leissi Margarita Castañeda Leon e Paulo André Vechiatto de Miranda. Efficient interactive multi-object segmentation in medical images. Em Laura Leal-Taixé e Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, páginas 705–710, Cham. Springer International Publishing. ISBN 978-3-030-11018-5. Citado na pág. 3
- Leon et al. (2021a)** Leissi MC Leon, Krzysztof C Ciesielski e Paulo AV Miranda. Efficient hierarchical multi-object segmentation in layered graphs. *Mathematical Morphology-Theory and Applications*, 5(1):21–42. Citado na pág. 108
- Leon et al. (2021b)** Leissi M.C. Leon, Krzysztof C. Ciesielski e Paulo A.V. Miranda. Efficient hierarchical multi-object segmentation in layered graphs. 5(1):21–42. doi: doi:10.1515/mathm-2020-0108. Citado na pág. 3, 44, 51
- Lézoray e Grady (2012)** O. Lézoray e L. Grady. *Image Processing and Analysis with Graphs: Theory and Practice*. CRC Press, California, USA. ISBN 1439855072. Citado na pág. 1, 3
- Lotufo e Falcão (2000)** R.A. Lotufo e A.X. Falcão. The ordered queue and the optimality of the watershed approaches. Em *Proceedings of the Intl. Symposium on Mathematical Morphology and its Applications to Signal and Image Processing (ISMM)*, volume 18, páginas 341–350. Kluwer. Citado na pág. 27, 31
- Lotufo et al. (2002)** R.A. Lotufo, A.X. Falcão e F. Zampiroli. IFT-Watershed from gray-scale marker. Em *Proceedings of the XV Brazilian Symposium on Computer Graphics and Image Processing*, páginas 146–152. IEEE. Citado na pág. 1, 51

- Machairas et al. (2014)** V. Machairas, E. Decencière e T. Walter. Waterpixels: Superpixels based on the watershed transformation. Em *2014 IEEE International Conference on Image Processing (ICIP)*, páginas 4343–4347. doi: 10.1109/ICIP.2014.7025882. Citado na pág. 86
- Madabhushi e Udupa (2005)** Anant Madabhushi e Jayaram K Udupa. Interplay between intensity standardization and inhomogeneity correction in mr image processing. *IEEE Transactions on Medical Imaging*, 24(5):561–576. Citado na pág. 1
- Mansilla e Miranda (2016)** L. A. C. Mansilla e P. A. V. Miranda. Oriented image foresting transform segmentation: Connectivity constraints with adjustable width. Em *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, páginas 289–296. doi: 10.1109/SIBGRAPI.2016.047. Citado na pág. 3, 6, 9, 44, 51
- Mansilla et al. (2013)** L. A. C. Mansilla, P. A. V. V. Miranda e F. A. M. Cappabianco. Image segmentation by image foresting transform with non-smooth connectivity functions. Em *2013 XXVI Conference on Graphics, Patterns and Images*, páginas 147–154. doi: 10.1109/SIBGRAPI.2013.29. Citado na pág. 51
- Mansilla et al. (2016)** L. A. C. Mansilla, P. A. V. Miranda e F. A. M. Cappabianco. Oriented image foresting transform segmentation with connectivity constraints. Em *2016 IEEE International Conference on Image Processing (ICIP)*, páginas 2554–2558. doi: 10.1109/ICIP.2016.7532820. Citado na pág. 3, 6, 9, 44, 51
- Mansilla e Miranda (2013a)** L.A.C. Mansilla e P.A.V. Miranda. Image segmentation by oriented image foresting transform: Handling ties and colored images. Em *18th Intl. Conf. on Digital Signal Processing*, páginas 1–6, Greece. Citado na pág. 3, 32, 33, 44, 55
- Mansilla e Miranda (2013b)** LucyA.C. Mansilla e PauloA.V. Miranda. Image segmentation by oriented image foresting transform with geodesic star convexity. Em Richard Wilson, Edwin Hancock, Adrian Bors e William Smith, editors, *Computer Analysis of Images and Patterns*, volume 8047 of *Lecture Notes in Computer Science*, páginas 572–579. Springer Berlin Heidelberg, York, UK. ISBN 978-3-642-40260-9. doi: 10.1007/978-3-642-40261-6_69. URL http://dx.doi.org/10.1007/978-3-642-40261-6_69. Citado na pág. 3, 6, 9, 44, 48, 51
- Martin et al. (2004)** David R. Martin, Charless C. Fowlkes e Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549. ISSN 0162-8828. Citado na pág. 48, 49
