



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS



FACULDADE DE TECNOLOGIA

---

**Corte Normalizado em Grafos: um Algoritmo  
Aglomerativo para Segmentação de  
Imagens de Colônias de Bactérias**

---

André Luís da COSTA

Limeira  
2013



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS



FACULDADE DE TECNOLOGIA

---

**Corte Normalizado em Grafos: um Algoritmo Aglomerativo para  
Segmentação de Imagens de Colônias de Bactérias**

---

*Normalized cut on graphs: an agglomerative algorithm  
for bacterial colonies image segmentation*

*Dissertação de Mestrado apresentada ao programa de Pós-Graduação da Faculdade de Tecnologia da Universidade Estadual de Campinas para obtenção do título de Mestre em Tecnologia, área de concentração AA – Tecnologia e Inovação, na linha de pesquisa Engenharia da Informação.*

**André Luís da COSTA**

**Orientador:**

**Prof. Dr. Marco Antonio Garcia de CARVALHO**

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO  
DEFENDIDA PELO ALUNO ANDRÉ LUÍS DA COSTA E ORIENTADO PELO  
PROF. DR. MARCO ANTONIO GARCIA DE CARVALHO.

ASSINATURA DO ORIENTADOR

---

Limeira  
2013

FICHA CATALOGRÁFICA ELABORADA POR VANESSA EVELYN COSTA CRB-8/8295  
BIBLIOTECA UNIFICADA FT/CTL  
UNICAMP

Costa, André Luís da, 1982-  
C823c Corte normalizado em grafos: um algoritmo aglomerativo  
para segmentação de imagens de colônias de bactérias /  
André Luís da Costa. -  
Limeira, SP : [s.n.], 2013.

Orientador: Marco Antonio Garcia de Carvalho.  
Dissertação (mestrado) – Universidade Estadual de  
Campinas, Faculdade de Tecnologia.

1. Corte normalizado. 2. Teoria dos grafos. 3. Segmentação

I. Carvalho, Marco Antonio Garcia de. II. Universidade

Informações para Biblioteca Digital

Título em inglês: Normalized cut on graphs: an agglomerative algorithm for bacterial colonies image segmentation.

Palavras-chave em inglês (Keywords):

- 1- Normalized cut
- 2- Graph theory
- 3- Image segmentation
- 4- Microbiology – culture media and culture

Área de concentração: Tecnologia e Inovação

Titulação: Mestre em Tecnologia

Banca examinadora: Hélio Pedrini e João Paulo Papa.

Data da Defesa: 22-02-2013

Programa de Pós-Graduação em Tecnologia

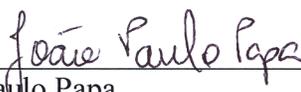
**DISSERTAÇÃO DE MESTRADO EM TECNOLOGIA**  
**ÁREA DE CONCENTRAÇÃO: TECNOLOGIA E INOVAÇÃO**

Corte normalizado em grafos: um algoritmo aglomerativo para segmentação de imagens de colônias de bactérias

**Autor:** André Luis da Costa

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:

  
\_\_\_\_\_  
Prof. Dr. Marco Antonio Garcia de Carvalho, Presidente  
FT/UNICAMP

  
\_\_\_\_\_  
Prof. Dr. João Paulo Papa  
UNESP

  
\_\_\_\_\_  
Prof. Dr. Hélio Pedrini  
IC/UNICAMP

**Errata:** onde se lê “André Luis da Costa”, leia-se André Luís da Costa.

# Agradecimentos

Agradeço primeiramente ao meu orientador Prof. Dr. Marco A. G. de Carvalho por me acompanhar nesta importante etapa da minha formação. Sempre com serenidade, soube analisar minhas demandas e indicar o que foi realmente importante.

À minha companheira e amiga, Elaine Ayumi Chiba, agradeço pela admiração e respeito ao meu trabalho, sempre me apoiando nas mais diversas dificuldades da empreitada.

Agradeço a toda equipe técnica do Laboratório de Ecotoxicologia e Microbiologia Ambiental Prof. Dr. Abílio Lopes – LEAL – em especial ao técnico Gilberto de Almeida que pacientemente me apoiou no processo de aquisição das imagens de placas com colônias de bactérias. Agradeço também a todos que ajudaram na tarefa de anotação das imagens: Caroline Vieira Aio, Wilson Augusto Venancio, Luciane de Oliveira Leite Santos, Francielle Fernandes, Samuel Jonathan da Silva Pereira, Eduardo Henrique Martins, Vinícius Gottsholl, Rita Maria da Silva e Marielly Reis Resende.

À Fapesp, agradeço pela confiança depositada em meus projetos. Espero sempre poder corresponder aos investimentos feitos nas diversas etapas da minha formação. *Este trabalho foi apoiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (Fapesp) por meio de bolsa regular de mestrado, processo nº 2010/14759-0.*

# Resumo

O problema de segmentação de colônias de bactérias em placas de Petri possui algumas características bem distintas daquelas encontradas, por exemplo, em problemas de segmentação de imagens naturais. A principal característica é o alto número de colônias que podem ser encontradas em uma placa. Desta forma, é primordial que o algoritmo de segmentação seja capaz de realizar a segmentação da imagem em um grande número de regiões. Este cenário extremo é ideal para analisar limitações dos algoritmos de segmentação. De fato, neste trabalho foi verificado que o algoritmo de corte normalizado original, que se fundamenta na teoria espectral de grafos, é inadequado para aplicações que exigem que a segmentação seja realizada em um grande número de regiões. Contudo, a utilização do critério de corte normalizado para segmentar imagens de colônias de bactérias ainda é possível graças a um novo algoritmo que está sendo introduzido neste trabalho. O novo algoritmo fundamenta-se no agrupamento hierárquico dos nós do grafo, ao invés de utilizar conceito da teoria espectral. Experimentos mostram também que o biparticionamento de um grafo pelo novo algoritmo apresenta um valor de corte normalizado médio cerca de 40 vezes menor que o biparticionamento pelo algoritmo baseado na teoria espectral.

**Palavras-chave:** Corte normalizado; Teoria dos grafos; Segmentação de imagens; Microbiologia – cultura e meios de cultura

# Abstract

The problem of bacteria colonies segmentation in Petri dishes has some very different characteristics from those found, for example, in segmenting natural images. The main feature is the high number of colonies that can be found on a plate. Thus, it is essential that the segmentation algorithm is capable of performing the image segmentation into a huge number of regions. This extreme scenario is ideal for analyzing segmentation algorithms limitations. In fact, this study showed that the original normalized cut algorithm, which is based on the spectral graph theory, is inappropriate for applications that require that the segmentation be performed on a large number of regions. However, the use of normalized cut criteria for segmenting bacteria colonies images is still possible thanks to a new algorithm that is being introduced in this paper. The new algorithm is based on hierarchical clustering of the graph nodes, instead of using the spectral theory concepts. Experiments also show that the bi-partitioning of a graph by the new algorithm has a normalized cut average value about 40 times lesser than the bi-partitioning by the algorithm based on the spectral theory.

**Keywords:** Normalized cut; Graph theory; Image segmentation; Microbiology – culture media and culture

# Sumário

<b>Agradecimentos</b>	<b>iv</b>
<b>Resumo</b>	<b>v</b>
<b>Lista de figuras</b>	<b>xi</b>
<b>Lista de tabelas</b>	<b>xii</b>
<b>Lista de algoritmos</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	2
1.2 Objetivos . . . . .	3
1.3 Contribuições . . . . .	4
1.4 Organização do texto . . . . .	5
<b>2 Teoria e trabalhos relacionados</b>	<b>6</b>
2.1 Segmentação de imagens . . . . .	6
2.1.1 Redução de ruído . . . . .	7
2.2 Teoria dos grafos . . . . .	7
2.2.1 Corte em grafos . . . . .	8
2.2.2 Corte normalizado . . . . .	9
2.3 <i>Benchmark</i> de segmentação de imagens da Berkeley . . . . .	11
2.4 Contagem de bactérias em placas . . . . .	12

2.4.1	Contagem automática . . . . .	14
<b>3</b>	<b>Modelos de grafos de similaridade</b>	<b>16</b>
3.1	Modelos baseados na grade de <i>pixels</i> . . . . .	17
3.1.1	Multiescala . . . . .	17
3.2	Modelos fundamentados em estruturas hierárquicas . . . . .	18
3.2.1	Modelo baseado na árvore dos componentes . . . . .	18
3.2.2	Modelo baseado na <i>quadtree</i> . . . . .	21
3.3	Teste de desempenho dos modelos . . . . .	23
3.3.1	Configuração do experimento . . . . .	23
3.3.2	Resultados . . . . .	25
3.3.3	Custo computacional . . . . .	28
3.3.4	Aproveitando as características do modelo baseado na <i>quadtree</i> . . . . .	29
3.4	Conclusão . . . . .	31
<b>4</b>	<b>Um novo algoritmo para aproximar o corte normalizado ótimo</b>	<b>33</b>
4.1	Definições preliminares . . . . .	34
4.2	O algoritmo . . . . .	35
4.2.1	Heurística . . . . .	36
4.3	Teste de desempenho do algoritmo . . . . .	38
4.4	Conclusão . . . . .	41
<b>5</b>	<b>Utilizando o corte normalizado para segmentar colônias de bactérias em placas de Petri</b>	<b>42</b>
5.1	O corte normalizado aplicado à segmentação de colônias de bactérias . . . . .	43
5.1.1	Experimentos . . . . .	44
5.2	Conclusão . . . . .	45
<b>6</b>	<b>Considerações finais</b>	<b>49</b>
6.1	A trajetória do curso de mestrado . . . . .	49
6.1.1	Aspectos práticos da contagem de colônias . . . . .	50

6.1.2	A mudança de foco do trabalho . . . . .	51
6.2	Trabalhos futuros . . . . .	52
6.3	Conclusão . . . . .	53
	<b>Bibliografia</b>	<b>57</b>

# Lista de Figuras

2.1	Etapas do algoritmo de corte normalizado baseado na TEG aplicado à segmentação de imagens . . . . .	10
2.2	Ilustração da diluição seriada para contagem de colônias de bactérias em placas de Petri . . . . .	13
2.3	Diagrama do sistema de contagem automática de colônias de bactérias proposto por Chen e Zhang . . . . .	15
3.1	Limiarização da imagem . . . . .	19
3.2	Representação da árvore dos componentes e da árvore dos componentes reversa . .	20
3.3	Passos para gerar um grafo de similaridade pelo modelo baseado na árvore dos componentes . . . . .	20
3.4	Geração de uma <i>quadtree</i> a partir de uma imagem binária . . . . .	21
3.5	Passos para gerar um grafo de similaridade pelo modelo baseado na <i>quadtree</i> . . .	22
3.6	Pontuações para cada imagem do conjunto de validação do <i>benchmark</i> BSR500 . .	26
3.7	Pontuações para cada imagem do conjunto de teste do <i>benchmark</i> BSR500 . . . . .	27
3.8	Segmentação utilizando o modelo <i>quadtree</i> . . . . .	30
4.1	Agrupamento dos nós $u$ e $v$ ao passar da iteração $n$ para a iteração $n + 1$ . . . . .	35
4.2	Resultados do particionamento de 500 grafos gerados a partir das imagens do <i>benchmark</i> BSDS500, em 2 subgrafos . . . . .	38
4.3	Resultados do particionamento dos 500 grafos gerados a partir das imagens do <i>benchmark</i> BSDS500, em 10 e em 20 subgrafos . . . . .	39
4.4	Tempo de processamento para particionar os 500 grafos gerados a partir das imagens do <i>benchmark</i> BSDS500 . . . . .	40
5.1	Uma imagem em alta definição de uma placa de Petri cultivada com bactérias heterotróficas . . . . .	44

5.2	Segmentações resultantes para dois fragmentos de imagens de placas de Petri . . .	46
5.3	Segmentações resultantes para dois fragmentos de imagens de placas de Petri com menor densidade de colônias . . . . .	47

# Lista de Tabelas

3.1	Pontuação geral dos modelos nas imagens dos conjuntos de validação e de teste do <i>benchmark</i> BSDS500 . . . . .	25
3.2	Tempo médio que cada modelo levou para computar as segmentações em 10 e em 30 regiões . . . . .	28
3.3	Pontuação do modelo baseado na <i>quadtree</i> utilizando o ajuste dado pela Equação 3.6	31

# Lista de Algoritmos

4.1	Algoritmo para aproximação do valor ótimo de corte normalizado em um grafo de similaridade pelo agrupamento de nós . . . . .	37
-----	------------------------------------------------------------------------------------------------------------------------------	----

# Capítulo 1

## Introdução

O Processamento Digital de Imagens (PDI) é definido como um conjunto de técnicas para manipulação dos dados contidos em imagens digitais. Um dos objetivos do PDI é a interpretação das informações presentes na imagem (GONZALEZ E WOODS, 2010; PEDRINI E SCHWARTZ, 2008). Uma vez que este objetivo requer um processo complexo para ser atingido, é usual realizar-se uma tarefa intermediária de segmentação para dividir a imagem em estruturas com conteúdo relevante para a aplicação (PEDRINI E SCHWARTZ, 2008). Essas estruturas correspondem a objetos ou partes de objetos que auxiliarão no processo de interpretação. Algumas técnicas existentes para se realizar a segmentação de imagens são bem simples, *e. g.*, a limiarização global (GONZALEZ E WOODS, 2010; PEDRINI E SCHWARTZ, 2008). Outras baseiam-se em algoritmos e procedimentos mais sofisticados, *e. g.*, a segmentação por corte em grafos (SHI E MALIK, 2000). Em geral, as técnicas mais simples visam à alta eficiência computacional, enquanto que as mais sofisticadas buscam melhorar a qualidade da segmentação. A escolha da técnica de segmentação a ser utilizada é altamente dependente das características e dos requerimentos do problema a ser resolvido (PEDRINI E SCHWARTZ, 2008). Mais ainda, essa escolha pode ir além de simplesmente optar-se por maior eficiência ou maior eficácia.

Shi e Malik (1997, 2000) propuseram um *framework* de segmentação de imagens fundamentado na teoria dos grafos. Em linhas gerais, o *framework* realiza a segmentação por meio do particionamento de um grafo de similaridade que representa a imagem. O particionamento do grafo é feito utilizando-se o conceito de corte, com base em um critério global denominado *Corte Normalizado* (NCut, do inglês *Normalized Cut*). Assim, dado o particionamento de um grafo de similaridade  $G$  em subgrafos  $A$  e  $B$ , o critério de corte normalizado mede o quanto o subgrafo  $A$  é dissimilar ao subgrafo  $B$ , em relação ao quanto os elementos internos de cada subgrafo são similares entre si. O critério de corte normalizado busca solucionar um problema de tendência ao desbalancea-

mento do particionamento apresentado pelo critério de corte mínimo (WU E LEAHY, 1993; SHI E MALIK, 2000). Maiores detalhes sobre o funcionamento do *framework* de segmentação por corte normalizado estão disponíveis na Seção 2.2.2.

Segundo Shi e Malik (2000), o particionamento ótimo de um grafo de similaridade é aquele que apresenta o menor valor de corte normalizado. Porém, a minimização do critério de corte normalizado é um problema de complexidade NP-Completo. Desta forma, Shi e Malik (2000) propuseram também um algoritmo, fundamentado na teoria espectral de grafos, que provê uma aproximação do particionamento ótimo. O princípio do algoritmo está em determinar o particionamento com base nos autovetores calculados para uma matriz obtida a partir de um grafo de similaridade. O algoritmo pode ser aplicado em dois modos de operação: *2-way*, em que cada região da imagem é recursivamente particionada em duas novas regiões; ou *k-way*, em que o particionamento é realizado diretamente em um dado número  $k$  de regiões. No caso do modo *2-way*, a cada iteração, um novo autovetor deve ser calculado a partir de uma nova matriz que representa o respectivo subgrafo. No caso do modo *k-way*, é necessário o cálculo de  $k$  autovetores a partir da matriz que representa o grafo. Desta forma, existe a hipótese de que, para ambos os modos de operação, o algoritmo não seja adequado para problemas de segmentação que envolvem o particionamento da imagem em um número muito grande de regiões, uma vez que o aumento no número de partições implica também aumentar o custo computacional.

Uma aplicação que pode envolver um elevado número de regiões é a segmentação de colônias de bactérias em placas de Petri. Uma única placa contável pode conter centenas de colônias (TOR-TORA *et al.*, 2005). Considerando uma abordagem do problema em que cada colônia corresponda a um segmento, espera-se que os resultados apresentem partições compostas por muitas regiões. Esta característica extrema desse tipo de aplicação tem o potencial de revelar aspectos importantes sobre o funcionamento do *framework* de corte normalizado.

## 1.1 Motivação

Nosso interesse pelo corte normalizado vem desde nosso projeto de iniciação científica intitulado “Segmentação de imagens modeladas por estruturas hierárquicas em grafos utilizando a técnica de corte normalizado”, executado no período de setembro de 2009 a agosto de 2010. Na ocasião do projeto de iniciação científica, a principal motivação para se estudar o corte normalizado foi porque ele era uma técnica de segmentação relativamente recente e que, desta forma, teria um grande potencial para revelar assuntos ainda pouco explorados. Os resultados obtidos durante o projeto de iniciação científica nos motivaram a continuar investigando a técnica de corte normalizado. Desta

vez, porém, estávamos interessados em saber se a técnica poderia ser aplicada universalmente a qualquer problema de segmentação de imagens.

Evidentemente, seria muita pretensão tentar compilar um conjunto de problemas que pudesse representar efetivamente todos os possíveis problemas de segmentação de imagens. Porém, seria perfeitamente possível encontrar ao menos um problema suficientemente específico que pudesse revelar propriedades importantes da técnica de segmentação por corte normalizado. O problema de segmentação de colônias de bactérias foi inicialmente escolhido para o estudo. Na ocasião, porém, sabíamos apenas que este seria um problema bastante distinto do problema de segmentação de imagens naturais que utilizamos durante o projeto de iniciação científica. Com o andamento da pesquisa pudemos verificar que de fato o problema de segmentação de colônias de bactérias oferece alguns desafios às técnicas de segmentação:

1. dado a existência de microcolônias, é necessário trabalhar com imagens de alta resolução;
2. uma única placa pode conter milhares de colônias, podendo exigir que a segmentação resulte em um grande número de regiões;
3. uma colônia pode ser muito pequena, da ordem de apenas alguns pixels (mesmo em imagens de alta resolução), ou pode ser grande o suficiente para ocupar uma parcela significativa da imagem.

Desta forma, a aplicação de segmentação de colônias de bactérias revelou-se promissora para evidenciar características pouco exploradas do *framework* de segmentação de imagens por corte normalizado.

## 1.2 Objetivos

Inicialmente, o objetivo do projeto era o de verificar se a segmentação por corte normalizado poderia produzir melhores resultados que outras técnicas de segmentação aplicadas ao problema de contagem de colônias de bactérias. No entanto, durante a execução do projeto percebemos que este seria um objetivo muito ambicioso, uma vez que não sabíamos sequer se seria minimamente viável segmentar as colônias utilizando o corte normalizado. Assim, o objetivo principal passou a ser verificar se é possível aplicar a técnica de corte normalizado ao problema de segmentação das colônias.

O objetivo principal definiu o foco do trabalho e deu respaldo a um número de objetivos secundários relativos à compreensão das características do *framework* de corte normalizado. Estes objetivos foram:

- melhorar os modelos para grafos de similaridade propostos durante o trabalho de iniciação científica, compreendendo melhor a influência do modelo no processo de segmentação;
- ampliar o conhecimento sobre o *framework* de corte normalizado, buscando evidenciar suas características e possíveis limitações;
- melhorar a técnica de forma geral e, se possível, solucionar eventuais problemas.

### 1.3 Contribuições

A principal contribuição deste trabalho é a proposta de um novo algoritmo para realizar a aproximação do corte normalizado ótimo em um grafo de similaridade. Ao invés de utilizar conceitos da teoria espectral como no algoritmo proposto por Shi e Malik (2000), esse novo algoritmo fundamenta-se no agrupamento hierárquico dos nós do grafo, respeitando-se um conjunto de heurísticas. Este novo algoritmo apresenta propriedades distintas em relação ao algoritmo fundamentado na teoria espectral de grafos, alterando significativamente a perspectiva de aplicação do corte normalizado. Especialmente para problemas de mesma natureza que a segmentação de colônias de bactérias, a qual pode resultar em um elevado número de regiões.