- Miranda et al. (2012)** Paulo A. V. Miranda, Alexandre X. Falcão e Thiago Vallin Spina. Riverbed: A novel user-steered image segmentation method based on optimum boundary tracking. *IEEE Transactions on Image Processing*, 21(6):3042–3052. doi: <http://dx.doi.org/10.1109/TIP.2012.2188034>. Citado na pág. 28, 31, 44
- Miranda e Falcão (2011)** P.A.V. Miranda e A.X. Falcão. Elucidating the relations among seeded image segmentation methods and their possible extensions. Em *XXIV Conference on Graphics, Patterns and Images*, páginas 289–296, Maceió, AL. Citado na pág. 28
- Miranda e Mansilla (2014)** P.A.V. Miranda e L.A.C. Mansilla. Oriented image foresting transform segmentation by seed competition. *IEEE Transactions on Image Processing*, 23(1):389–398. ISSN 1057-7149. Citado na pág. 3, 32, 33, 44, 51, 55
- Miranda et al. (2010)** P.A.V. Miranda, A.X. Falcão e J.K. Udupa. Synergistic arc-weight estimation for interactive image segmentation using graphs. *Computer Vision and Image Understanding*, 114(1):85–99. ISSN 1077-3142. Citado na pág. 32, 43, 48
- Moravec (1980)** Hans Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Relatório técnico, Carnegie Mellon University, Pittsburgh, PA. Citado na pág. 35

- Nixon e Aguado (2019)** Mark Nixon e Alberto Aguado. *Feature extraction and image processing for computer vision*. Academic press. Citado na pág. 34
- Oliveira et al. (2021)** Débora E. C. Oliveira, Caio L. Demario e Paulo A. V. Miranda. Image segmentation by relaxed deep extreme cut with connected extreme points. Em Joakim Lindblad, Filip Malmberg e Nataša Sladoje, editors, *Discrete Geometry and Mathematical Morphology*, páginas 441–453, Cham. Springer International Publishing. ISBN 978-3-030-76657-3. Citado na pág. 1
- Oquab et al. (2015)** Maxime Oquab, Léon Bottou, Ivan Laptev e Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. Em *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 685–694. doi: 10.1109/CVPR.2015.7298668. Citado na pág. 1
- Paiva et al. (2020)** Pedro V. Paiva, Camila Cogima, Eloisa Dezen Kempter e Marco Carvalho. Historical building point cloud segmentation combining hierarchical watershed transform and curvature analysis. *Pattern Recognition Letters*, 135:114–121. ISSN 0167–8655. doi: <https://doi.org/10.1016/j.patrec.2020.04.010>. URL <https://www.sciencedirect.com/science/article/pii/S0167865520301306>. Citado na pág. 108
- Papa et al. (2007)** J.P. Papa, A. Falcão e P.A.V. Miranda. Design of robust pattern classifiers based on optimum-path forests. Em *Proceedings of the Eighth International Symposium on Mathematical Morphology*, páginas 337–348, São José dos Campos, SP, Brazil. Gerald J. F. Bannan, MCT/INPE. ISBN 978-85-17-00032-4. URL <http://urlib.net/dpi.inpe.br/ismm@80/2007/04.13.23.19>. Citado na pág. 28
- Peng et al. (2011)** Bo Peng, Lei Zhang, David Zhang e Jian Yang. Image segmentation by iterated region merging with localized graph cuts. *Pattern Recognition*, 44(10):2527 – 2538. ISSN 0031-3203. doi: <http://dx.doi.org/10.1016/j.patcog.2011.03.024>. URL <http://www.sciencedirect.com/science/article/pii/S0031320311001282>. Semi-Supervised Learning for Visual Content Analysis and Understanding. Citado na pág. 3
- Perret et al. (2015)** Benjamin Perret, Jean Cousty, Olena Tankyevych, Hugues Talbot e Nicolas Passat. Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1162–1176. doi: 10.1109/TPAMI.2014.2366145. Citado na pág. 4
- Rauber et al. (2013)** P. E. Rauber, A. X. Falcão, T. V. Spina e P. J. de Rezende. Interactive segmentation by image foresting transform on superpixel graphs. Em *2013 XXVI Conference on Graphics, Patterns and Images*, páginas 131–138. Citado na pág. 49
- Rocha et al. (2009)** Leonardo Marques Rocha, Fábio A. M. Cappabianco e Alexandre Xavier Falcão. Data clustering as an optimum-path forest problem with applications in image analysis. *Int. J. Imaging Syst. Technol.*, 19(2):50–68. ISSN 0899-9457. doi: 10.1002/ima.v19:2. URL <http://dx.doi.org/10.1002/ima.v19:2>. Citado na pág. 28
- Romero-Ramirez et al. (2018)** Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas e Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38–47. ISSN 0262-8856. doi: <https://doi.org/10.1016/j.imavis.2018.05.004>. URL <https://www.sciencedirect.com/science/article/pii/S0262885618300799>. Citado na pág. 73
- Saha e Udupa (2001)** P.K. Saha e J.K. Udupa. Relative fuzzy connectedness among multiple objects: Theory, algorithms, and applications in image segmentation. *Comp. Vision and Image Understanding*, 82(1):42–56. ISSN 1077-3142. Citado na pág. 2, 8, 9

- Schoenemann et al. (2009)** T. Schoenemann, F. Kahl e D. Cremers. Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. Em *2009 IEEE 12th International Conference on Computer Vision*, páginas 17–23. Citado na pág. 46
- Shi e Malik (2000)** J. Shi e J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905. Citado na pág. 51, 54, 75
- Shi e Tomasi (1994)** Jianbo Shi e Tomasi. Good features to track. Em *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, páginas 593–600. doi: 10.1109/CVPR.1994.323794. Citado na pág. 34
- Silva e Lotufo (2011)** Alexandre Gonçalves Silva e Roberto de Alencar Lotufo. Efficient computation of new extinction values from extended component tree. *Pattern Recogn. Lett.*, 32(1): 79–90. ISSN 0167-8655. doi: 10.1016/j.patrec.2010.07.019. Citado na pág. 51, 53, 71, 75
- Souza et al. (2017)** Roberto Souza, Letícia Rittner, Rubens Machado e Roberto Lotufo. iamxt: Max-tree toolbox for image processing and analysis. *SoftwareX*, 6:81 – 84. ISSN 2352-7110. doi: <https://doi.org/10.1016/j.softx.2017.03.001>. Citado na pág. 51, 53, 75
- Strand et al. (2013)** Robin Strand, Krzysztof Chris Ciesielski, Filip Malmberg e Punam K. Saha. The minimum barrier distance. *Computer Vision and Image Understanding*, 117(4): 429–437. ISSN 1077-3142. doi: <https://doi.org/10.1016/j.cviu.2012.10.011>. URL <https://www.sciencedirect.com/science/article/pii/S1077314212001750>. Special issue on Discrete Geometry for Computer Imagery. Citado na pág. 8, 9
- Suetens (2009)** Paul Suetens. *Fundamentals of Medical Imaging*. Cambridge University Press, 2 ed. doi: 10.1017/CBO9780511596803. Citado na pág. 1
- Tavares et al. (2017a)** Anderson C. M. Tavares, Hans H. C. Bejar e Paulo A. V. Miranda. Seed robustness of oriented relative fuzzy connectedness: core computation and its applications. Em *Proceedings of SPIE on Medical Imaging*, volume 10133, páginas 1013316–1013316–10, Orlando, Florida, US. doi: 10.1117/12.2254646. Citado na pág. 9, 56, 106
- Tavares et al. (2017b)** Anderson Carlos Moreira Tavares, Hans Harley Ccacyahuillca Bejar e Paulo André Vechiatto Miranda. Seed robustness of oriented image foresting transform: Core computation and the robustness coefficient. Em Jesús Angulo, Santiago Velasco-Forero e Fernand Meyer, editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, páginas 119–130, Cham. Springer International Publishing. ISBN 978-3-319-57240-6. Citado na pág. 4, 9, 51, 66, 106
- Tomasi e Kanade (1992)** Carlo Tomasi e Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2):137–154. Citado na pág. 36
- Torres et al. (2004)** R da S Torres, Alexandre X Falcao e L da F Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174. Citado na pág. 27, 39
- Tsai et al. (1999)** Du-Ming Tsai, H.-T. Hou e H.-J. Su. Boundary-based corner detection using eigenvalues of covariance matrices. *Pattern Recognition Letters*, 20(1):31–40. ISSN 0167-8655. doi: [https://doi.org/10.1016/S0167-8655\(98\)00130-5](https://doi.org/10.1016/S0167-8655(98)00130-5). URL <https://www.sciencedirect.com/science/article/pii/S0167865598001305>. Citado na pág. 37