Outras contribuições são as alterações realizadas nos modelos *quadtree* (CARVALHO *et al.*, 2010B) e árvore dos componentes (CARVALHO *et al.*, 2010A) para gerar um grafo de similaridade. No caso da *quadtree*, as alterações foram muito significativas, melhorando consideravelmente os resultados de segmentação. É apresentado um estudo comparativo entre os dois modelos melhorados neste trabalho e outros dois modelos previamente utilizados pelo *framework* de corte normalizado: o modelo *pixel grid* (SHI E MALIK, 2000) e o modelo *pixel grid* em multiescala (COUR *et al.*, 2005). Um estudo comparativo semelhante foi realizado também por Ferreira (2011). Porém, sem as melhorias realizadas neste trabalho, e utilizando outra base de testes. Resultados parciais do estudo comparativo realizado neste trabalho foram publicados em (CARVALHO E COSTA, 2012).

## 1.4 Organização do texto

O texto deste documento está organizado em capítulos, da seguinte forma:

- o Capítulo 2 apresenta as fundamentação teórica e os principais trabalhos relacionados;
- os modelos *quadtree* e árvore dos componentes atualizados são apresentados no Capítulo 3, o qual apresenta também o estudo comparativo com os modelos *pixel grid* e *pixel grid* em multiescala;
- o Capítulo 4 descreve o novo algoritmo proposto neste trabalho para aproximar o valor ótimo de corte normalizado, juntamente com um experimento utilizando 500 grafos;
- o Capítulo 5 o estudo sobre a aplicabilidade da técnica de segmentação por corte normalizado em problemas de segmentação de colônias de bactérias;
- finalmente, no Capítulo 6 são realizadas algumas considerações finais, e também apresentadas algumas sugestões para trabalhos futuros que podem complementar o estudo realizado neste trabalho de mestrado.

# Capítulo 2

## Teoria e trabalhos relacionados

Neste Capítulo são apresentados os principais conceitos teóricos e trabalhos relacionados com as ferramentas desenvolvidas durante o curso de mestrado, e que estão descritas neste documento.

### 2.1 Segmentação de imagens

A segmentação é um processo intermediário em sistemas computacionais que buscam interpretar os dados contidos em uma imagem digital (GONZALEZ E WOODS, 2010; PEDRINI E SCHWARTZ, 2008). Ela visa simplificar a representação da imagem de forma a facilitar o processo de interpretação. A segmentação pode ser descritiva, *i. e.*, a representação da imagem é alterada para evidenciar certas características como, por exemplo, pontos, linhas, contornos e bordas. Algumas técnicas clássicas de segmentação descritiva são a **transformada de Hough**, utilizada para detectar conjuntos de pontos que pertencem a uma curva específica como retas, circunferências, elipses, etc; e o **operador de Canny**, utilizado para realizar a detecção de bordas. A segmentação pode também ser direta, *i. e.*, resultar em um conjunto de  $n \in \mathbb{N}_{>1}$  regiões  $\{R_1, R_2, \dots, R_n\}$ , tal que:

1.  $\cup_{i=1}^n R_i = R$ , onde  $R$  é a região definida pela própria imagem;
2.  $R_i$  é uma função conexa,  $\forall i$ ;
3.  $R_i \cap R_j = \emptyset$ ,  $\forall i$  e  $j/i \neq j$ ;
4.  $P(R_i) = VERDADEIRO$ ,  $\forall i$ ;
5.  $P(R_i \cup R_j) = FALSO$  para  $i \neq j$  e  $R_i$  adjacente a  $R_j$ .

Onde  $P(R_i)$  é um predicado lógico sobre os pontos de  $R_i$ , *i. e.*, os elementos da região  $R_i$  devem satisfazer certas propriedades como, por exemplo, possuírem a mesma intensidade (PEDRINI E SCHWARTZ, 2008).

### 2.1.1 Redução de ruído

Ruídos podem levar os métodos de segmentação a distorcer as formas dos objetos, comprometendo o processo de interpretação (PEDRINI E SCHWARTZ, 2008). Desta forma, é conveniente que seja realizado um pré-processamento da imagem com o objetivo de reduzir o ruído. Neste trabalho está sendo utilizado o *filtro da mediana* para reduzir o efeito do ruído impulsivo do tipo sal-pimenta. O filtro da mediana é um dos mais importantes filtros não-lineares, o qual consiste em substituir a intensidade de cada pixel pela mediana das intensidades na vizinhança do *pixel*. O filtro da mediana possui a característica de não introduzir níveis de cinza diferentes daqueles já presentes na imagem. Além disso, ele afeta pouco as bordas dos objetos.

## 2.2 Teoria dos grafos

Um grafo  $G = (V, E)$  é uma estrutura de dados definida por um conjunto de nós  $u \in V$ , e por um conjunto de arestas  $e(u, v) \in E$ , em que  $u, v \in V$ . Dois nós  $u$  e  $v$  são vizinhos, denotado por  $u \sim v$ , se eles estão conectados por uma aresta (WILSON E WATKINS, 1990; GOLDBARG E GOLDBARG, 2012). Em um grafo ponderado um peso ou conjunto de pesos é associado a cada aresta, representado da forma  $w(u, v) \in W$ . Neste caso, o grafo pode ser denotado por  $G = (V, E, W)$ , onde  $W$  representa o conjunto de pesos. O grau de um nó  $\text{Deg}(u)$ , em que  $u \in V$ , indica a sua quantidade de arestas incidentes. A força de um nó  $S(u) = \sum_{u \sim v} w(u, v)$  é o total de pesos das arestas incidentes (COSTA *et al.*, 2007). O tamanho de um grafo, denotado neste trabalho por  $\text{Tamanho}(G)$ , é a cardinalidade do conjunto de arestas  $|E|$  do grafo. A ordem do grafo  $\text{Ordem}(G)$  é a cardinalidade do conjunto de nós  $|V|$  do grafo.

Um caminho em um grafo é uma sequência de nós  $u_1, u_2, \dots, u_{n-1}, u_n$ , tal que  $u_n \sim u_{n-1}$  (EULER, 1968; WILSON E WATKINS, 1990; GOLDBARG E GOLDBARG, 2012). Observe que em um caminho não pode haver nós repetidos. Um ciclo é definido como um caminho que começa e termina no mesmo nó, da forma  $u_1, u_2, \dots, u_{n-1}, u_n, u_1$ . Um grafo é dito conexo se para qualquer par de nós  $u, v \in V$  é possível determinar um caminho que começa em  $u$  e termina em  $v$ . Uma árvore é um caso especial de grafo, com as propriedades de ser conexo e sem ciclos. Em uma árvore pode haver a relação lógica de hierarquia.

Grafos podem ser representados por meio de matrizes. Algumas das formas mais utilizadas são:

1. **Matriz de Adjacência**  $A_{n \times n}$ , onde  $n = \text{Ordem}(G)$  (WILSON E WATKINS, 1990; CHUNG, 1997; GOLDBARG E GOLDBARG, 2012), tal que

$$[A]_{uv} = \begin{cases} 1 & \text{se } u \sim v \\ 0 & \text{caso contrário} \end{cases} . \quad (2.1)$$

Analogamente, para grafos ponderados há a **Matriz de Pesos**  $W_{n \times n}$ , tal que

$$[W]_{uv} = \begin{cases} w(u, v) & \text{se } u \sim v \\ 0 & \text{caso contrário} \end{cases} ; \quad (2.2)$$

2. **Matriz Laplaciana**, que é utilizada pelo algoritmo de corte normalizado pela teoria espectral de grafos (CHUNG, 1997; SHI E MALIK, 2000; CVETKOVIC *et al.*, 2010). Para grafos não ponderados, a matriz Laplaciana é definida como  $L_{n \times n} = D_{n \times n} - A_{n \times n}$ . Para grafos ponderados ela é definida como  $L_{n \times n} = D_{n \times n} - W_{n \times n}$ . A matriz  $D_{n \times n}$  é diagonal, sendo

$$[D]_{uv} = \begin{cases} \text{Deg}(u) & \text{se } u = v \\ 0 & \text{caso contrário} \end{cases} . \quad (2.3)$$

## 2.2.1 Corte em grafos

Particionar (ou cortar) um grafo conexo se refere ao processo de dividí-lo em subgrafos obedecendo alguma propriedade. O corte em grafos é amplamente utilizado em segmentação de imagens, sendo a imagem representada por um grafo ponderado (SHI E MALIK, 2000; WENBING *et al.*, 2008), também conhecido como grafo de similaridade ou de distância. Em um grafo de similaridade ou de distância, os nós são associados a *pixels* ou a regiões (conjuntos de *pixels*) da imagem. Os pesos das arestas podem representar a similaridade, onde valores maiores significam maior similaridade ou proximidade entre os nós; ou a distância, onde valores maiores significam maior distância ou dissimilaridade entre os nós. Neste trabalho, as principais técnicas de segmentação citadas utilizam o grafo de similaridade.

Dado um grafo ponderado  $G$ , particionado em dois subgrafos  $A$  e  $B$ , uma forma de medir a qualidade do particionamento é utilizando a métrica de corte, definida como (WU E LEAHY, 1993; SHI E MALIK, 2000; COSTA *et al.*, 2007)

$$\text{Cut}(A, B) = \sum_{u \in A, v \in B} w(u, v). \quad (2.4)$$

Note que em um grafo de similaridade quanto menor o valor do corte, melhor é a qualidade do particionamento (WU E LEAHY, 1993). Similarmente, porém de forma inversa, em um grafo de distância quanto maior o valor do corte, melhor é a qualidade do particionamento.

### 2.2.2 Corte normalizado

Wu e Leahy (1993) observaram que o critério de corte mínimo favorece ao desbalanceamento das ordens dos subgrafos gerados no particionamento, já que o valor do corte dado pela Equação 2.4 cresce com o aumento do número de arestas removidas. Para minimizar esse problema, Shi e Malik (2000) propuseram o **Corte Normalizado** (NCut — do inglês *Normalized Cut*). O valor do corte normalizado para o particionamento de um grafo  $G$  em dois subgrafos  $A$  e  $B$  é dado por

$$\text{NCut}(A, B) = \frac{\text{Cut}(A, B)}{\text{Assoc}(A, G)} + \frac{\text{Cut}(A, B)}{\text{Assoc}(B, G)}, \quad (2.5)$$

onde  $\text{Assoc}(A, G) = \sum_{u \in A, v \in G} w(u, v)$ .

Análogo ao corte, em um grafo de similaridade quanto menor o valor do NCut, melhor é a qualidade do particionamento (SHI E MALIK, 2000). Assim, o particionamento ótimo de um grafo de similaridade  $G$  em dois subgrafos  $A$  e  $B$  é aquele cujo valor  $\text{NCut}(A, B)$  é mínimo. Porém, encontrar tal particionamento é um problema de complexidade NP-Completo.

### Aproximando o valor ótimo de corte normalizado

Embora seja inviável encontrar o particionamento de um grafo  $G$  em dois subgrafos  $A$  e  $B$  tal que o valor  $\text{NCut}(A, B)$  seja mínimo, Shi e Malik (2000) descobriram que, de forma computacionalmente viável, pela expansão da Equação 2.5, é possível encontrar um particionamento do grafo cujo valor de corte normalizado seja próximo do mínimo. A solução proposta por Shi e Malik (2000) explora as propriedades do espectro do grafo, como descritas por Fiedler (1975).

O estudo da *Teoria Espectral dos Grafos* (TEG) (CVETKOVIC *et al.*, 2010) basicamente se caracteriza pelo relacionamento entre as propriedades algébricas do espectro de certas matrizes associadas a um grafo. O espectro de um grafo  $G$  é o conjunto dos seus autovalores, geralmente apresentados em ordem decrescente, não necessariamente no sentido estrito, associado às suas respectivas multiplicidades algébricas (ABREU, 2005).

Seja  $A$  uma matriz quadrada de ordem  $n$ . Um autovalor de  $A$  é um escalar  $\lambda$  tal que  $AX = \lambda X$ , para  $X \neq 0$ ,  $X \in \mathbb{R}^n$ . Todo  $X$  que satisfaça esta equação é denominado um autovetor de  $A$  associado ao autovalor de  $\lambda$ .

A TEG pode ser aplicada à segmentação de imagens representadas como grafos de similaridade. A segmentação é feita com base na análise dos autovetores e autovalores calculados para as matrizes laplacianas geradas a partir dos grafos de similaridade. Algumas das técnicas de particionamento de grafos baseadas em TEG são o *corte médio* (*average cut*), que é considerado o método padrão de particionamento pela TEG (GUATERRY E MILLER, 1995); a *associação média* (*average association*), em que o corte do grafo é realizado através do autovetor associado ao maior autovalor e amplamente trabalhada por Sarkar e Boyer (1996); e o *corte normalizado*, cujo algoritmo fundamentado na TEG foi proposto por Shi e Malik (2000).

O diagrama apresentado na Figura 2.1 ilustra as etapas do processo de segmentação de uma imagem  $I$  pelo algoritmo de corte normalizado fundamentado na TEG. Como é possível observar, cria-se um grafo de similaridade  $G = (V, E, W)$  utilizando algum modelo, como os que serão descritos no Capítulo 3. Em seguida, obtém-se a matriz Laplaciana  $L(G) = D(G) - W(G)$  a partir do grafo  $G$ .

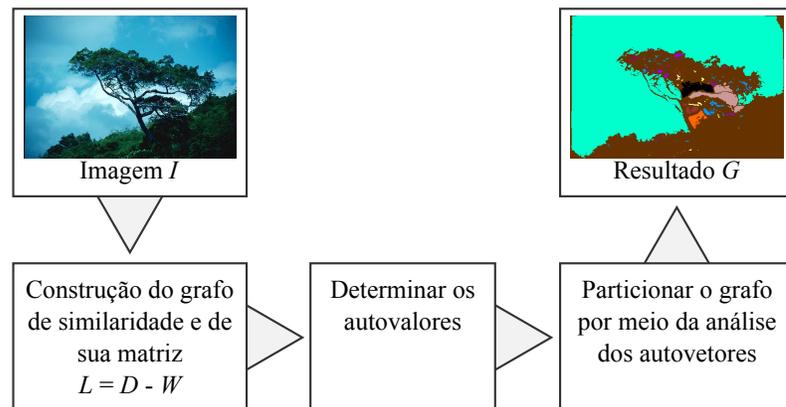


Figura 2.1: Etapas do algoritmo de corte normalizado baseado na TEG aplicado à segmentação de imagens. Adaptado de (CARVALHO E COSTA, 2012).

Então, calcula-se os autovetores pela resolução da equação  $Lv = \lambda Dv$ . Por meio da análise dos autovetores obtidos é possível determinar a qual partição pertence cada nó do grafo de similaridade.

Finalmente, basta fazer o mapeamento do grafo para a imagem para obter as partições (SHI E MALIK, 2000; CARVALHO E COSTA, 2012).

Observe que o algoritmo proposto por Shi e Malik (2000) pode ser aplicado em dois modos de operação:

1. *2-way*, em que cada região da imagem, iniciando pela região  $R$  definida pela própria imagem, é recursivamente particionada em duas novas regiões, por meio da análise do autovetor associado ao segundo menor autovalor;
2. *k-way*, em que o particionamento é realizado diretamente em um dado número  $k$  de regiões previamente especificado. Nesse modo é necessário o cálculo de  $k$  autovetores a partir da matriz que representa o grafo, sendo que as regiões são determinadas por uma análise de todos os  $k$  autovetores calculados.

Em todos os experimentos neste trabalho que envolveram a aplicação do algoritmo de corte normalizado baseado na TEG, o modo de operação utilizado foi o *k-way*.

### 2.3 *Benchmark* de segmentação de imagens da Berkeley

O *benchmark* de segmentação de imagens **BSDS500** da Berkeley (ARBELÁEZ *et al.*, 2011) é uma ferramenta desenvolvida para apoiar a avaliação quantitativa da qualidade dos resultados produzidos por um algoritmo de segmentação. Ele é composto por 500 imagens naturais, *i. e.*, imagens com assuntos encontrados na natureza: urbana (pessoas, edificações, veículos, etc); ou rural e selvagem (paisagens, vegetação, animais, plantações, etc). Para cada uma das 500 imagens há também um conjunto de segmentações realizadas manualmente por humanos. Essas segmentações servem como base de comparação para determinar a pontuação dos algoritmos. As imagens do *benchmark* são organizadas em três conjuntos:

1. de **treinamento**, composto por 200 imagens. Esse conjunto deve ser utilizado na etapa de ajuste dos algoritmos de segmentação;
2. de **validação**, composto por 100 imagens. Esse conjunto destina-se ao suporte de desenvolvimento de algoritmos que possuem uma etapa de aprendizado. Quando o algoritmo não precisa de uma etapa de aprendizado, esse conjunto pode ser ignorado ou utilizado como um conjunto de teste complementar, como é feito neste trabalho;

3. de *teste*, composto por 200 imagens. Esse conjunto deve ser utilizado somente quando o algoritmo está pronto e ajustado, definindo a pontuação definitiva do algoritmo.

Dentre as métricas disponibilizadas pelo BSDS500, três estão sendo utilizadas neste trabalho. Uma delas avalia a segmentação com base das bordas das regiões, gerando uma pontuação dada pela *medida-f*

$$F = 2 \cdot \frac{P \cdot R}{P + R}, \quad (2.6)$$

onde  $P = \frac{TP}{TP+FP}$  e  $R = \frac{TP}{TP+FN}$  são respectivamente medidas de *Precisão* e de *Revocação* (DAVIS E GOADRICH, 2006; ARBELÁEZ *et al.*, 2011).  $TP$  é o número de verdadeiros positivos, *i. e.*, o número de *pixels* preditos como pertencentes a uma borda que de fato pertencem a uma. Analogamente,  $FP$  é o número de falsos positivos e  $FN$  é o número de falsos negativos.

As outras duas métricas avaliam a segmentação com base nas regiões. Uma delas é a *cobertura de região* (ARBELÁEZ *et al.*, 2011) definida como

$$C(S' \rightarrow S) = \frac{1}{N} \sum_{R \in S} |R| \cdot \max_{R' \in S'} O(R, R'), \quad (2.7)$$

onde  $O(R, R') = \frac{|R \cap R'|}{|R \cup R'|}$  é a sobreposição entre duas regiões  $R$  e  $R'$ ;  $S$  e  $S'$  são partições da imagem; e  $N$  é o número total de *pixels* na imagem. A outra é o *Probabilistic Rand Index* (PRI) (ARBELÁEZ *et al.*, 2011), definido como

$$PRI(S, \{G_k\}) = \frac{1}{T} \sum_{i < j} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})], \quad (2.8)$$

onde  $S$  é uma partição da imagem;  $\{G_k\}$  é um conjunto de segmentações da base de comparação (ground-truth);  $c_{ij}$  é o evento onde os *pixels*  $i$  e  $j$  possuem o mesmo rótulo e  $p_{ij}$  é a sua probabilidade; e  $T$  é o total de pares de *pixels*.

## 2.4 Contagem de bactérias em placas

Quando se fala em crescimento microbiano, o que realmente interessa é a população, e não o tamanho dos indivíduos (TORTORA *et al.*, 2005). Os microrganismos se reproduzem, aumentando seu número, e se agrupam em colônias que podem conter milhões ou bilhões de células. Esses agrupamentos podem ser visualizados sem que seja necessária a utilização de um microscópio.

Um método para quantificar diretamente o crescimento microbiano é a *contagem em placa de Petri* (TORTORA *et al.*, 2005). É, também, o mais utilizado na determinação do tamanho de uma população bacteriana. A vantagem da contagem em placa é ser possível contar os microrganismos viáveis, ou seja, aqueles que são capazes de se reproduzir e formar uma colônia, as Unidades Formadoras de Colônia (UFC). Observe que, em geral, uma única bactéria pode gerar uma colônia, uma vez que sua reprodução ocorre por um processo de divisão celular. Uma placa de Petri é um recipiente transparente de vidro ou de plástico que provê um ambiente para cultivo de microorganismos livre de agentes externos.

Para realizar a contagem em placa é essencial que somente um número limitado de colônias cresça em cada ensaio. Em placas muito saturadas de colônias poderá ocorrer o impedimento do crescimento de algumas UFC's. Isso conduzirá a uma contagem errônea da população microbiana. Para garantir que o número de colônias fique numa faixa confiável, é utilizado o método da *diluição seriada*, em que 1 mL de amostra do meio em análise, denominado inóculo inicial, é diluído em um tubo com 9 mL de água esterilizada. Depois, novas diluições podem ser feitas similarmente utilizando-se 1 mL da última diluição. A Figura 2.2 ilustra o processo de diluição seriada.

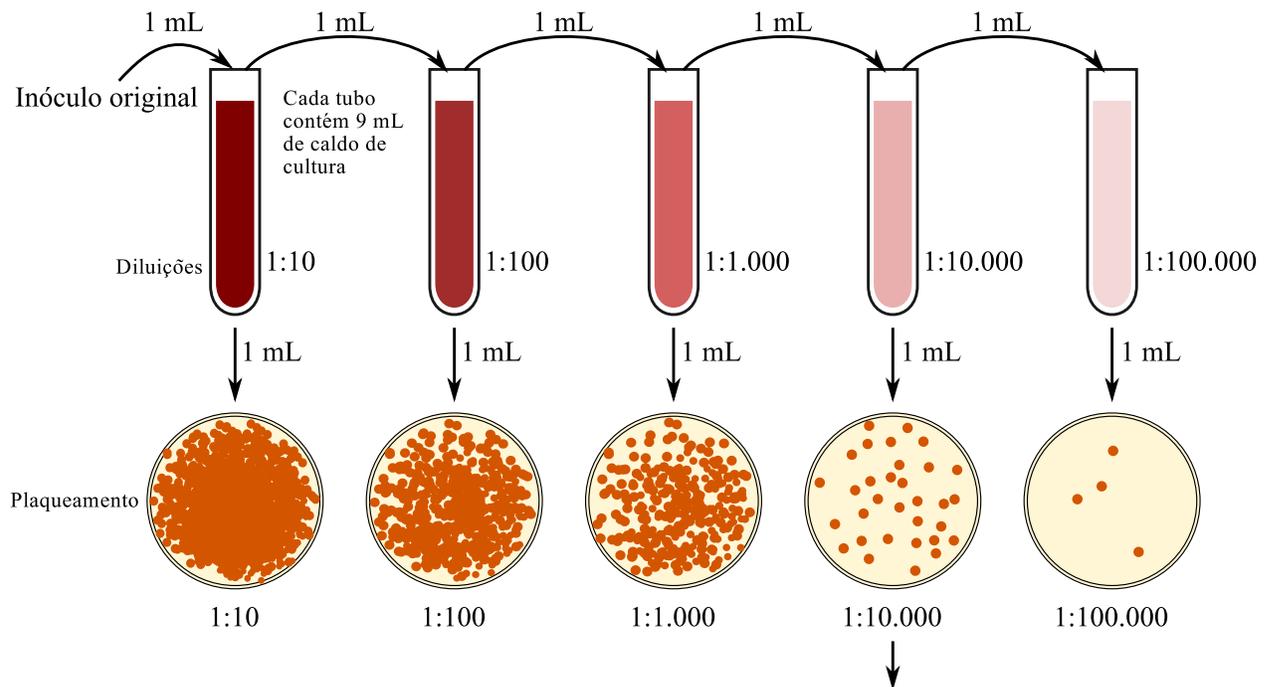


Figura 2.2: Ilustração da diluição seriada para contagem de colônias de bactérias em placas de Petri. Observe que neste exemplo a diluição ideal para ser contada é a 1:10.000 (TORTORA *et al.*, 2005).

O crescimento das bactérias em placa ocorre em meio de cultivo sólido, denominado agar. Há dois métodos para preparar uma placa, o *pour plate* e o *espalhamento*. No *pour plate*, o inóculo

é colocado diretamente sobre a placa e o agar fundido, a uma temperatura aproximada de 50° C, é colocado sobre a placa, cobrindo o inóculo. No método do espalhamento, o inóculo é colocado sobre o agar já solidificado. O método *pour plate* possui as desvantagens de possibilitar a destruição de algumas UFC's em decorrência da temperatura do agar fundido, e de possibilitar o crescimento das colônias dentro do meio de cultivo. Em geral, as características visuais das colônias de bactérias não são discriminativas para sua classificação. Quando é necessário verificar se no ensaio existe um tipo específico de microrganismo, utilizar-se um meio de cultivo seletivo. Contudo, em alguns casos, a identificação do tipo de microrganismo depende de características visuais que as colônias produzem na *superfície* do meio seletivo.

### 2.4.1 Contagem automática

Atualmente, há diversas soluções para o problema de contagem automática de colônias de bactérias. Em geral, são sistemas que possuem seus próprios dispositivos de imageamento, capazes de registrar imagens em alta definição a fim de otimizar a eficácia do contador. Contudo, esses sistemas são inviáveis para grande parte dos laboratórios devido a seus altos preços. Adicionalmente, alguns contadores exigem que o cultivo das bactérias seja feito usando um substrato fluorogênico. Esses sistemas são extremamente sensíveis e são adequados para detectar micro colônias, mas impõem um custo adicional relacionado ao meio de cultivo. Além disso, alguns contadores automáticos ainda exigem que o usuário especifique a região de interesse e os parâmetros antes de realizar a enumeração das colônias. Nesses casos, operários humanos ainda precisam estar altamente envolvidos na operação de contagem (CHEN E ZHANG, 2009).

Recentemente, porém, estão surgindo sistemas de contagem automática baseados em *software* (CHEN E ZHANG, 2009; CLARKE *et al.*, 2010). Essas soluções não exigem equipamentos de imageamento específicos, sendo possível utilizar câmeras fotográficas digitais comuns ou *scanners*, como os utilizados em escritórios. Naturalmente, esses novos sistemas representam uma opção de menor custo aos laboratórios por não exigirem o *hardware* de custo elevado. Contudo, espera-se também que as imagens a serem processadas sejam de menor qualidade que nos sistemas tradicionais. Assim, os algoritmos desses sistemas baseados em *software* devem ser robustos o suficiente para compensar a perda de qualidade no processo de imageamento.

Chen e Zhang (CHEN E ZHANG, 2009) propuseram um contador automático baseado em *software* que é capaz de processar imagens coloridas e monocromáticas, como pode ser observado no diagrama apresentado na Figura 2.3.

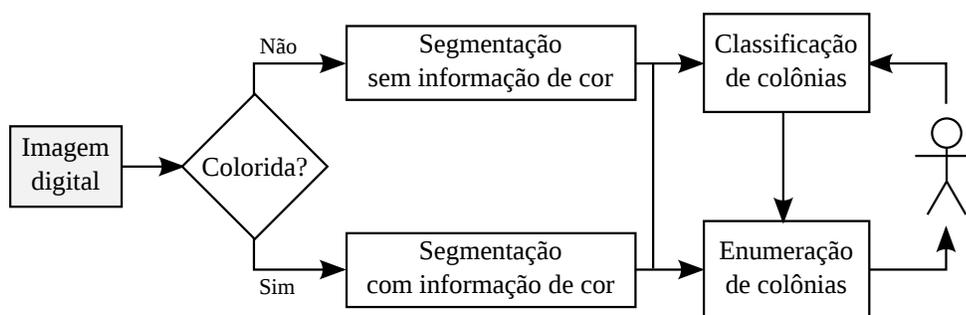


Figura 2.3: Diagrama do sistema de contagem automática de colônias de bactérias proposto por Chen e Zhang. Adaptado de Chen e Zhang (2009).

No sistema proposto por Chen e Zhang, a etapa de segmentação para ambos os tipos de imagem inicia com a definição automática da região de interesse, que é a área da placa. Em seguida, é realizada a segmentação das colônias. Para eliminar ruído, as regiões obtidas são submetidas a um teste com o objetivo de identificar se elas são colônia ou não colônia. Finalmente, as colônias agrupadas são submetidas a um processo de separação utilizando a técnica *Watershed* (CHEN E ZHANG, 2009).

Após realizada a etapa de segmentação, as colônias obtidas (regiões) são enumeradas e contadas. Nesse ponto, o usuário pode ainda selecionar amostras representativas de um tipo específico de colônia. O sistema então utiliza essas amostras para classificar as colônias e contar apenas as do tipo escolhido pelo usuário.

Em um outro sistema mais simples, proposto por Clarke *et al.* (CLARKE *et al.*, 2010), as colônias são identificadas por um processo de limiarização. Os mínimos regionais são utilizados para auxiliar na separação de colônias agrupadas. Apesar de eficiente, esse sistema parece ser mais sensível a ruído e, conseqüentemente, menos robusto.

## Capítulo 3

# Modelos de grafos de similaridade

O modelo do grafo de similaridade representa um papel importante na segmentação de imagens por meio do *framework* do corte normalizado, tendo influência diretamente sobre o resultado da segmentação (SHI E MALIK, 2000; COUR *et al.*, 2005; CARVALHO E COSTA, 2012). O modelo do grafo é capaz de representar a imagem sob a óptica de diferentes conceitos importantes como a multiescala (COUR *et al.*, 2005) e a hierarquia (CARVALHO *et al.*, 2010A,B). Ele pode também estabelecer a relação entre regiões da imagem (agrupamentos de *pixels*), ao invés de apenas *pixels* isolados. Estas regiões podem ser obtidas por meio da aplicação de outras técnicas de segmentação como o *Watershed* (MA E WAN, 2008; CARVALHO *et al.*, 2009) e a limiarização (CARVALHO *et al.*, 2010A). Isto possibilita a utilização de descritores de região como, por exemplo, forma, área e textura, para determinar a similaridade entre os nós do grafo. Mais ainda, a pré-segmentação por meio de outras técnicas pode conferir características interessantes ao método, que incluem redução do custo computacional e melhor qualidade na segmentação.

Neste capítulo são apresentados quatro modelos de grafo de similaridade: o modelo *pixel grid* (SHI E MALIK, 2000), o modelo *pixel grid* em multiescala (COUR *et al.*, 2005), o modelo baseado na árvore de componentes (CARVALHO *et al.*, 2010A) e o modelo baseado nas regiões homogêneas da *quadtrees* (CARVALHO *et al.*, 2010B). Os dois últimos modelos utilizam estruturas hierárquicas na montagem do grafo. Neste trabalho, eles receberam modificações para solucionar alguns problemas que limitavam suas aplicações e também para melhorar a qualidade das segmentações, especialmente o modelo fundamentado na *quadtrees*.

É também realizado um teste comparativo para avaliar o desempenho dos modelos, considerando a qualidade da segmentação e também a eficiência computacional. Para tanto, foi utilizado o *benchmark* de segmentação de imagens **BSD500**, da Berkeley (ARBELÁEZ *et al.*, 2011).

## 3.1 Modelos baseados na grade de *pixels*

Neste trabalho, denomina-se *pixel grid* o modelo de grafo de similaridade fundamentado na grade de *pixels* da imagem. Ele foi o primeiro modelo utilizado pelo *framework* de segmentação de imagens por corte normalizado (SHI E MALIK, 2000). Seja  $G = (V, E, W)$  o grafo de similaridade gerado por meio do modelo *pixel grid*. Cada *pixel* da imagem está representado por um nó  $u \in V$ . Dois nós  $u$  e  $v \in V$  estão conectados por uma aresta  $e(u, v) \in E$  se a distância entre seus *pixels* correspondentes não for maior que um dado raio de conexão  $r \in \mathbb{R}_{\geq 1}$ . Em geral, raios de conexão maiores produzem segmentações de melhor qualidade (COUR *et al.*, 2005). Todavia, raios de conexão longos podem produzir regiões desconexas. Isto pode ser evitado utilizando funções de similaridade adequadas. Porém, a produção de regiões desconexas pode ser um requisito para algumas aplicações.

Este modelo gera um grafo de ordem igual ao número de *pixels* da imagem. Portanto, quanto maior for a imagem, maior será ordem do grafo, podendo inviabilizar algumas aplicações que utilizam imagens muito grandes. Este problema torna-se mais grave se é necessário utilizar um raio de conexão também grande, pois desta forma aumenta a densidade do grafo e, conseqüentemente, o seu tamanho. De fato, se for exigido um raio de conexão muito grande pela aplicação, o tamanho das imagens poderá ser muito limitado, dependendo da quantidade de memória disponível.

### 3.1.1 Multiescala

Cour *et al.* (2005) introduziram um modelo de grafo de similaridade também fundamentado na grade de *pixels* da imagem. Contudo, o modelo gera o grafo a partir de diferentes escalas da imagem, utilizando raios de conexão menores para escalas mais finas e raios de conexão maiores para escalas mais rústicas. Cada escala da imagem é obtida por meio da amostragem dos *pixels* em uma distância  $\rho(s) = (2r + 1)^{s-1}$ , onde  $r \in \mathbb{R}_{\geq 1}$  é o raio de conexão. Observe que a primeira escala  $s = 1$  é a mais fina, e corresponde à imagem original.

Para cada escala da imagem um subgrafo  $G_s$  é criado. Uma aresta conecta dois nós se a distância entre seus *pixels* correspondentes não for maior que  $\rho(s)r$ . O grafo de similaridade final é dado por  $G = G_1 \cup G_2 \cup \dots \cup G_s$ , onde  $s \in \mathbb{N}_{\geq 1}$  é o número de escalas (COUR *et al.*, 2005). Note que a ordem do grafo gerado por este modelo é ainda maior que a ordem do grafo gerado pelo modelo *pixel grid*. Contudo, ele permite reduzir o tamanho do raio de conexão sem prejudicar a qualidade da segmentação, resultando em um grafo bastante esparso para reduzir o custo computacional.

## 3.2 Modelos fundamentados em estruturas hierárquicas

Os modelos baseados na árvore dos componentes (CT, do inglês *Component Tree*) (CARVALHO *et al.*, 2010A) e na *quadtree* (CARVALHO *et al.*, 2010B) foram introduzidos em trabalhos anteriores do autor. Contudo, alguns detalhes dos modelos impunham limitações quanto ao tamanho das imagens que podiam ser utilizadas. Neste trabalho, esses modelos foram modificados para eliminar tais limitações, possibilitando o processamento de imagens maiores. No caso do modelo baseado na *quadtree*, além da remoção da limitação a imagens quadradas com lado  $2^n$ , onde  $n \in \mathbb{N}_{\geq 1}$ , também está sendo proposta uma forma de balanceamento dos pesos das arestas que melhora consideravelmente a qualidade da segmentação.

### 3.2.1 Modelo baseado na árvore dos componentes

A *árvore dos componentes* é uma hierarquia de componentes conexos de diferentes limiares de uma imagem (MOSOROV E KOWALSKI, 2002; NAJMAN E COUPRIE, 2006). Seja  $F$  uma imagem digital em escala de cinza. Há um limiar

$$F_k(x) = \begin{cases} 1, & \text{se } F(x) \geq k \\ 0, & \text{caso contrário} \end{cases}, \quad (3.1)$$

para cada nível de cinza  $k \in \{F(x)\}$ , onde  $x \in F$ . Denomina-se  $F_k$  uma seção transversal, ou simplesmente seção,  $k$  da imagem  $F$ . A Figura 3.1 ilustra uma imagem  $F$  em escala de cinza, e suas seções  $F_k$ , onde  $k \in \{1, 2, 3, 4, 5\}$ .

Há uma relação de inclusão entre componentes conexos em seções subsequentes da imagem (NAJMAN E COUPRIE, 2006). A construção da árvore dos componentes fundamenta-se nessa observação. Na árvore dos componentes clássica  $CT = (V_{CT}, E_{CT})$ , uma aresta  $e(u, v)$  conecta dois nós  $u$  e  $v \in V_{CT}$ , que são respectivamente associados aos componentes conexos  $CC_a \in F_a$  e  $CC_b \in F_b$ , onde  $a > b$ , se  $CC_a \subseteq CC_b$ . O componente conexo da seção transversal  $F_{\min(k)}$  corresponde a todo o domínio da imagem, sendo ele associado à raiz da árvore  $CT$ .

Observe que as regiões determinadas pelos *pixels* de  $F_k$  com valor zero não são considerados componentes conexos pela definição clássica da árvore dos componentes e, portanto, exceto pela raiz, a união das regiões representadas pelos nós de um dado nível da árvore não formam uma representação completa do domínio da imagem. Porém, uma representação completa do domínio da imagem em cada nível da árvore é um requisito do modelo. Desta forma, foi definida a árvore dos componentes reversa (CARVALHO *et al.*, 2010A), que é similar à árvore dos componentes clássica,

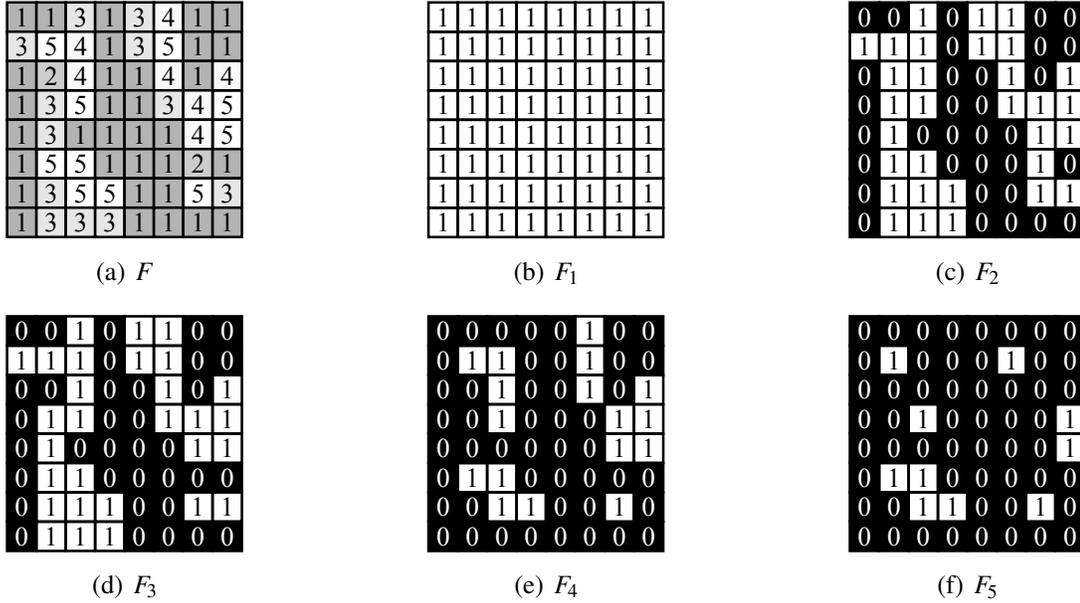


Figura 3.1: Limiarização da imagem. Figura 3.1(a), uma imagem  $F$ ,  $8 \times 8$ , em escala de cinza. Figura 3.1(b)–Figura 3.1(f), seções  $F_k$  de  $F$ .

mas construída a partir dos componentes conexos do complemento  $\bar{F}_k$  das seções transversais. Seja a árvore dos componentes reversa  $RCT = (V_{RCT}, E_{RCT})$ . Uma aresta  $e(u, v)$  conecta dois nós  $u$  e  $v \in V_{RCT}$ , que são respectivamente associados aos componentes conexos  $CC_a \in \bar{F}_a$  e  $CC_b \in \bar{F}_b$ , onde  $a < b$ , se  $CC_a \subseteq CC_b$ . Os componentes conexos da seção transversal complementar  $\bar{F}_{\max(k)}$  irão definir as raízes das árvores dos componentes reversas. Assim, em alguns casos será gerada uma floresta de árvores dos componentes reversas. A Figura 3.2 ilustra as árvores  $CT$  e  $RCT$  geradas a partir da imagem  $F$  apresentada na Figura 3.1(a). Observe que não há componente conexo  $CC_{RCT} \in \bar{F}_{\min(k)}$ .

O modelo baseado na árvore dos componentes explora a relação de hierarquia dos componentes conexos em diferentes seções da imagem. A idéia é manter componentes conexos que estão em diferentes seções, mas que são muito similares, na mesma partição. Desta forma, espera-se que a relação de hierarquia influencie a segmentação de cada uma das seções transversais individualmente. Os passos para gerar um grafo de similaridade a partir de uma imagem usando o modelo baseado na árvore dos componentes são apresentados no diagrama da Figura 3.3.

O passo *pré-processamento* consiste na aplicação de filtros na imagem como, por exemplo, redução de ruído para reduzir a quantidade de componentes conexos em cada seção transversal  $F_k$ . A profundidade de cor da imagem também pode ser reduzida nesse passo. Assim, a imagem pré-processada vai requerer menos esforço computacional que a imagem original.

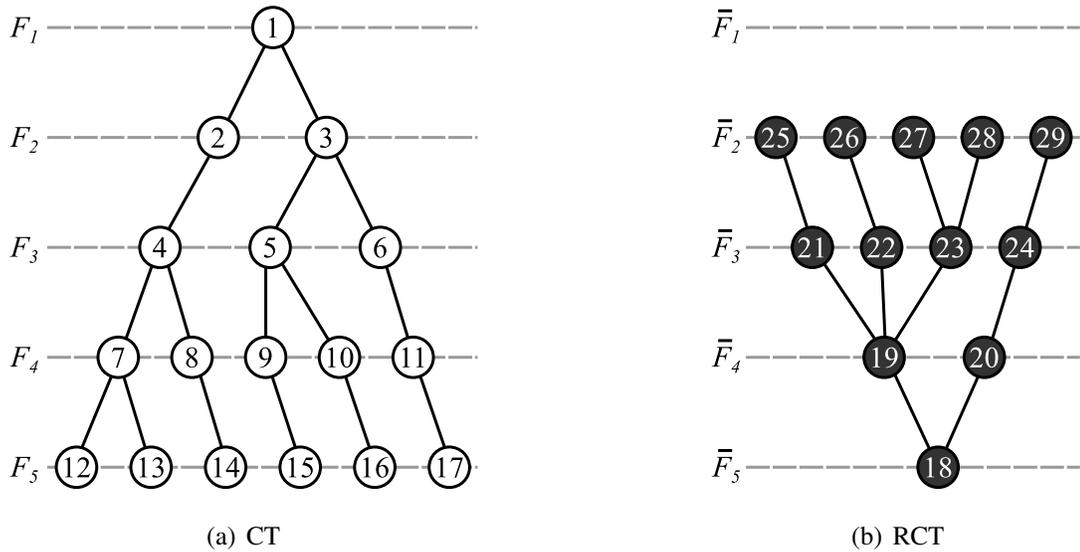


Figura 3.2: Representação da árvore dos componentes e da árvore dos componentes reversa geradas a partir da imagem  $F$ , apresentada na Figura 3.1(a). Figura 3.2(a), árvore gerada com os componentes conexos das seções  $F_k$ . 3.2(b), árvore reversa gerada com os componentes conexos das seções complementares  $\bar{F}_k$ .