- Vallin Spina et al. (2014)** T. Vallin Spina, P.A.V. de Miranda e A. Xavier Falcão. Hybrid approaches for interactive image segmentation using the live markers paradigm. Em *IEEE Conference on Computer Vision and Pattern Recognition*, páginas 5756–5769, Campinas, Brazil. IEEE. Citado na pág. 43

- Vargas-Muñoz et al. (2018)** John E. Vargas-Muñoz, Ananda S. Chowdhury, Eduardo Barreto-Alexandre, Felipe L. Galvão, Paulo André Vechiatto Miranda e Alexandre X. Falcão. An iterative spanning forest framework for superpixel segmentation. *CoRR*, abs/1801.10041. URL <http://arxiv.org/abs/1801.10041>. Citado na pág. 51, 57, 89, 95
- Vargas-Muñoz et al. (2019)** J. E. Vargas-Muñoz, A. S. Chowdhury, E. B. Alexandre, F. L. Galvão, P. A. V. Miranda e A. X. Falcão. An iterative spanning forest framework for superpixel segmentation. 28(7):3477–3489. Citado na pág. 6, 9, 75
- Vezhnevets e Konouchine (2005)** V. Vezhnevets e V. Konouchine. "growcut- interactive multi-label N-D image segmentation by cellular automata. Em *Proc. Graphicon.*, páginas 150–156. Citado na pág. 28
- Wang et al. (2016)** Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang e Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. Em *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 2285–2294. Citado na pág. 1
- Wang e Sinkind (2003)** S. Wang e J.M. Sinkind. Image segmentation with ratio cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675–690. Citado na pág. 51, 54
- Wang e Siskind (2001)** S. Wang e J.M. Siskind. Image segmentation with minimum mean cut. Em *International Conference on Computer Vision (ICCV)*, volume 1, páginas 517–525. Citado na pág. 51, 54
- Wolf et al. (2017)** S. Wolf, L. Schott, U. Kothe e F. Hamprecht. Learned watershed: End-to-end learning of seeded segmentation. Em *International Conference on Computer Vision (ICCV)*, páginas 2030–2038. doi: 10.1109/ICCV.2017.222. Citado na pág. 1, 48
- Wu e Leahy (1993)** Z. Wu e R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its applications to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113. Citado na pág. 51
- Yang et al. (2010)** W. Yang, J. Cai, J. Zheng e J. Luo. User-friendly interactive image segmentation through unified combinatorial user inputs. *IEEE Transactions on Image Processing*, 19(9):2470–2479. ISSN 1057-7149. doi: 10.1109/TIP.2010.2048611. Citado na pág. 43
- Zeng et al. (2019)** Jiexian Zeng, Min Liu, Xiang Fu, Ruiyu Gu e Lu Leng. Curvature bag of words model for shape recognition. *IEEE Access*, 7:57163–57171. doi: 10.1109/ACCESS.2019.2913688. Citado na pág. 34
- Zhang e Alhajj (2006)** M. Zhang e R. Alhajj. Improving the graph-based image segmentation method. Em *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, páginas 617–624. doi: 10.1109/ICTAI.2006.66. Citado na pág. 51, 107
- Zhao et al. (2019)** Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu e Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232. Citado na pág. 1

Índice Remissivo

- Live Wire*, 31
- avaliação, 21
 - dice, 22
 - falso negativo, 21
 - falso positivo, 21
 - jaccard, 23
 - matriz confusão, 21
 - usuário robô, 24
 - verdadeiro negativo, 21
 - verdadeiro positivo, 21
- curvatura
 - k-curvatura, 40
- digrafo
 - arcos, 12
 - vértices, 12
- grafos, 12
 - caminho, 14
 - componentes conexos, 15
 - DCC, 15
 - simétrico, 12
 - transposto, 12
- hierarquia, 17
 - mapa de saliência, 19
- imagem, 11
 - bidimensional, 11
 - multibanda, 11
 - multidimensional, 11
- imagem como grafo
 - adjacência, 13
- OIFT-LCF, 43
- segmentação, 16, 27
 - partição, 16
 - RAG, 16
 - superpixel, 16
- Transformada Imagem-Floresta, 27
- transformada imagem-floresta
 - função de conectividade, 28
 - MI, 29
 - NMI, 29
- UOIFT, 51
 - algoritmo, 56
 - extinção de volume, 68
 - hierarquia, 68
 - imagens coloridas, 57
 - partição, 68
- UOIFT+LCCF
 - algoritmo, 93
 - côncava, 93
- UOIFT+LCF, 85
 - algoritmo, 87
 - curvatura, 85