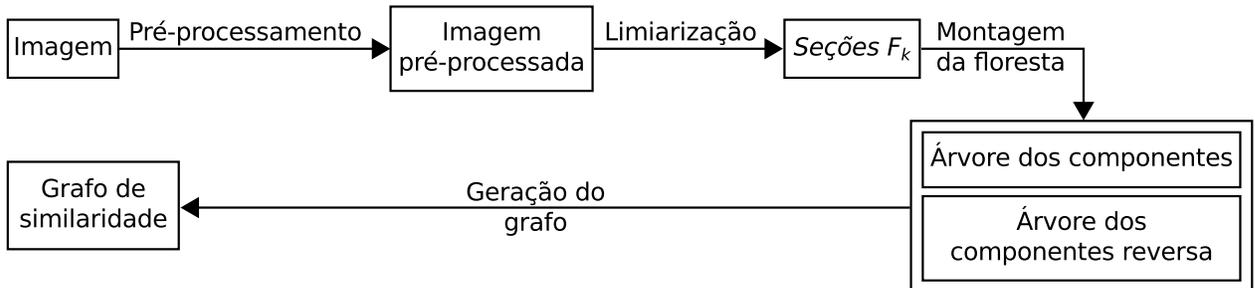


Figura 3.3: Passos para gerar um grafo de similaridade pelo modelo baseado na árvore dos componentes a partir de uma imagem.

No passo de *limiarização*, todas as seções  $F_k$  da imagem  $F$  são calculadas, onde  $k \in \{F(x) : x \in F\}$ . Depois disso, os componentes conexos em  $F_k$  e em  $\bar{F}_k, \forall k$ , são rotulados.

No passo *montagem da floresta*, a árvore dos componentes e a árvore dos componentes reversa são criadas. Um nó é associado a cada componente conexo em  $F_k$  e em  $\bar{F}_k, \forall k$ . Então, cada nó  $u$  é associado ao seu antecessor  $v$ , conforme as regras estabelecidas previamente nesta seção, definindo a aresta  $e(u, v) \in E_{antecessores}$ .

No passo final, **geração do grafo**, dois nós  $u$  e  $v$  são conectados por uma aresta  $e(u, v) \in E_{vizinhos}$  se a distância entre os centróides dos seus componentes conexos correspondentes não for maior que o raio

$$r' = \frac{\max(W(B_u) + W(B_v), H(B_u) + H(B_v))}{2} + r, \quad (3.2)$$

onde  $r \in \mathbb{R}_{\geq 1}$ ;  $B_u$  e  $B_v$  são, respectivamente, os retângulo de delimitadores do componentes conexos associados aos nós  $u$  e  $v$ ; e  $W(B)$  e  $H(B)$  são a largura e a altura do retângulo delimitador  $B$ , respectivamente. Na implementação realizada neste trabalho, o centróide de um componente conexo é definido como o centro do seu retângulo delimitador. Finalmente, os pesos para as arestas são calculados por meio de funções de similaridade apropriadas. Observe que diferentes funções de similaridade podem ser utilizadas para as arestas pertencentes aos conjuntos  $E_{antecessores}$  e  $E_{vizinhos}$ .

### 3.2.2 Modelo baseado na *quadtree*

O termo *quadtree* refere-se a uma classe de estruturas de dados hierárquicas determinadas pela decomposição recursiva do espaço (SAMET, 1984). Em PDI, uma *quadtree* usualmente mapeia regiões de uma imagem em uma árvore onde todo nó não folha possui exatamente quatro filhos. Nesses casos, a região inicial costuma ser todo o domínio da imagem, e é associada ao nó raiz da árvore. Então, recursivamente, cada região que não satisfaz um dado critério de homogeneidade é decomposta em quatro novas regiões disjuntas. A Figura 3.4 apresenta um esquema de decomposição para uma imagem binária, onde as regiões devem com ter apenas elementos com valor zero ou apenas elementos com valor um, para serem consideradas homogêneas.

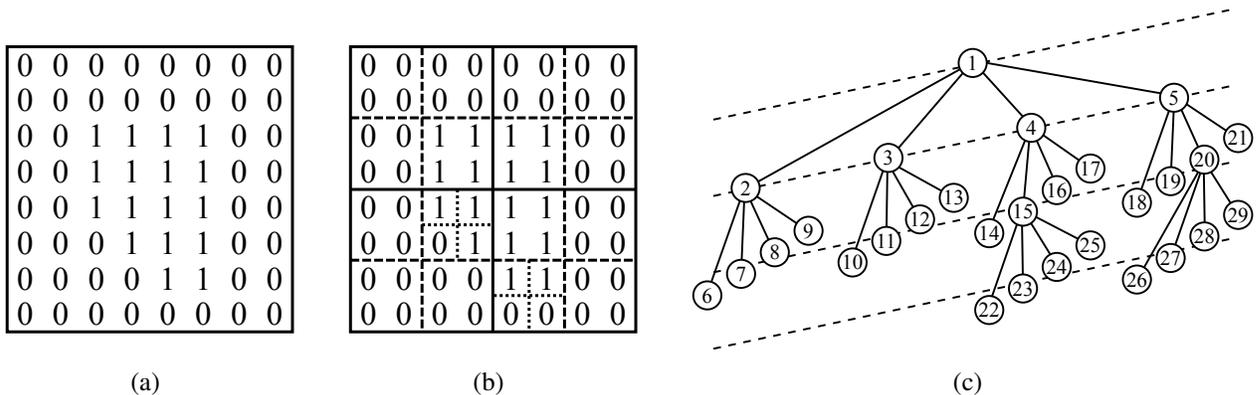


Figura 3.4: Geração de uma *quadtree* a partir de uma imagem binária. Figura 3.4(a), uma imagem binária  $F$  com tamanho  $8 \times 8$ . Figura 3.4(b), a decomposição de  $F$  em regiões homogêneas. Figura 3.4(c), a representação da estrutura da *quadtree* gerada.

Na prática, respeitar a regularidade da *quadtree* limita a aplicação do modelo a imagens quadradas com lado  $2^n$ ,  $n \in \mathbb{N}_{\geq 1}$ . A solução adotada quando a imagem não satisfaz esses requisitos quanto ao tamanho foi criar várias árvores com tamanhos adequados para preencher todo o domínio da imagem. Uma forma de fazer isso é encaixar a maior *quadtree* regular possível no espaço da imagem original, e depois repetir o processo para todos os espaços restantes até que não haja mais espaços residuais. Uma vez que o modelo utiliza apenas as folhas da *quadtree* para gerar o grafo de similaridade, descartando sua estrutura, não é necessário reunir as diferentes árvores em uma única.

O modelo baseado na *quadtree* foi concebido com o propósito de diminuir o custo computacional pela redução da ordem do grafo de similaridade. Para tanto, ele utiliza as regiões associadas às folhas da *quadtree* para gerar o grafo, *i. e.*, regiões consideradas homogêneas. O diagrama apresentado na Figura 3.5 mostra os passos necessários para gerar um grafo de similaridade a partir de uma imagem usando o modelo baseado na *quadtree*.

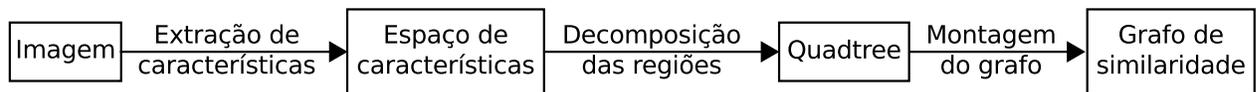


Figura 3.5: Passos para gerar um grafo de similaridade pelo modelo baseado na *quadtree* a partir de uma imagem.

O passo ***extração de características*** têm o propósito de converter os dados da imagem em uma representação adequada ao critério de homogeneidade em uso. Por exemplo, Carvalho e Costa (2012) utilizam um mapeamento de bordas obtido pelo filtro de Canny (CANNY, 1986) como espaço de características. Quando o espaço de características requerido é a própria imagem, esse passo pode ser ignorado.

No passo ***decomposição das regiões*** uma *quadtree* é criada pela decomposição recursiva das regiões da imagem, como descrito na Seção 3.2.2. O critério de decomposição deve ser adequado ao espaço de características definido no passo anterior. Voltando ao exemplo do trabalho de Carvalho e Costa (2012), uma vez que o espaço de características era binário, uma região foi definida como homogênea se todos os seus *pixels* eram pretos ou se todos eram brancos. Esse passo pode incluir também tarefas de pré-processamento como, por exemplo, remoção de ruído.

No último passo, ***montagem do grafo***, um grafo de similaridade é construído a partir da *quadtree*. Como o objetivo principal desse modelo é reduzir a ordem do grafo, somente as folhas da *quadtree* são utilizadas. Assim sendo, dado um grafo de similaridade  $G = (V, E, W)$ , cada folha é associada a um nó  $u \in V$ . Então, dois nós  $u$  e  $v \in V$  são conectados por uma aresta  $e(u, v) \in E$  se existir ao menos um par de *pixels*  $(p_u, p_v)$ , tal que  $p_u$  e  $p_v$  estejam respectivamente contidos

nas regiões associadas aos nós  $u$  e  $v$ , cuja distância entre eles esteja dentro de um dado raio de conexão  $r \in \mathbb{R}_{\geq 1}$ . Observe que o grafo de similaridade gerado poderá ser irregular, *i. e.*, uma rede complexa. Uma vez que a irregularidade dos graus dos nós pode influenciar negativamente a segmentação, propõe-se realizar um balanceamento dos pesos das arestas. Assim, o peso ajustado de uma aresta  $e(u, v) \in E$  é dado por

$$w'(u, v) = w(u, v) \text{Deg}(u, v), \quad (3.3)$$

onde  $w(u, v) \in W$  é o peso calculado para a aresta  $e(u, v)$  por uma função de similaridade qualquer; e  $\text{Deg}(u, v)$  é o número de pares de *pixels*  $(p_u, p_v)$  que definem a aresta  $e(u, v)$ .

### 3.3 Teste de desempenho dos modelos

O *benchmark* da Berkeley **BSDS500** (ARBELÁEZ *et al.*, 2011), disponibilizado recentemente, foi utilizado como base para comparar a qualidade da segmentação. Os testes foram executados nas imagens em escala de cinza. Os resultados são medidos quanto à qualidade das bordas dos segmentos, usando a Equação 2.6, e quanto à qualidade das regiões geradas, usando a Equação 2.7 e a Equação 2.8.

#### 3.3.1 Configuração do experimento

O propósito desse experimento foi avaliar o desempenho da segmentação gerada pela utilização de cada um dos modelos de grafo de similaridade apresentados neste Capítulo. Assim, uma mesma função de similaridade foi utilizada em todos os modelos, definida por Cour *et al.* (2005), em que o peso de uma aresta  $e(u, v)$  é dado por

$$w(u, v) = \exp^{-\max_{x \in \text{Line}(u, v)} \frac{\|\text{Edge}(x)\|^2}{\sigma_C}}, \quad (3.4)$$

onde  $\text{Line}(u, v)$  é um segmento de reta com as extremidades nos *pixels*  $u$  e  $v$ ; e  $\text{Edge}(x)$  é a força da borda na posição  $x$ . Esta função de similaridade é aplicada em um gradiente da imagem.

Especificamente para o conjunto de arestas  $E_{\text{antecessores}}$  do modelo baseado na árvore dos componentes, os pesos são dados por

$$w(u, v) = \exp^{1 - \frac{\max(|u|, |v|)}{\min(|u|, |v|)}}, \quad (3.5)$$

onde  $u$  e  $v$  são os nós conectados por uma aresta  $e(u, v) \in E_{antecessores}$ ; e  $|x|$  denota a área do componente conexo associado ao nó  $x$ .

As imagens foram segmentadas utilizando o algoritmo concebido por Shi e Malik (2000) em modo  $k$ -way. Para se ter uma noção sobre como a quantidade de regiões interfere no processo e na qualidade da segmentação, foram escolhidos  $k = \{10, 30\}$ . A escolha dos valores de  $k$  foi feita arbitrariamente, mesmo sabendo-se que em um cenário de testes ideal cada imagem deveria ser segmentada em um número diferente de regiões, de acordo com seu conteúdo. Contudo, determinar automaticamente o número ideal de regiões para cada imagem é um problema complexo que não estava no escopo deste trabalho.

No modelo *pixel grid*, um dado *pixel* conecta-se aos *pixels* dentro de uma vizinhança definida por  $r = 10$ , com uma taxa de amostragem de 30%. Já o modelo *pixel grid* em multiescala utiliza um raio para cada uma das três escalas utilizadas, os quais são  $r_1 = 2$ ,  $r_2 = 3$  e  $r_3 = 7$ . O modelo baseado na *quadtree* utiliza um raio de conexão  $r = \sqrt{10}$ . E, finalmente, o modelo baseado na árvore dos componentes utiliza um raio de conexão  $r = 10$ .

Na geração da *quadtree*, uma região é considerada homogênea se a diferença máxima do nível de cinza médio ou do desvio padrão entre as quatro novas regiões possíveis não exceda um dado limiar  $t$ , sendo  $t = 16$ . Além disso, regiões maiores que  $16 \times 16$  pixels são diretamente consideradas heterogêneas. O espaço de características é a imagem com uma equalização de histograma.

No passo de pré-processamento do modelo baseado na árvore dos componentes foi aplicado um filtro de mediana com máscara  $3 \times 3$  para reduzir o ruído. Posteriormente, foi aplicada uma equalização de histograma e a profundidade de cor foi reduzida de 8 para 6 *bits*. A redução na profundidade é necessária para reduzir o custo computacional da geração do grafo de similaridade. Uma vez que há informação redundante em várias seções cruzadas, esta operação não produz um impacto significativo nos resultados das segmentações pelo modelo da árvore dos componentes.

As implementações dos modelos fundamentados na *quadtree* e na árvore dos componentes foram feitas em linguagem MATLAB (CHAPMAN, 2010; GILAT, 2012). As implementações dos modelos *pixel grid* e *multiscale pixel grid*, assim como a do algoritmo de corte normalizado por teoria espectral, utilizadas nos experimentos, foram as disponibilizadas publicamente por Cour *et al.* (2005). Os testes foram executados em uma máquina de arquitetura i386, com processador de 2.2 GHz, memória de 3 GB e sistema operacional Linux.

### 3.3.2 Resultados

A Tabela 3.1 apresenta os resultados das segmentações das imagens dos conjuntos de validação e de teste. No caso do modelo baseado na árvore dos componentes, uma vez que ele resulta em múltiplas partições da imagem, a pontuação foi calculada utilizando a melhor partição de cada imagem.

Tabela 3.1: Pontuação geral dos modelos nas imagens dos conjuntos de validação e de teste do *benchmark* BSDS500. As melhores pontuações de cada métrica utilizada estão destacadas em negrito.

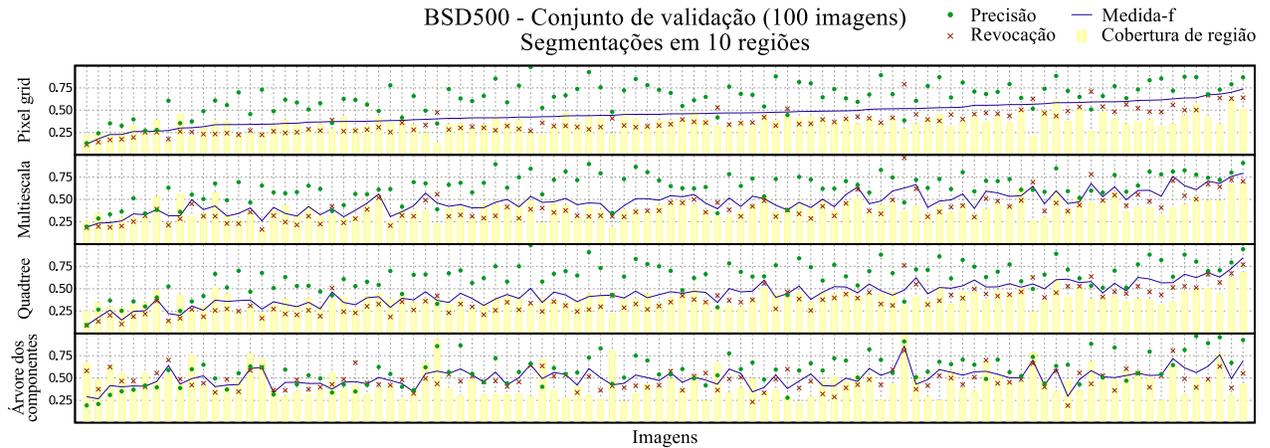
		Conjunto de validação			Conjunto de teste		
		Borda, medida-f	Cobertura de região	PRI	Borda, medida-f	Cobertura de região	PRI
10 regiões	Pixel grid	0.4610	0.3571	0.7236	0.4538	0.3629	0.7430
	Multiescala	0.4756	0.3920	<b>0.7415</b>	0.4805	0.4128	<b>0.7649</b>
	Quadtree	0.4476	0.3841	0.7273	0.4566	0.3897	0.7469
	Árvore dos componentes	<b>0.4997</b>	<b>0.4499</b>	0.6522	<b>0.5053</b>	<b>0.4255</b>	0.6377
30 regiões	Pixel grid	0.5556	0.2455	0.7141	0.5572	0.2541	0.7356
	Multiescala	<b>0.5805</b>	0.3029	<b>0.7263</b>	<b>0.5819</b>	0.3218	<b>0.7501</b>
	Quadtree	0.5584	0.2986	0.7224	0.5584	0.3080	0.7448
	Árvore dos componentes	0.5347	<b>0.4651</b>	0.6832	0.5276	<b>0.4439</b>	0.6661

O modelo baseado na árvore dos componentes obteve as melhores pontuações de borda, medida-f, considerando segmentações em 10 regiões. Contudo, quando a segmentação foi realizada em 30 regiões, o modelo *pixel grid* em multiescala passou a ter as melhores pontuações de borda. Apesar de o modelo baseado na árvore dos componentes ter melhorado a pontuação de borda quando a quantidade de regiões foi aumentada, a taxa de melhora dos outros modelos foi superior.

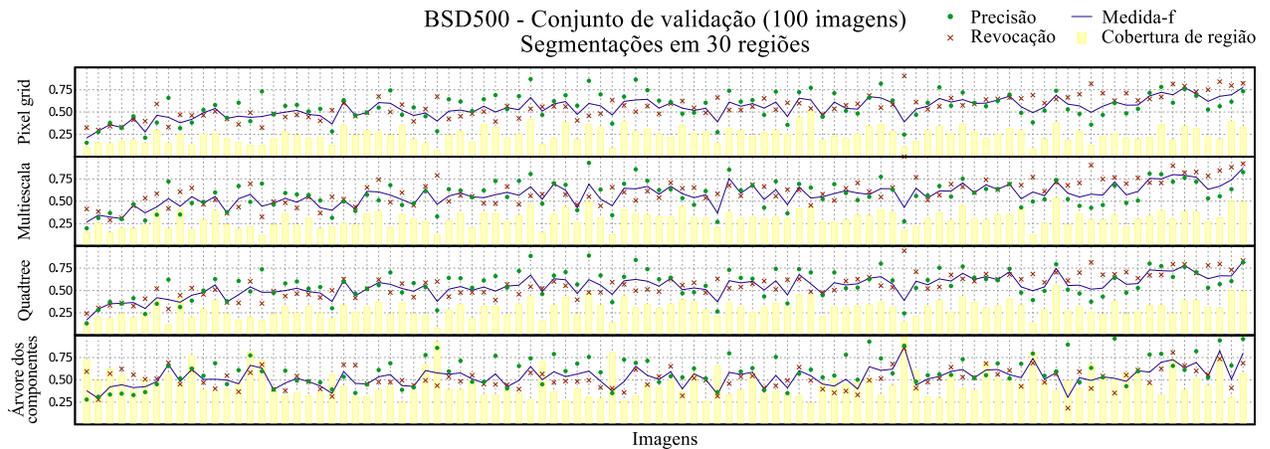
O modelo baseado na árvore dos componentes, além de ter as melhores pontuações para segmentações em 10 e em 30 regiões, foi o único cujas medidas de cobertura de região melhoraram, enquanto as pontuações dos outros modelos pioraram. Esse mesmo comportamento também foi apresentado pelas pontuações obtidas pela métrica PRI, também fundamentada em regiões. Porém, nesse caso foi o modelo *pixel grid* que obteve as melhores pontuações pela métrica PRI, tanto para segmentações em 10 regiões quanto para segmentações em 30 regiões.

A Figura 3.6 e a Figura 3.7 apresentam as pontuações individuais para cada imagem dos conjuntos de validação e de teste do *benchmark* BSDS500, respectivamente. Esses resultados estão sendo apresentados em uma representação gráfica para facilitar a visualização das informações. A métrica baseada nas bordas, medida-f, e uma métrica baseada nas regiões, cobertura de região,

foram utilizadas para compor o gráfico. A precisão e a revocação, que são utilizadas para calcular a medida-f, também estão disponíveis nos gráficos. Observe que as imagens estão ordenadas de acordo com as pontuações da medida-f para segmentações em 10 regiões utilizando o modelo *pixel grid*.



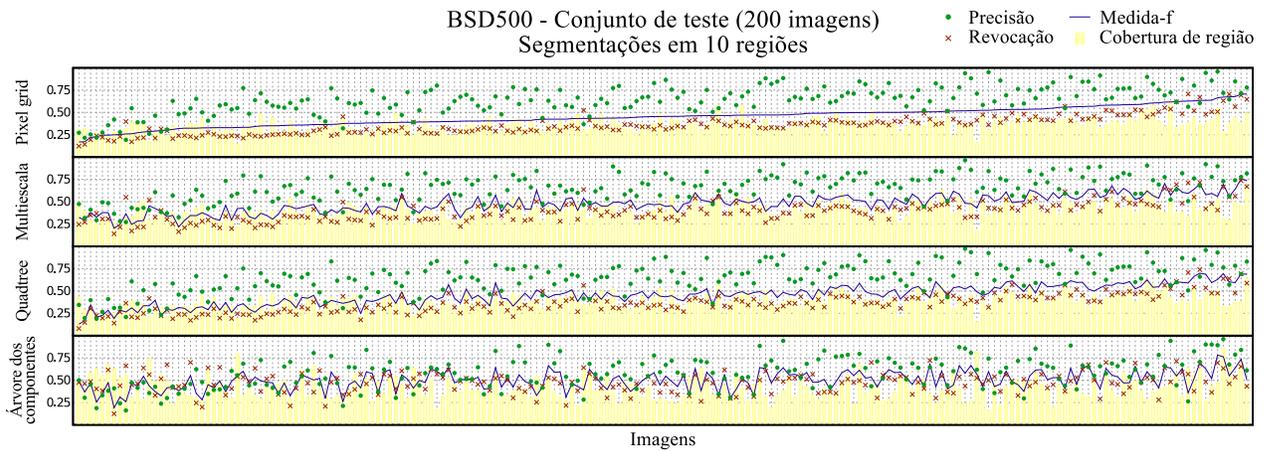
(a) Resultados das segmentações em 10 regiões.



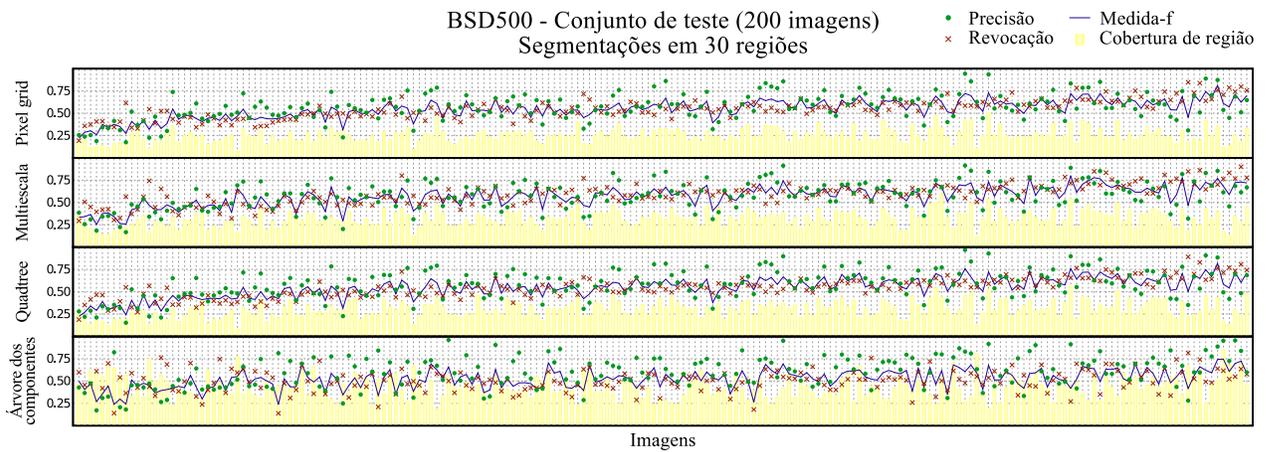
(b) Resultados das segmentações em 30 regiões.

Figura 3.6: Pontuações para cada imagem do conjunto de validação do *benchmark* BSR500, que possui 100 imagens (ARBELÁEZ *et al.*, 2011).

Em concordância com os resultados gerais para o modelo baseado na árvore dos componentes, apresentados na Tabela 3.1, as pontuações para cada imagem individualmente são menos sensíveis à mudança na quantidade de regiões do que no caso dos outros modelos, para ambos os conjuntos de validação e de teste. Isso é devido às características da modelagem pela árvore dos componentes, onde o grafo de similaridade contém várias representações completas do domínio da imagem. Desta forma, um subgrafo pode não conter nós associados a todas as representações e, conseqüentemente, uma representação também pode não participar de todos os subgrafos.



(a) Resultados das segmentações em 10 regiões.



(b) Resultados das segmentações em 30 regiões.

Figura 3.7: Pontuações para cada imagem do conjunto de teste do *benchmark* BSR500, que possui 200 imagens (ARBELÁEZ *et al.*, 2011).

Pode-se também observar nos gráficos da Figura 3.6 e da Figura 3.7 que as melhoras nas pontuações da medida- $f$ , apresentadas pelos modelos *pixel grid*, *pixel grid* em multiescala e *quadtree* para segmentações em 30 regiões, não se aplicam uniformemente para todas as imagens. Enquanto algumas imagens apresentam uma melhora significativa, outras apresentam um resultado pior. Além disso, os resultados para segmentações em 10 regiões mostram maior precisão do que revocação para quase todas as imagens. Porém, para segmentações em 30 regiões, a diferença entre as medidas de precisão e de revocação reduz consideravelmente e, para várias imagens, a revocação passa a ser maior que a precisão.

### 3.3.3 Custo computacional

Um problema relacionado ao *framework* de segmentação de imagens por corte normalizado é o custo computacional. A aproximação do corte normalizado ótimo pelo uso da teoria espectral de grafos possui ordem  $O(n^3)$  para realizar o cálculo de todos os autovetores em um grafo completo com  $n$  nós (SHI E MALIK, 2000). Desta forma, é desejável que o modelo do grafo de similaridade não apenas melhore a qualidade da segmentação, mas também crie meios de reduzir a ordem e o tamanho do grafo. A Tabela 3.2 apresenta os tempos médios que cada modelo levou para calcular as segmentações em 10 e em 30 regiões.

Tabela 3.2: Tempo médio que cada modelo levou para computar as segmentações em 10 e em 30 regiões. Os menores tempos estão destacados em negrito.

	10 regiões			30 regiões			Aumento do tempo médio
	Tempo em segundos		$\frac{\sigma}{\mu}$	Tempo em segundos		$\frac{\sigma}{\mu}$	
	$\mu$	$\sigma$		$\mu$	$\sigma$		
Pixel grid	62.45	$\pm 11.48$	0.1838	138.60	$\pm 48.23$	0.3480	$2.22 \times$
Multiescala	21.38	$\pm 4.09$	0.1913	75.00	$\pm 19.29$	0.2572	$3.51 \times$
Quadtree	<b>17.23</b>	$\pm 7.44$	0.4318	<b>26.59</b>	$\pm 14.56$	0.5476	$1.54 \times$
Árvore dos componentes	71.29	$\pm 55.97$	0.7851	72.47	$\pm 57.43$	0.7925	$1.02 \times$

Os resultados mostram um aumento do custo computacional proporcional ao aumento no número de regiões segmentadas. Uma exceção é o modelo baseado na árvore dos componentes, que teve o custo computacional estável tanto para segmentações em 10 regiões quanto para segmentações em 30 regiões. Isso ocorre por causa da ordem do grafo gerado pelo modelo, que é muito menor que a ordem dos outros modelos. De fato, o aumento do custo computacional ao aumentar a quantidade de regiões é proporcional à ordem do grafo de similaridade. Mesmo assim, o custo computacional para o modelo baseado na árvore dos componentes foi muito alto, causado pelo custo de construção do grafo de similaridade.

O modelo baseado na *quadtree* apresentou o menor custo computacional para segmentações em 10 e em 30 regiões, e o segundo menor coeficiente de aumento do tempo médio ao aumentar a quantidade de regiões. Além disso, os coeficientes de variação do tempo de processamento para os modelos baseados na *quadtree* e na árvore dos componentes foram maiores que os coeficientes dos outros modelos, sendo os coeficientes da *quadtree* um pouco menores que os da árvore dos componentes. Um alto coeficiente de variação revela a alta dependência dos dados da imagem.

### 3.3.4 Aproveitando as características do modelo baseado na *quadtree*

Enquanto o custo computacional apresentado pelo modelo baseado na *quadtree* foi o melhor dentre os modelos testados neste trabalho, a qualidade da segmentação foi, em termos gerais, apenas um pouco melhor que o modelo *pixel grid*. Porém, pode-se explorar uma característica da modelagem usando a *quadtree* para melhorar os resultados das segmentações. Essa característica é a irregularidade do grafo de similaridade, como descrito na Seção 3.2.2. Assim, considere as imagens apresentadas na Figura 3.8.

A Figura 3.8(b) foi gerada a partir das regiões homogêneas da *quadtree* criada pela decomposição da imagem mostrada na Figura 3.8(a). O critério utilizado foi o mesmo descrito na Seção 3.3.1. A Figura 3.8(c), por sua vez, é uma plotagem do ponto central de cada região homogênea da mesma *quadtree*. Observe que onde os pontos estão mais espaçados, as regiões são maiores. Observe também que as regiões menores concentram-se próximos às bordas dos objetos da imagem. Com base nessas observações, é razoável manter regiões adjacentes que são maiores juntas, uma vez que elas representam áreas “distantes” das bordas dos objetos. Esse objetivo pode ser atingido por um ajuste na função de similaridade, dado por

$$w''(u, v) = w(u, v) \text{Deg}(u, v)^2, \quad (3.6)$$

onde  $w(u, v) \in W$  é o peso calculado para a aresta  $e(u, v)$  por uma função de similaridade qualquer; e  $\text{Deg}(u, v)$  é o grau da conexão entre os nós  $u$  e  $v$ , como definido também para a Equação 3.3. Isso aumenta a similaridade relativa entre regiões homogêneas grandes adjacentes, forçando-as a permanecerem juntas.

A segmentação apresentada na Figura 3.8(d) foi gerada pelo modelo baseado na *quadtree*, com função de similaridade  $w(u, v) = 1$ , mas sem o ajuste dado pela Equação 3.3. Observe como as regiões possuem tamanhos diferentes e não representam os objetos da imagem. Já no caso da segmentação apresentada na Figura 3.8(e), que foi gerada pelo mesmo modelo e mesma função de similaridade, porém com o ajuste dado pela Equação 3.3, as regiões estão com o tamanho mais uniforme, apesar de também não representarem os objetos da imagem. Pode-se interpretar que a influência da irregularidade do grafo nos resultados da segmentação foi bastante reduzida. Agora observe a segmentação apresentada na Figura 3.8(f). Ela também foi gerada pelo mesmo modelo e mesma função de similaridade, mas utilizando o ajuste dado pela Equação 3.6. Observe que desta vez as regiões da segmentação representam bem alguns objetos da imagem. Pode-se dizer que, nesse caso, houve uma influência positiva da irregularidade do grafo na qualidade da segmentação. A Tabela 3.3 mostra os resultados obtidos pela aplicação desta nova abordagem do modelo baseado na *quadtree* na segmentação das imagens do *benchmark* BSDS500.



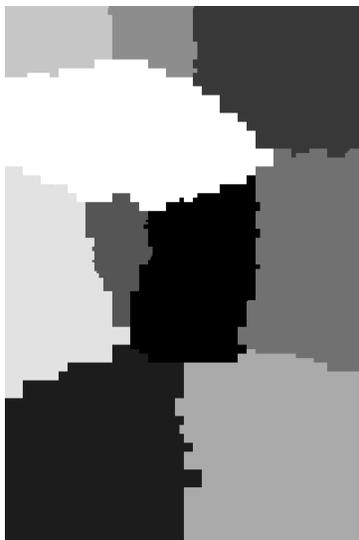
(a) Imagem original.



(b) Reconstrução das regiões da *quadtree*.



(c) Plotagem dos centróides de cada região da *quadtree*.



(d) Sem ajuste dos pesos, influência negativa.



(e) Pesos ajustados pela Equação 3.3, influência neutra.



(f) Pesos ajustados pela Equação 3.6, influência positiva.

Figura 3.8: Segmentação utilizando o modelo *quadtree*. Ilustração da influência na segmentação da irregularidade do grafo e dos esquemas de ajuste dos pesos das arestas propostos.

Utilizando o ajuste dos pesos do grafo de similaridade dado pela Equação 3.6, a qualidade da segmentação pelo modelo baseado na *quadtree* melhorou consideravelmente. Agora, as pontuações dadas pela medida- $f$  e PRI estão muito próximas das pontuações do modelo *pixel grid* em multi-escala, de acordo com o que apresenta a Tabela 3.1. Em relação à métrica de cobertura de região, as pontuações obtidas pelo modelo *quadtree* com o ajuste dos pesos dados pela Equação 3.6 foram significativamente melhores que as pontuações dadas pelo modelo *pixel grid* em multiescala.

Tabela 3.3: Pontuação do modelo baseado na *quadtree* utilizando o ajuste dado pela Equação 3.6, com a similaridade dada pela Equação 3.4.

	Conjunto de validação			Conjunto de teste		
	Borda, medida-f	Cobertura de região	PRI	Borda, medida-f	Cobertura de região	PRI
10 regiões	0.4962	0.4425	0.7392	0.4871	0.4473	0.7528
30 regiões	0.5768	0.3733	0.7362	0.5719	0.3887	0.7587

### 3.4 Conclusão

Neste capítulo foram apresentados os modelos melhorados para se gerar um grafo de similaridade com base na *quadtree* e na árvore dos componentes. Os resultados produzidos pela segmentação por corte normalizado utilizando estes modelos foram comparados com os resultados obtidos por outros dois modelos: o modelo *pixel grid* e o modelo *pixel grid* em multiescala. As grandes variações nos resultados das segmentações e também nos tempos de processamento evidenciam que o modelo utilizado para gerar o grafo de similaridade influencia significativamente o processo de segmentação por corte normalizado.

Notavelmente, os resultados de qualidade das segmentações obtidas pela aplicação dos modelos *pixel grid*, *pixel grid* em multiescala e *quadtree* apresentaram um padrão bem definido. Considerando a combinação das três métricas utilizadas para aferir as qualidades das segmentações e o tempo médio de processamento, o modelo fundamentado na *quadtree* obteve um desempenho ligeiramente melhor que os modelos *pixel grid* e *pixel grid* em multiescala, no contexto dos testes realizados. No caso do modelo baseado na árvore dos componentes, os resultados obtidos fugiram ao padrão apresentado pelos outros três modelos, mostrando claramente que o modelo da árvore dos componentes é o mais singular dentre os modelos testados. Desta forma, é difícil comparar seus resultados de qualidade da segmentação com os resultados dos outros modelos, sendo que as características peculiares do modelo da árvore dos componentes podem torná-lo muito interessante no contexto de alguma aplicação específica.

Os tempos de processamento observados aumentam conforme aumenta a quantidade de regiões requeridas. Porém, a taxa de aumento é diferente para cada modelo, sendo que o modelo *pixel grid* em multiescala e o modelo da árvore dos componentes tiveram respectivamente a maior e a menor taxas de aumento. Assim, a determinação do modelo mais eficiente depende do requerimento da aplicação em quantidade de regiões, utilizando o algoritmo de corte normalizado baseado na teoria espectral proposto por Shi e Malik (2000). Sugere-se que o fator determinante da taxa de aumento no tempo de processamento seja a ordem do grafo de similaridade gerado.

Os quatro modelos comparados neste capítulo foram desenvolvidos para ser utilizados com a técnica de segmentação por corte normalizado. Porém, é razoável esperar que estes modelos também sejam adequados a outros métodos de corte em grafos com função de custo como, por exemplo, o *corte mínimo* (WU E LEAHY, 1993) e o *corte médio* (WANG E SISKIND, 2001). Contudo, maiores investigações são necessárias para se verificar esta hipótese, e também para verificar se estes modelos são aplicáveis a outros métodos de segmentação que utilizam grafos como, por exemplo, os baseados em árvores geradoras (PENG *et al.*, 2013), em análise de ligação simples (GOWER E ROSS, 1969) e em caminhos (PENG *et al.*, 2013). De qualquer forma, este trabalho ressalta a importância em se continuar a pesquisa sobre os modelos de grafos, dado o seu grande impacto no processo de segmentação.

## Capítulo 4

# Um novo algoritmo para aproximar o corte normalizado ótimo

Particionar um grafo de forma a satisfazer o critério ótimo de corte normalizado é um problema de complexidade NP-Completo (SHI E MALIK, 2000). Desta forma, Shi e Malik (2000) utilizaram a teoria espectral de grafos para desenvolver um algoritmo que realiza a aproximação do valor ótimo de corte normalizado. Embora não seja a solução ideal, o algoritmo tornou viável a aplicação do critério de corte normalizado para realizar o particionamento de grafos de alta ordem. Porém, como também observado por Dhillon *et al.* (2004), não é necessário recorrer à teoria espectral de grafos para obter um particionamento do grafo com valor de corte normalizado próximo do mínimo.

Neste Capítulo é apresentado um novo algoritmo para realizar o particionamento de grafos com base no critério do corte normalizado. O novo algoritmo não utiliza os conceitos da teoria espectral de grafos. Ele fundamenta-se em realizar o agrupamento hierárquico dos nós do grafo dois a dois de forma iterativa. Uma heurística é utilizada para determinar quais dois nós são unidos em cada iteração. O desempenho do novo algoritmo é comparado com o desempenho do algoritmo fundamentado na teoria espectral de grafos, proposto por Shi e Malik (2000), utilizando 500 grafos gerados a partir das imagens do *benchmark* BSDS500 da Berkeley. Foram realizados testes para particionar os grafos em 2, 10 e 20 subgrafos. Além de realizar a comparação dos valores de corte normalizado, são comparados também os tempos de processamento de cada algoritmo para cada número de subgrafos. Os resultados revelam características interessantes sobre os dois algoritmos, sendo o novo algoritmo distinto do algoritmo baseado na teoria espectral em muitos aspectos.

## 4.1 Definições preliminares

Dado o particionamento de um grafo  $G = (V_G, E_G, W_G)$  nos subgrafos  $A = (V_A, E_A, W_A)$  e  $B = (V_B, E_B, W_B)$ , o valor do corte normalizado é dado pela Equação 2.5 e, sendo  $V_G = V_A \cup V_B$ , tem-se que

$$\text{Assoc}(A, G) = \sum_{u \in V_A, v \in V_G} W_G(u, v) = \sum_{u \in V_A, v \in V_B} W_G(u, v) + \sum_{i \in V_A, j \in V_A} W_G(i, j) = \text{Cut}(A, B) + Z(A), \quad (4.1)$$

onde  $Z(A) = \sum_{u, v \in V_A} W_A(u, v)$ . Ou seja, a associação de um subgrafo  $A$  com o grafo original  $G$  é o valor do corte somado ao valor de todas as arestas do subgrafo  $A$ . Desta forma, pode-se reescrever a Equação 2.5 como

$$\text{NCut}(A, B) = \frac{\text{Cut}(A, B)}{\text{Cut}(A, B) + Z(A)} + \frac{\text{Cut}(A, B)}{\text{Cut}(A, B) + Z(B)}. \quad (4.2)$$

Para cada iteração  $n$  do algoritmo há um grafo  $G^n = (V_G^n, E_G^n, W_G^n, Z_G^n)$  associado, onde cada elemento do conjunto  $Z_G^n$  é um atributo associado a um nó do grafo. Um nó de um grafo  $G^n$  pode representar um subgrafo  $A$  de um outro grafo  $G^m$ , onde  $m < n$ . Dado que um nó  $u \in V_G^n$  representa um subgrafo  $A \subseteq G^m$ , então  $Z_G^n(u) = Z(A)$ . Assim, pode-se definir o valor do corte normalizado para o subgrafo  $A$  como o valor do corte normalizado para o nó  $u$ , dado por

$$\text{NCut}(A, B) = \text{NCut}(u, B) = \frac{\text{Cut}(u, B)}{\text{Cut}(u, B) + Z_G^n(u)} + \frac{\text{Cut}(u, B)}{\text{Cut}(u, B) + Z(B)}. \quad (4.3)$$

Considerando

$$S(u) = \text{Cut}(u, B) = \sum_{u \sim v / v \in V_G^n} W_G^n(u, v), \quad (4.4)$$

então

$$\text{NCut}(u, B) = \frac{S(u)}{S(u) + Z_G^n(u)} + \frac{S(u)}{Z(G^n) - Z_G^n(u)}, \quad (4.5)$$

uma vez que  $Z(G^n) = \text{Cut}(u, B) + Z_G^n(u) + Z(B) \Rightarrow Z(G^n) - Z_G^n(u) = \text{Cut}(u, B) + Z(B)$ , onde  $B$  é um subgrafo de  $G^n$ .

## 4.2 O algoritmo

O algoritmo aqui proposto fundamenta-se na união iterativa dos nós do grafo dois a dois, respeitando uma heurística. Os grafos utilizados pelo algoritmo, além de serem ponderados, possuem também um atributo associado aos nós. A cada iteração, um novo grafo é gerado, cuja ordem é reduzida em uma unidade em relação ao grafo da iteração anterior. Os dois nós escolhidos para serem unidos são agora representados por um único nó no novo grafo. A Figura 4.1 ilustra a união de dois nós  $u$  e  $v \in G^n$  em um nó  $y \in G^{n+1}$ , onde  $n \in \mathbb{N}^+$  é a iteração.

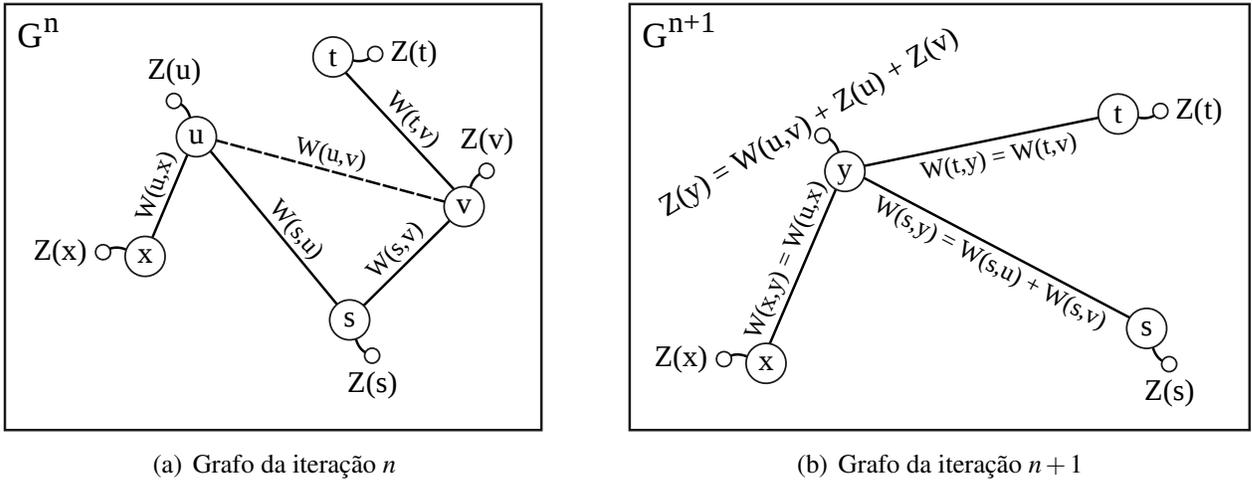


Figura 4.1: Agrupamento dos nós  $u$  e  $v$  ao passar da iteração  $n$  para a iteração  $n + 1$ .

A união dos nós deve ser consistente, *i. e.*, o valor do somatório dos pesos mais o somatório dos atributos dos grafos  $G^n$ ,  $\forall n$ , deve permanecer o mesmo. Note que o atributo do novo nó possui os atributos dos nós antigos somados ao valor da aresta que os conectava, *i. e.*,  $Z(y) = Z(u) + Z(v) + W(u, v)$ , onde  $u$  e  $v \in V^n$ ; e  $y \in V^{n+1}$ . Observe que as arestas também devem ser ajustadas, como ilustrado na Figura 4.1. Note que se um nó  $s$  é tal que  $s \sim u$  e  $s \sim v$ , então o peso da nova aresta  $E(s, y)$  deve ser o somatório dos pesos das arestas  $E(s, u)$  e  $E(s, v)$ , *i. e.*,  $W(s, y) = W(s, u) + W(s, v)$ . Uma vez que não há alteração nas forças e nos valores dos atributos dos nós não envolvidos na união, o valor de corte normalizado deles continua o mesmo.

### 4.2.1 Heurística

Para compreender a heurística que determina quais nós  $u$  e  $v$  são unidos em cada iteração, é importante entender o conceito de valor do corte normalizado de um nó, como definido na Equação 4.5. Desta forma, seja o grafo inicial  $G^0 = (V^0, E^0, W^0, Z^0)$ , onde  $Z^0(u) = 0, \forall u \in V^0$ , os valores de corte normalizado são dados por

$$\text{NCut}(u) = \frac{S(u)}{S(u)} + \frac{S(u)}{Z(G^0)} = 1 + \frac{S(u)}{Z(G^0)}. \quad (4.6)$$

Desta forma, na inicialização do algoritmo são calculados os valores de corte normalizado para todos os nós do grafo. Observe que nesse estágio os valores de corte normalizado dos nós dependem exclusivamente de suas forças. Assim, quanto maior a força do nó, maior será o seu valor de corte. Note que, se o valor de corte normalizado do nó é grande, ele pode estar conectado a nós muito similares e, potencialmente, pertence ao interior de um possível grupo. Outra situação é quando o nó possui um alto grau de conexões, o que pode levar a um valor de corte normalizado também grande. Em ambos os casos, é grande a probabilidade de haver ao menos um vizinho do nó que, se unidos, o nó formado terá um valor de corte normalizado menor.

**Regra nº 1:** A primeira regra da heurística define um dos nós que serão unidos. Assim sendo, o nó  $u$  deve ser tal que possua o maior valor de corte normalizado dentre todos os nós da iteração corrente.

**Regra nº 2:** Esta regra determina qual o segundo nó envolvido na operação de união. Assim, o nó  $v$  deve ser tal que, se unido ao nó  $u$ , o valor de corte normalizado do novo nó seja o menor possível dentre os valores que os vizinhos do nó  $u$  podem oferecer.

**Regra nº 3:** Esta regra impõe uma restrição para impedir que os valores de corte normalizado dos agrupamentos não sejam degradados pela operação de união, o que pode ocorrer se o agrupamento resultante da fusão possuir valor de corte normalizado maior que os valores dos dois nós envolvidos na operação. Desta forma, a união só pode ocorrer se o valor de corte normalizado do nó gerado for menor que os valores de corte normalizado de ambos os nós  $u$  e  $v$ . Caso nenhum dos vizinhos do nó  $u$  possa satisfazer esta condição, ele deve ser descartado na iteração corrente, sendo trocado pelo próximo nó com maior valor de corte normalizado.

O Algoritmo 4.1 apresentado a seguir realiza o agrupamento dos nós do grafo de similaridade utilizando a heurística aqui definida. O critério de parada do algoritmo é o número de subgrafos especificado especificado por meio da variável  $\phi$ . Contudo, o algoritmo pode parar antes de atingir o número de subgrafos desejado caso a heurística não mais seja satisfeita para todos os nós do grafo da iteração corrente. Neste caso, o número de subgrafos será sempre maior que o valor  $\phi$  especificado.

**Entrada:** Grafo de similaridade  $G = (V, E, W)$ ;

**Saída:** Agrupamentos de nós  $C$ ;

$C(u) \leftarrow \{u\}, \forall u \in V$ ;

$i \leftarrow 0$ ;

$Z^i(u) \leftarrow 0, \forall u \in V$ ;

$G^i \leftarrow (V, E, W, Z^i)$ ;

$N \leftarrow V^i$ ;

**enquanto**  $\text{Ordem}(G^i) > \phi$  e  $N \neq \emptyset$  **faça**

$o \leftarrow u \in N$ , tal que  $\text{NCut}(o)$  seja máximo;

$N \leftarrow N \setminus \{o\}$ ;

$M \leftarrow v \in V^i / o \sim v$ ;

$d \leftarrow \text{NULO}$ ;

$d \leftarrow v \in M$ , tal que  $\text{NCut}(\{o, d\}, B)$  seja o menor valor que satisfaça

$\text{NCut}(\{o, d\}, B) < \text{NCut}(o)$  e  $\text{NCut}(\{o, d\}, B) < \text{NCut}(d)$ , onde  $B = V^i \setminus \{o, d\}$ ;

**se**  $d \neq \text{NULO}$  **então**

        Realiza a união dos nós  $o$  e  $d \in V^i$ , criando um novo nó  $p \in V^{i+1}$ . Note que

$V^{i+1} = \{V^i \setminus \{o, d\}\} \cup \{p\}$ ;

        Realiza a atualização dos atributos, arestas e pesos, como ilustrado na Figura 4.1;

$C(p) \leftarrow C(o) \cup C(d)$ ;

$C(o) \leftarrow C(d) \leftarrow \emptyset$ ;

$i \leftarrow i + 1$ ;

$N \leftarrow V^i$ ;

**Algoritmo 4.1:** Algoritmo para aproximação do valor ótimo de corte normalizado em um grafo de similaridade pelo agrupamento de nós.

### 4.3 Teste de desempenho do algoritmo

O teste apresentado nesta Seção tem o propósito de comparar o desempenho do novo algoritmo em relação ao algoritmo fundamentado na teoria espectral de grafos, que foi proposto por Shi e Malik (2000). A base do teste são 500 grafos de similaridade gerados a partir de todas as imagens do *benchmark* BSDS500. O modelo *pixel grid* foi utilizado para gerar os grafos, com raio de conexão  $r = \sqrt{2}$ , que corresponde à vizinhança-8 de cada *pixel* na imagem (PEDRINI E SCHWARTZ, 2008). A Figura 4.2 mostra os resultados dos particionamentos dos grafos da base de teste em dois subgrafos, pelo novo algoritmo e pelo algoritmo fundamentado na teoria espectral de grafos.

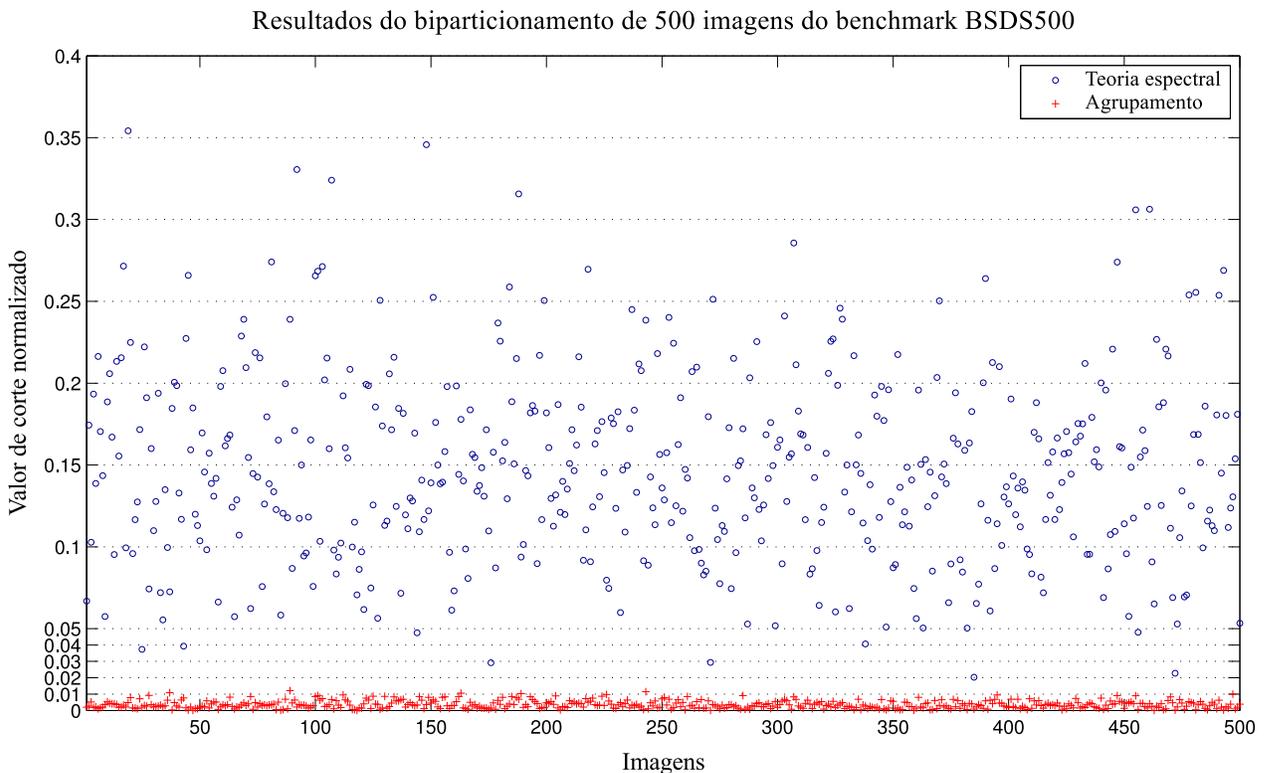


Figura 4.2: Resultados do particionamento de 500 grafos gerados a partir das imagens do *benchmark* BSDS500, em 2 subgrafos.

O gráfico apresentado na Figura 4.2 revela que o novo algoritmo resultou em valores de corte normalizado expressivamente menores que os valores obtidos pelo algoritmo de Shi e Malik (2000). Enquanto que a média dos valores de corte normalizados dados pelo algoritmo fundamentado na teoria espectral foi  $0.1469 \pm 0.0567$ , a média do novo algoritmo aqui proposto foi  $0.0036 \pm 0.0024$ , cerca de 40 vezes menor. Em absolutamente nenhum caso o algoritmo fundamentado na teoria espectral apresentou um valor de corte normalizado menor que o apresentado pelo algoritmo proposto neste trabalho.

Para verificar o desempenho do algoritmo em situações nas quais é necessário particionar o grafo em mais que dois subgrafos, foram realizados também experimentos particionando o grafo de similaridade em 10 e em 20 subgrafos. Os resultados são apresentados nos gráficos da Figura 4.3. O valor de corte normalizado para cada subgrafo  $A$  foi dado por  $\text{NCut}(A, G \setminus A)$ , onde  $G$  é o grafo original. Para cada particionamento, os subgrafos foram organizado por ordem crescente de valor de corte normalizado.

Valores de corte normalizado para particionamentos em 10 e em 20 subgrafos

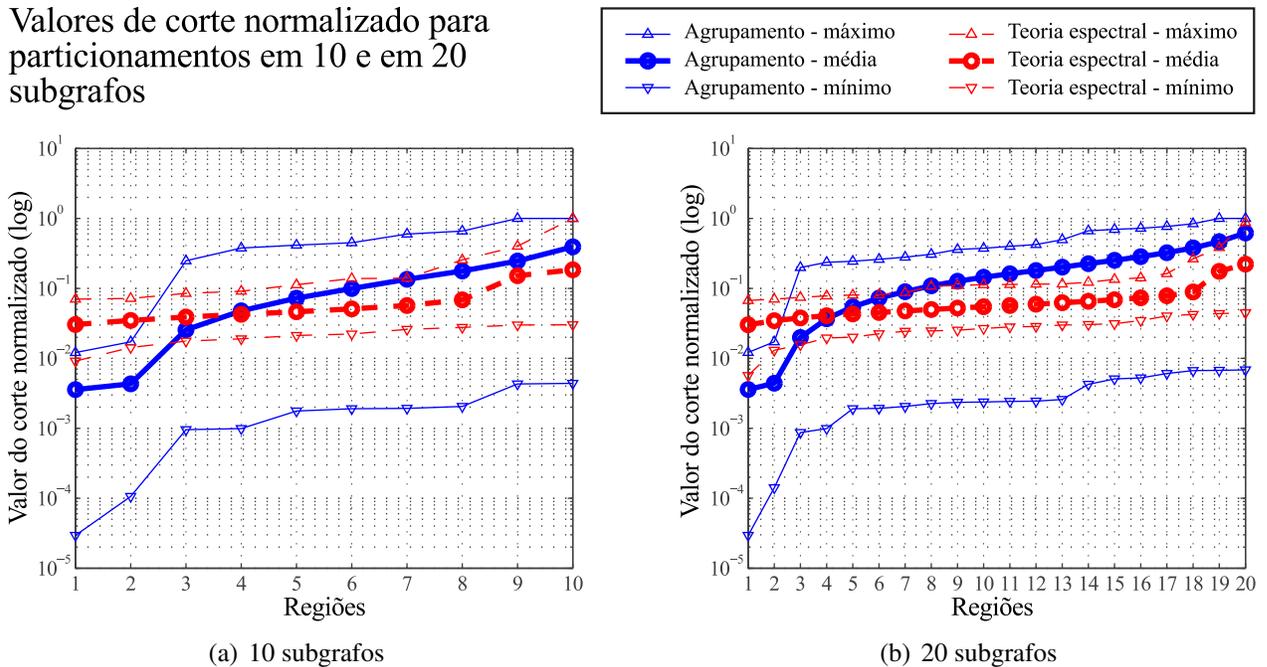


Figura 4.3: Resultados do particionamento dos 500 grafos gerados a partir das imagens do *benchmark* BSDS500, em 10 e em 20 subgrafos.

Nos gráficos apresentados na Figura 4.3 é possível notar que os valores de corte normalizado para cada subgrafo dado pelo algoritmo fundamentado na teoria espectral variam menos que os valores dados pelo novo algoritmo. Se comparados os três subgrafos com menor valor de corte normalizado, o novo algoritmo obteve um valor médio menor que o algoritmo da teoria espectral. Porém, para o subgrafo com o quarto menor valor de corte normalizado pode-se considerar um empate, e a partir do subgrafo com o quinto menor valor de corte normalizado, os valores obtidos pelo novo algoritmo são maiores que o algoritmo da teoria espectral. Um fato interessante é que os dois menores valores de corte normalizado obtidos pelo novo algoritmo são muito menores que os outros, tanto para segmentações em 10 subgrafos quanto segmentações em 20 subgrafos. No caso do algoritmo fundamentado na teoria espectral ocorre uma relação semelhante, porém inversa, sendo os dois maiores valores de corte normalizado consideravelmente maiores que os demais.

A Figura 4.4 apresenta um gráfico com os tempos de processamento para cada algoritmo, considerando particionamentos dos grafos em 2, 10 e 20 subgrafos. Os testes foram executados em uma máquina com processador de 2.2 GHz, memória de 3 GB e sistema operacional Linux. Pode-se observar uma grande variação no tempo de processamento gasto pelo algoritmo baseado na teoria espectral, enquanto que o novo algoritmo mostra-se estável, com muito pouca variação. Isso significa que o novo algoritmo, diferente do algoritmo fundamentado na teoria espectral, é muito pouco dependente dos dados da imagem.

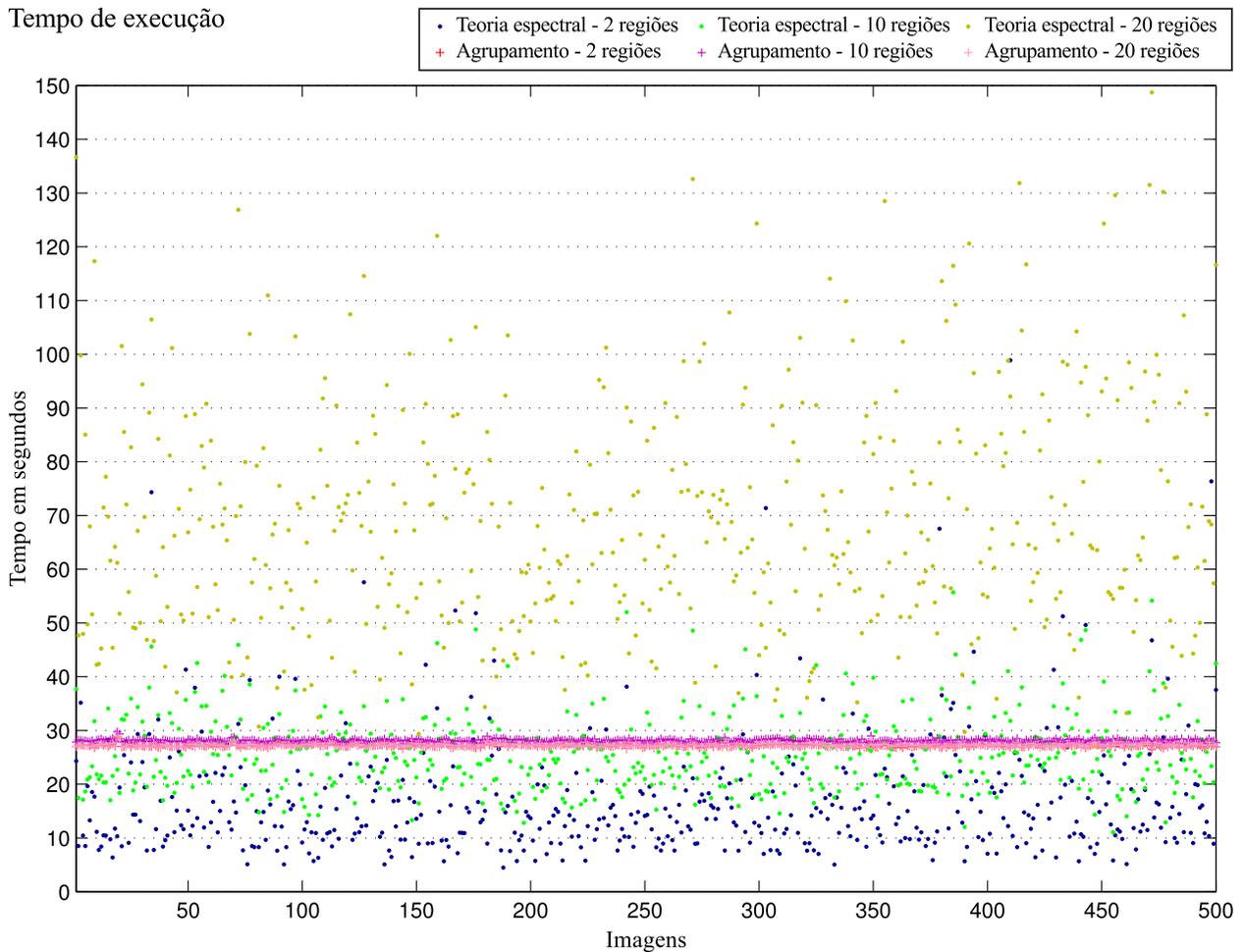


Figura 4.4: Tempo de processamento para particionar os 500 grafos gerados a partir das imagens do *benchmark* BSDS500.

Considerando o particionamento dos grafos em dois subgrafos, o algoritmo fundamentado na teoria espectral apresentou um tempo de processamento médio bem inferior ao tempo gasto pelo novo algoritmo. Porém, em alguns casos o novo algoritmo ainda apresenta menor tempo de processamento. Quando o particionamento é realizado em 10 subgrafos o tempo médio do dois algoritmos fica bem próximo, sendo que, para muitos casos, o tempo do novo algoritmo foi menor. Agora, para particionamentos em 20 regiões, a média de tempo gasto pelo novo algoritmo foi

muito menor que a média de tempo gasto pelo algoritmo baseado na teoria espectral. De fato, o algoritmo fundamentado na teoria espectral apresenta um crescimento no tempo de processamento não linear, à medida que aumenta o número de subgrafos. Para o novo algoritmo, não houve diferença expressiva nos tempos de processamento para particionamento em 2, 10 e 20 regiões. Em suma, dada sua forma de trabalho, ao aumentar o número de subgrafos, o tempo de processamento deve diminuir, ao invés de aumentar.

## 4.4 Conclusão

Neste Capítulo foi introduzido um novo algoritmo para realizar o particionamento de grafos com base no critério de corte normalizado. O algoritmo fundamenta-se em um processo iterativo de agrupamento de nós, dois a dois. Uma heurística é utilizada para determinar quais dois nós são unidos em cada iteração.

Os resultados de um teste preliminar utilizando grafos de similaridade gerados a partir das imagens do *benchmark* BSDS500 foram muito promissores. O novo algoritmo apresenta características bastante distintas das características do algoritmo fundamentado na teoria espectral de grafos, proposto por Shi e Malik (2000). Contudo, muita pesquisa ainda deve ser feita para explorar totalmente as características desse novo algoritmo.

Embora os resultados obtidos pelo novo algoritmo de corte normalizado nos testes apresentados neste capítulo tenham sido melhores que os resultados obtidos pelo algoritmo da teoria espectral, esta análise não pode ser conclusiva, uma vez que muitos aspectos do novo algoritmo ainda são desconhecidos. Por exemplo, não se sabe como é o comportamento do novo algoritmo em grafos mais densos. Também, uma análise do valor de corte normalizado para cada subgrafo, como a apresentada na Figura 4.3, precisa ser feita para maiores quantidades de subgrafos, da ordem de centenas.

## Capítulo 5

# Utilizando o corte normalizado para segmentar colônias de bactérias em placas de Petri

Uma aplicação de PDI com crescente interesse há muitas décadas é a contagem automática de colônias de bactérias (MANSBERG, 1957; CLARKE *et al.*, 2010). A medição da taxa de sobrevivência das UFC's em uma cultura microbiológica é amplamente utilizada para a realização de análises biomédicas, ambientais, de saúde pública e etc. A contagem manual das UFC's é um processo de trabalho intenso e que consome muito tempo, uma vez que podem existir de centenas a milhares de colônias de bactérias em uma única placa de Petri. Desta forma, a contagem manual exige a alocação do tempo de especialistas, limitando a quantidade de análises que um laboratório é capaz de realizar (CHEN E ZHANG, 2009).

O problema de segmentação de colônias de bactérias possui algumas características bem distintas daquelas encontradas, por exemplo, em problemas de segmentação de imagens naturais. A principal característica é o alto número de colônias que podem ser encontradas em uma placa. Por isso, esta aplicação foi escolhida para testar as técnicas de segmentação por corte normalizado. Porém, o objetivo não é o de comparar a segmentação por corte normalizado com outras técnicas utilizadas por sistemas existentes, como os propostos por Chen e Zhang (2009) e por Clarke *et al.* (2010). Nem tão pouco é objetivo comparar o desempenho de diferentes modelos de grafo de similaridade para a aplicação específica de segmentação de colônias de bactérias, de forma que seria similar ao estudo realizado no Capítulo 3. O objetivo neste Capítulo é o de simplesmente verificar a aplicabilidade do critério de corte normalizado para segmentar as colônias de bactérias e, assim, ter uma visão por um outro ângulo sobre as características dos algoritmos de corte normalizado.

## 5.1 O corte normalizado aplicado à segmentação de colônias de bactérias

A segmentação das colônias de bactérias pode ser realizada de duas formas:

1. como um problema de segmentação dos *pixels* em duas regiões, *i. e.*, determinar se o *pixel* pertence a uma colônia ou ao fundo. A limiarização global é um exemplo de técnica que utiliza essa abordagem. Depois da segmentação, os componentes conexos da classe objeto (nesse caso, as colônias) são rotulados. Em geral, é também necessário verificar se um componente conexo pertence a mais de uma colônia, sendo necessário realizar uma nova segmentação para separar as colônias agrupadas;
2. como um problema de segmentação multiregião, onde cada colônia é segmentada em uma região distinta. Apesar de ser uma abordagem de maior complexidade, ela é mais interessante para técnicas que envolvem a utilização de grafos. Isso porque a abordagem de segmentação binária exigiria um grafo completo ou, no mínimo, de alta densidade.

Uma característica da aplicação de contagem de colônias de bactérias em placas é a alta resolução das imagens, necessária para possibilitar a identificação de colônias muito pequenas. Essa característica torna o problema da segmentação binária utilizando grafos ainda mais crítica, uma vez que o grafo gerado terá também alta ordem. Observe a imagem de uma placa de Petri e seu detalhe, apresentados na Figura 5.1. Note como as microcolônias mostradas no detalhe não podem ser visualizadas adequadamente na imagem de menor resolução.

A imagem apresentada na Figura 5.1 revela uma outra característica do problema de segmentação de colônias de bactérias: o número de colônias em uma placa pode ser muito grande, sendo que as placas viáveis podem conter centenas de colônias (TORTORA *et al.*, 2005). Assim, considerando uma simples extrapolação dos resultados obtidos na Seção 4.3, uma abordagem de segmentação multiregião de colônias de bactérias utilizando o algoritmo fundamentado na teoria espectral é inviável. Contudo, a utilização do critério do corte normalizado para segmentar colônias de bactérias pode ser possível utilizando-se o novo algoritmo introduzido no Capítulo 4.

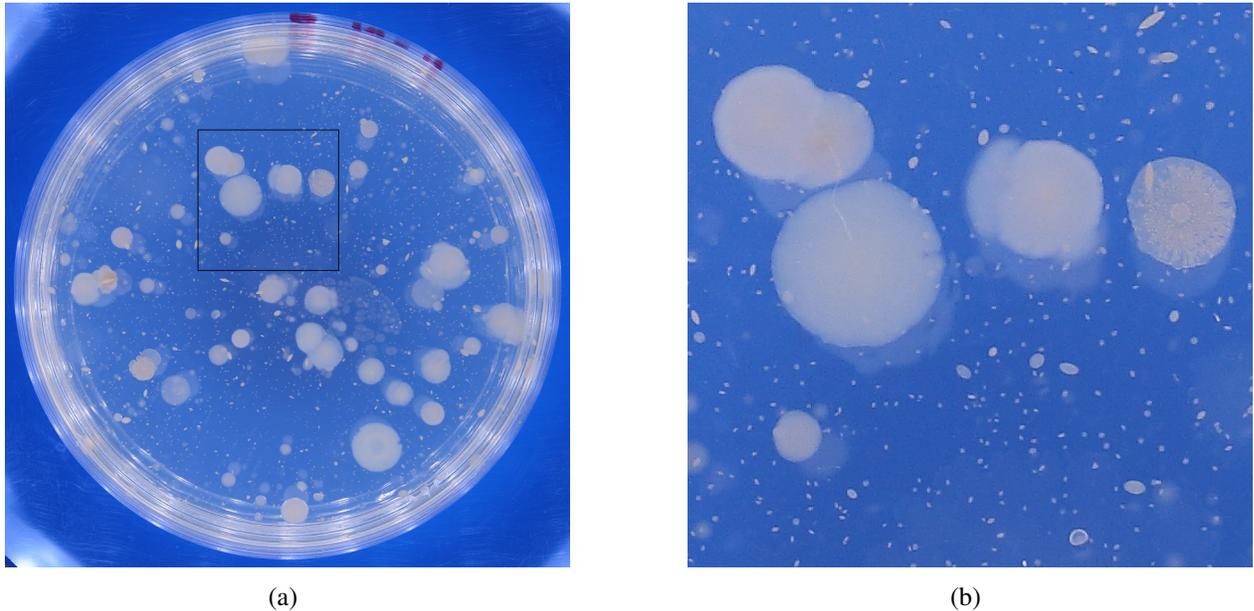


Figura 5.1: Uma imagem em alta definição de uma placa de Petri cultivada com bactérias heterotróficas. A Figura 5.1(b) mostra em alta definição a região destacada na imagem de baixa resolução apresentada na Figura 5.1(a).

### 5.1.1 Experimentos

A preparação dos testes levou em consideração que a segmentação deve ser passível de ser executada em microcomputadores modernos, com CPU<sup>1</sup> de aproximadamente 2 GHz e 4 GB de memória. A maioria dos laboratórios possuem ou podem adquirir ao menos um equipamento com essas configurações. Assim, o sistema utilizado para a execução dos testes foi um microcomputador com CPU de 2.2 GHz e 3 GB de memória.

Durante a preparação dos testes verificou-se que nenhuma das implementações dos algoritmos foi capaz de segmentar as imagens completas das placas com colônias de bactérias, que possuem tamanho de  $2200 \times 2200$  pixels. Mesmo com um grafo muito esparso, com grau médio  $\bar{d}(u) = 8$ , onde  $u \in V$ , a memória do sistema foi insuficiente. Por isso, foram utilizados fragmentos das imagens das placas, com tamanho  $440 \times 440$  pixels, apresentados nas Figuras 5.2(a), 5.2(c), 5.3(a) e 5.3(c). As placas foram cultivadas com diferentes diluições de água de esgoto bruto e tratado em meio de cultivo não seletivo. Desta forma, nas placas cresceram bactérias heterotróficas da água, cujas colônias podem apresentar morfologias distintas (TORTORA *et al.*, 2005).

---

<sup>1</sup>Unidade Central de Processamento.

O modelo utilizado para gerar os grafos de similaridade foi o *pixel grid*, com raio de conexão  $r = \sqrt{2}$ . Os pesos das arestas dos grafos foram dados pela função de similaridade

$$w(u, v) = \exp^{-(|F_r(u)-F_r(v)|+|F_g(u)-F_g(v)|+|F_b(u)-F_b(v)|)}, \quad (5.1)$$

onde  $F_r, F_g$  e  $F_b \subset F$  são, respectivamente, os canais vermelho, verde e azul da imagem  $F$ .

## Resultados

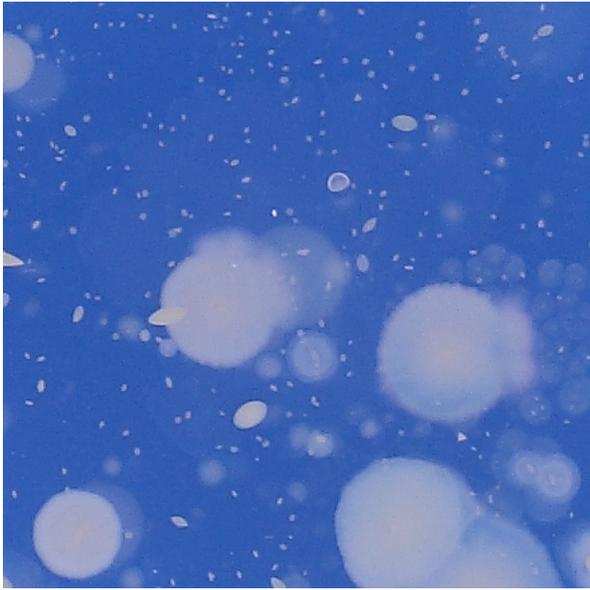
As Figuras 5.2(b), 5.2(d), 5.3(b) e 5.3(d) apresentam os resultados da segmentação realizada pelo novo algoritmo, respectivamente, em 1.259, 707, 93 e 180 regiões. Observe que o número de regiões foi dado automaticamente pelo novo algoritmo. Isso ocorre quando no grafo da iteração corrente, nenhum dos nós possui um vizinho que satisfaz a regra nº 3 da heurística descrita no Capítulo 4.

Em uma análise qualitativa das segmentações, pode-se afirmar que o número de regiões determinado automaticamente foi adequado. Contudo, é possível observar que algumas colônias possuem não apenas uma região representativa, mas sim um conjunto de regiões. Isso implica que o número de regiões obtidas não corresponde ao número de colônias, sendo ainda necessário um pós-processamento para classificar as regiões obtidas em colônias ou não colônias antes de realizar a tarefa de enumeração e contagem.

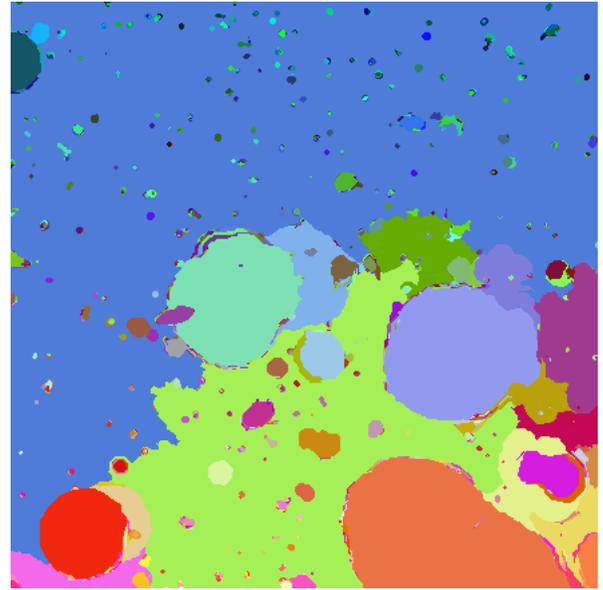
Uma vez que o algoritmo de segmentação de imagens fundamentado na teoria espectral exige a especificação do número de regiões, e para promover uma comparação justa, as quantidades de regiões determinadas pelo novo algoritmo foram utilizadas para segmentar as respectivas imagens pelo algoritmo baseado na teoria espectral. Porém, para todas as imagens utilizadas no teste, o processo de segmentação foi cancelado ao atingir 10 minutos, *i. e.*, em todos os casos a segmentação de apenas um fragmento da imagem levaria mais de 10 minutos pela utilização do algoritmo fundamentado na teoria espectral. A segmentação pelo novo algoritmo levou 79.49 s, 53.58 s, 39.81 s e 43.23 s, respectivamente, para as imagens apresentadas nas Figuras 5.2(a), 5.2(c), 5.3(a) e 5.3(c).

## 5.2 Conclusão

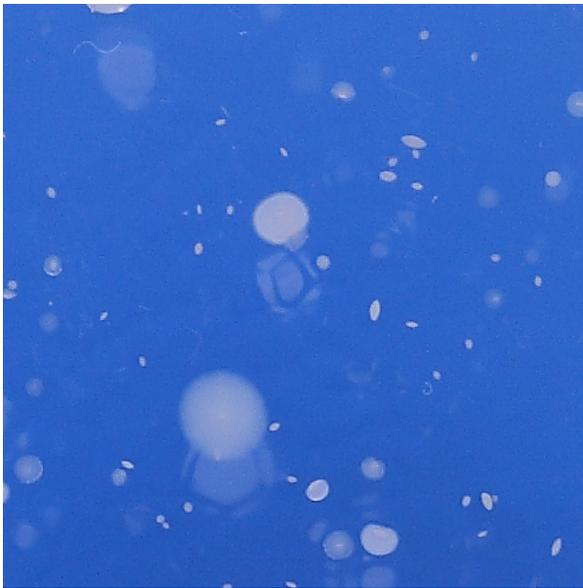
Neste Capítulo foi apresentada uma análise preliminar sobre a utilização do critério de corte normalizado para segmentar imagens de placas de Petri com colônias de bactérias heterotróficas da água. A análise vale-se de um experimento com quatro fragmentos de imagens de placas para revelar as



(a) Fragmento



(b) 1259 regiões



(c) Fragmento



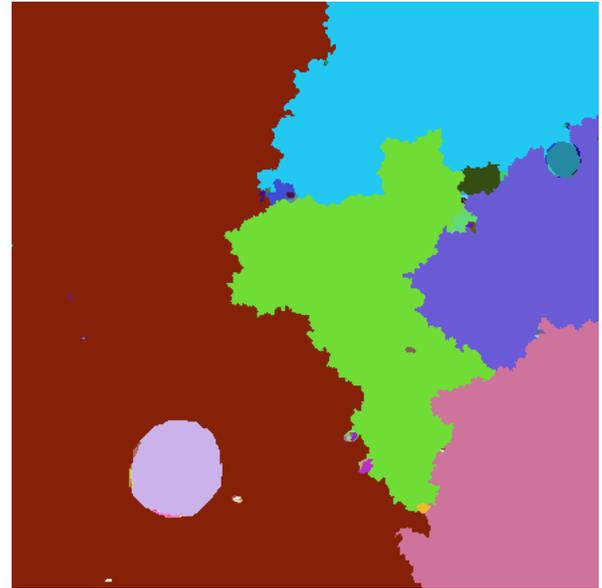
(d) 907 regiões

Figura 5.2: Segmentações resultantes para dois fragmentos de imagens de placas de Petri. O fragmento mostrado na Figura 5.2(a) possui alta densidade de colônias, e sua segmentação em 1.259 regiões é apresentada na Figura 5.2(b). O fragmento apresentado na Figura 5.2(c) possui menor densidade de colônias, e sua segmentação em 707 regiões é apresentada na Figura 5.2(d).

características da aplicação e o comportamento do novo algoritmo desenvolvido neste trabalho e do algoritmo fundamentado na teoria espectral, desenvolvido por Shi e Malik (2000).



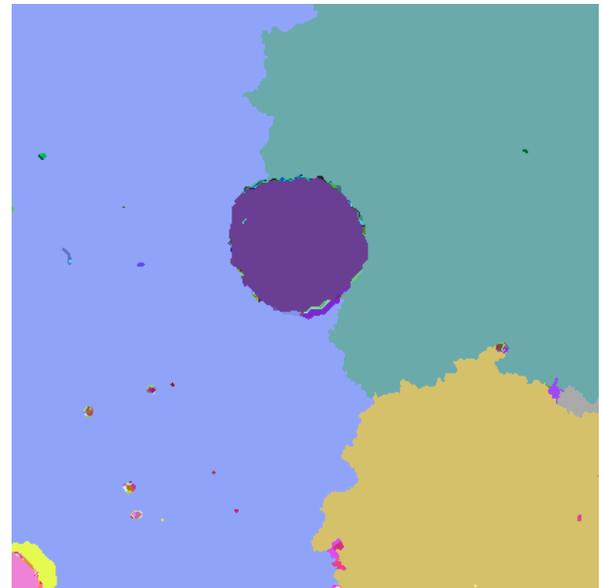
(a) Fragmento



(b) 93 regiões



(c) Fragmento



(d) 180 regiões

Figura 5.3: Segmentações resultantes para dois fragmentos de imagens de placas de Petri com densidade de colônias consideravelmente menor que a densidade dos fragmentos apresentados na Figura 5.2. A segmentação em 93 regiões do fragmento mostrado na Figura 5.3(a) é apresentada na Figura 5.3(b). Para o fragmento apresentado na Figura 5.3(c), a segmentação resultou em 180 regiões, como pode ser observado na Figura 5.3(d).

Mesmo utilizando apenas um fragmento das imagens, o algoritmo baseado na teoria espectral requer mais de 10 minutos para executar a segmentação com a mesma quantidade de regiões obtidas pelo novo algoritmo. Assim, ficou evidente que a utilização do algoritmo baseado na teoria

espectral para aplicações que demandam um elevado número de regiões é inviável. Contudo, a utilização do critério de corte normalizado ainda é possível para tais aplicações se utilizado o novo algoritmo, o qual apresentou tempo de processamento viável, tendo delimitado adequadamente a maioria das colônias presentes nos fragmentos de placas testados. Note que os resultados foram obtidos pela execução de implementações-protótipos, *i. e.*, as implementações podem ser otimizadas para reduzir o custo computacional.

Nos testes realizados pôde ser observada a característica do novo algoritmo de parar automaticamente o processo de agrupamento dos nós do grafo antes de atingir a bipartição. Tão importante quanto identificar quais as regiões, é determinar quantas são as regiões. Por isso, essa característica é muito interessante não apenas para a segmentação de imagens, mas também para toda a área de classificação não-supervisionada (WU, 2012). Todavia, muito trabalho nessa direção ainda precisa ser realizado.

# Capítulo 6

## Considerações finais

Neste capítulo são apresentadas algumas discussões complementares, citando como foi a trajetória do curso de mestrado e relacionando alguns possíveis trabalhos futuros. Uma conclusão geral deste trabalho também é apresentada.

### 6.1 A trajetória do curso de mestrado

Os dois primeiros semestres do mestrado tiveram de ser voltados quase que exclusivamente para o cumprimento dos créditos em disciplinas de pós-graduação previstos pelo programa. O pouco tempo restante foi empregado no estudo de assuntos relacionados ao corte normalizado e à aplicação de segmentação de colônias de bactérias. O primeiro assunto específico estudado foi dado como uma continuação natural dos trabalhos desenvolvidos no projeto de iniciação científica mencionado no Capítulo 1, e também no mestrado de Ferreira (2011). O trabalho de Ferreira (2011) foi o de realizar um estudo quantitativo sobre a qualidade da segmentação produzida pelos modelos propostos na iniciação científica. Contudo, os modelos e as implementações desenvolvidas na época limitavam consideravelmente o tamanho das imagens que poderiam ser processadas. Desta forma, neste primeiro estudo tratamos de promover alterações nas implementações e nos modelos de forma a tornar possível o processamento de imagens maiores, possibilitando segmentar as imagens do *benchmark* da Berkeley (MARTIN *et al.*, 2001) sem a necessidade de qualquer transformação. O resultado deste estudo foi publicado em um capítulo do livro “*New Frontiers on Graph Theory*” (CARVALHO E COSTA, 2012).

Posteriormente, já no segundo ano do mestrado, identificou-se que o motivo da baixa eficácia do modelo baseado na *quadtree* era a influência negativa da irregularidade do grafo gerado, dado que o corte normalizado é uma métrica global. Assim, por meio do ajuste dos pesos das arestas foi possível neutralizar a influência da irregularidade do grafo. Mais ainda, foi possível utilizar o ajuste dos pesos para tornar influência da irregularidade do grafo positiva, melhorando significativamente a eficácia do modelo *quadtree*. Este estudo está descrito no Capítulo 3 e também é tema de um artigo que foi enviado para publicação à revista *Pattern Recognition*.

Voltando ao primeiro ano do curso, o segundo assunto tratado foi a questão da aplicação de contagem de colônias de bactérias. No início o objetivo seria o de verificar se era possível melhorar a eficácia da contagem pela utilização da segmentação por corte normalizado. Desta forma, os primeiros estudos trataram sobre o planejamento de um *dataset* rotulado de imagens de placas, e também sobre alguns sistemas de contagem já existentes. Como já foi mencionado no Capítulo 1, o objetivo deste trabalho foi alterado. A alteração foi motivada por observações da banca de qualificação do mestrado, a qual sugeriu que o foco do trabalho deveria ser a ferramenta de corte normalizado ao invés de a aplicação de segmentação de colônias. De fato, após o exame de qualificação outras observações sobre o problema de contagem de colônias de bactérias reforçaram a motivação para a mudança de foco do trabalho. Maiores detalhes sobre estas observações são apresentados a seguir.

### 6.1.1 Aspectos práticos da contagem de colônias

Talvez a maior dificuldade encontrada em relação ao problema de contagem de colônias de bactérias foi a falta de familiaridade com o assunto. Naturalmente, a busca pelo conhecimento teórico foi o primeiro passo, uma vez que é mais fácil ler um livro do que simplesmente ir ao laboratório sem ter nem idéia do que fazer com os recursos nele disponíveis. Porém, alguns conceitos importantes foram de fato compreendidos somente depois de literalmente por a mão na massa e ver de perto todos os passos e dificuldades do processo de cultivo. Esta interação com o laboratório foi intensa durante os dois últimos semestres do curso.

Talvez o conceito mais importante sobre um processo de contagem de colônias de bactérias é o de que os resultados obtidos se referem a um instante “congelado” no tempo, *análogo a uma fotografia*. O ciclo de reprodução das bactérias é muito curto, da ordem de alguns minutos (TORTORA *et al.*, 2005), de forma que um estudo somente pode retratar uma condição passada do ambiente de onde a amostra foi retirada. Para se ter uma idéia sobre a dimensão deste problema, o simples ato de recolher a amostra pode fazer com que seu novo ambiente temporário de armazenamento seja mais ou menos favorável à reprodução das bactérias do que no ambiente original. Levando em

consideração que na maioria dos casos as amostras demoram horas até serem finalmente encubadas nas placas de Petri, devido ao tempo que toma o transporte do local da coleta até o laboratório, o processo inerentemente já insere um erro imensurável. Além disso, o processo está sujeito à várias fontes de contaminação que podem agravar a situação. Em suma, o procedimento de contagem de colônias é conscientemente utilizado pelos laboratórios apenas como uma estimativa do volume de bactérias no meio original. Dada esta característica do problema, percebeu-se que o aumento da eficácia dos algoritmos de contagem não é um requisito tão relevante quanto se imaginava no princípio do curso. Uma vez que já existem sistemas de contagem de boa eficácia, mesmo que sejam de difícil acesso a muitos laboratórios, o objetivo original da pesquisa tinha pouca validade científica.

### **6.1.2 A mudança de foco do trabalho**

A idéia para o algoritmo de corte normalizado apresentado no Capítulo 4 surgiu já no terceiro semestre do curso. Embora ele tenha características muito interessantes para o problema de segmentação de colônias de bactérias, como foi apresentado neste trabalho, seu advento não foi proveniente de uma tentativa de solucionar os problemas que estávamos enfrentando. Curiosamente, a inspiração para o novo algoritmo de corte normalizado veio do estudo de algoritmos de árvore geradora mínima e de ligação simples (GOWER E ROSS, 1969) que, na ocasião, estavam em pauta em uma disciplina sobre classificadores fundamentados na teoria dos grafos, ministrada na Faculdade de Engenharia Elétrica e de Computação da Unicamp.

Dada a necessidade em se conhecer melhor as propriedades do novo algoritmo – uma demanda ainda pertinente à trabalhos futuros – ficou claro que a aplicação de segmentação de colônias de bactérias poderia ser utilizada como um estudo de caso. Desta forma, a mudança no foco do trabalho estava definida. Complementarmente, grafos gerados a partir das imagens do benchmark da Berkeley (ARBELÁEZ *et al.*, 2011) foram úteis para se ter uma visão geral sobre o comportamento do algoritmo em relação ao comportamento do algoritmo da teoria espectral. Apesar de terem sido utilizados quinhentos grafos de ordem aproximada de 150.000 nós, este cenário de teste não foi suficientemente exaustivo para se conhecer todas as características do novo algoritmo. Por exemplo, todos os grafos tinham a mesma densidade, a qual foi também muito baixa. Assim, neste trabalho praticamente nada se sabe sobre o comportamento do novo algoritmo em grafos de alta densidade.

## 6.2 Trabalhos futuros

Diversos são os temas possíveis para trabalhos futuros que estão relacionados com o que foi estudado e desenvolvido durante o curso. Um deles pode ser buscar atingir o objetivo original do projeto de pesquisa, inserindo o corte normalizado em um processo completo de contagem de colônias de bactérias, visando melhorar a acurácia da contagem. Contudo, um estudo preliminar sobre a viabilidade científica em se buscar tal resultado é imprescindível. Ao invés disso, propõe-se utilizar o corte normalizado como uma ferramenta de classificação não supervisionada para determinar se as regiões, obtidas por técnicas quaisquer de segmentação, são ou não colônias. De fato, aplicar o novo algoritmo de corte normalizado à qualquer problema que envolva classificação é um bom tema para pesquisas futuras.

A aplicação de contagem de colônias de bactérias, ao menos quando se trata de bactérias heterotróficas (TORTORA *et al.*, 2005), as quais possuem morfologia variada, representa um excelente desafio a qualquer classificador. Desta forma, propõe-se também como trabalho futuro a compilação de uma base de imagens de colônias de bactérias heterotróficas anotadas que possa servir para testar sistemas de classificação. Neste trabalho uma base de imagens foi iniciada, mas necessita de mais imagens e anotações. A continuação deste trabalho representa uma contribuição considerável à toda a comunidade de pesquisa em análise de imagens.

Durante o processo de anotação da base de imagens de colônias de bactérias foi utilizado um software desenvolvido especificamente para esta tarefa. O software é muito simples, apenas exibe a imagem e permite ao operador marcar com um ponto a localização aproximada de uma colônia. Um único recurso adicional é a possibilidade de ampliar e reduzir a imagem – *zoom*. Surpreendentemente, os usuário gostaram muito da ferramenta e alegaram que este software seria de grande valia no cotidiano do laboratório mesmo que a contagem tivesse que ser realizada manualmente. Porém, um ponto negativo citado foi a necessidade do uso do mouse, o qual restringe a velocidade com a qual os usuário conseguem marcar as colônias. Assim, uma proposta de trabalho futuro é o desenvolvimento de uma ferramenta de contagem manual que possa ser executada em um dispositivo do tipo *tablet*. Este tipo de dispositivo oferece algumas vantagens para este tipo de aplicação:

1. **portabilidade**. Estes dispositivos são móveis e de fácil manipulação, não exigindo qualquer tipo de infraestrutura específica;
2. **câmera embutida**. Usualmente estes dispositivos são equipados com uma câmera embutida, a qual pode ser utilizada para realizar o registro fotográfico da placa;

3. *tela sensível ao toque*. A interface de entrada de um *tablet* é uma tela sensível ao toque. Muitos dispositivos possuem também uma “caneta” que possibilita maior precisão na interação. Este tipo de interface pode ser a solução ideal para o problema do uso do mouse, possibilitando uma maior velocidade no processo de contagem manual. Além disso, a tela sensível ao toque também facilita o controle do *zoom* e da navegação pela imagem.

Por fim, vários trabalhos futuros são possíveis no contexto do novo algoritmo de corte normalizado, uma vez que muitos dos seus aspectos podem não são conhecidos. Alguns deles são o comportamento com grafos mais densos, o desempenho teórico, a combinação com diferentes modelos para gerar o grafo de similaridade e a qualidade da segmentação proporcionada. De fato, há muito trabalho a ser feito para que se possa conhecer bem todas as características desse novo algoritmo, sendo que muitos pontos como, por exemplo, relacionados ao desempenho e à robustez, ainda são desconhecidos.

## 6.3 Conclusão

O trabalho com os modelos para gerar grafos de similaridade, apresentado no Capítulo 3, representa o amadurecimento de uma pesquisa que começou em 2009. Neste trabalho, os modelos e as implementações passaram por melhorias que tornaram possível a aplicação para as imagens do *benchmark* da Berkeley (ARBELÁEZ *et al.*, 2011) sem a necessidade de escalamento, possibilitando uma comparação mais confiável do que a realizada em trabalhos anteriores. Porém, o mais importante foi que as alterações nos modelos também melhoraram consideravelmente a qualidade das segmentações. Em particular para o modelo baseado na *quadtree*, a melhora expressiva na qualidade da segmentação foi devido ao balanceamento dos pesos do grafo de similaridade visando compensar a irregularidade dos graus dos nós. De fato, talvez o ponto mais importante na melhoria do modelo foi a constatação de que a irregularidade do grafo tem forte influência no particionamento por meio do critério de corte normalizado, sendo que este problema pode também ocorrer com outras técnicas de segmentação que utilizam modelagem por grafos.

Certamente, a maior contribuição deste trabalho foi o desenvolvimento do novo algoritmo de particionamento pelo critério de corte normalizado. Este novo método utiliza conceitos próprios de algoritmos aglomerativos de agrupamento hierárquico (WU, 2012) e, desta forma, herda algumas das suas características como, por exemplo, a possibilidade de rastrear as uniões realizadas em cada iteração por meio de um dendrograma. O experimento com a aplicação específica de segmentação de colônias de bactérias cultivadas em placas de Petri utilizando o critério de corte normalizado revelou diferenças importantes entre o algoritmo desenvolvido por Shi e Malik (2000) e o algoritmo

proposto neste trabalho. Além disso, uma primeira comparação com o algoritmo que utiliza a teoria espectral mostra que o novo algoritmo consegue aproximar 40 vezes mais o valor ótimo do corte normalizado, na média. Notavelmente, o novo algoritmo possibilita a aplicação do critério de corte normalizado em uma classe de problemas de segmentação para os quais o algoritmo fundamentado na teoria espectral de grafos é inadequado. Especula-se que outras técnicas de segmentação baseadas na teoria espectral de grafos também sofrem com a limitação na quantidade de regiões.

# Referências Bibliográficas

- ABREU, N.M.M. Teoria espectral dos grafos: um híbrido entre a álgebra linear e a matemática discreta e combinatória com origens na química quântica. **TEMA Tendências em Matemática Aplicada e Computacional**, v. 6, 1–10, 2005.
- ARBELÁEZ, P.; MAIRE, M.; FOWLKES, C. e MALIK, J. Contour detection and hierarchical image segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 33, n. 5, 898–916, maio 2011.
- CANNY, J. A computational approach to edge–detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, pp. 679–700, 1986.
- CARVALHO, M.A.G. e COSTA, A.L. **New Frontiers in Graph Theory**, capítulo Combining hierarchical structures on graphs and Normalized Cut for image segmentation. Intech, 2012.
- CARVALHO, M.A.G.; COSTA, A.L.; FERREIRA, A.C.B. e CESAR-JR, R.M. Image segmentation using component tree and normalized cut. In **23rd SIBGRAPI – Conference on Graphics, Patterns and Images**. Gramado–RS, Brasil, 2010a.
- CARVALHO, M.A.G.; FERREIRA, A.C.B. e COSTA, A.L. Image segmentation using quadtree–based similarity graph and normalized cut. In **15th CIARP – Iberoamerican Congress on Pattern Recognition**. São Paulo–SP, Brasil, 2010b.
- CARVALHO, M.A.G.; FERREIRA, A.C.B.; PINTO, T.W. e CESAR–JR, R.M. Image segmentation using watershed and normalized cuts. In **Proc. of 22th Conference on Graphics, Patterns and Images (SIBGRAPI)**. Rio de Janeiro - Brazil, 2009.
- CHAPMAN, S.J. **Programação em MATLAB para engenheiros**. Cengage Learning, 2<sup>a</sup> ed., 2010.
- CHEN, W.B. e ZHANG, C. An automated bacterial colony counting and classification system. **Information Systems Frontiers**, v. 11, n. 4, 349–368, 2009.
- CHUNG, F.R.K. Spectral graph theory. In A.M. Society, editor, **CBMS Regional Conference Series in Mathematics**, 92. 1997.
- CLARKE, M.L.; BURTON, R.L.; HILL, A.N.; LITORJA, M.; NAHM, M.H. e HWANG, J. Low-cost, high-throughput, automated counting of bacterial colonies. **Journal of the International Society for Advancement of Cytometry**, v. 77A, 790–797, 2010.

- COSTA, L.F.; RODRIGUES, F.A.; TRAVIESO, G. e VILLAS BOAS, P.R. Characterization of complex networks: A survey of measurements. **Advances in Physics**, v. 56, n. 1, 2007.
- COUR, T.; BÉNÉZIT, F. e SHI, J. Spectral segmentation with multiscale graph decomposition. In **Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition – CVPR’05**, v. 2, pp. 1124–1131. 2005.
- CVETKOVIC, D.; ROWLINSON, P. e SIMIC, S. **An introduction to the theory of graph spectra**. Cambridge University Press, 2010.
- DAVIS, J. e GOADRICH, M. The relationship between precision-recall and roc curves. In **ICML ’06: Proceedings of the 23rd international conference on Machine learning**, pp. 233–240. ACM, 2006.
- DHILLON, I.S.; GUAN, Y. e KULIS, B. Kernel k-means: spectral clustering and normalized cuts. In **Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, pp. 551–556. 2004.
- EULER, L. The seven bridges of köenisberg. **Mathematics in the Modern World**, pp. 141–144, 1968.
- FERREIRA, A. C. B. **Um Estudo Comparativo de Segmentação de Imagens por Aplicações do Corte Normalizado em Grafos**. 2011. Master. University of Campinas, Campinas, São Paulo, Brazil.
- FIEDLER, M.A. Property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. **Czech Math Journal**, v. 25, n. 100, 619–633, 1975.
- GILAT, A. **MATLAB: com aplicações em engenharia**. Bookman, 4<sup>a</sup> ed., 2012.
- GOLDBARG, M. e GOLDBARG, E. **Grafos: conceitos, algoritmos e aplicações**. Elsevier, 2012.
- GONZALEZ, R. e WOODS, R. **Processamento digital de imagens**. Addison–Wesley, 3<sup>a</sup> ed., 2010.
- GOWER, J.C. e ROSS, G.J.S. Minimum spanning trees and single linkage cluster analysis. **Applied statistics**, pp. 54–64, 1969.
- GUATERRY, S. e MILLER, G.L. On the performance of the spectral graph partitioning methods. In **Proceedings of the sixth annual ACM-SIAM: Symposium on Discrete algorithms**, pp. 233–242. San Francisco, California, 1995.
- MA, X. e WAN, W. Texture image segmentation on improved watershed and multiway spectral clustering. In **Proceedings of International Conference on Audio, Language and Image Processing – ICALIP**, pp. 1693–1697. 2008.
- MANSBERG, H.P. Automatic particle and bacterial colony counter. **Science**, v. 126, 823–827, 1957.

- MARTIN, D.; FOWLKES, C.; TAL, D. e MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In **Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on**, v. 2, pp. 416–423. 2001.
- MOSOROV, V. e KOWALSKI, T.M. The development of component tree for grayscale image segmentation. In **Proceedings of International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science – TECSET**, pp. 252–253. Slavska, Ukraine, 2002.
- NAJMAN, L. e COUPRIE, M. Building the component tree in quasi-linear time. **IEEE Transactions on Image Processing**, v. 15, n. 11, 3531–3539, 2006.
- PEDRINI, H. e SCHWARTZ, W.R. **Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações**. Thomson, 2008.
- PENG, B.; ZHANG, L. e ZHANG, D. A survey of graph theoretical approaches to image segmentation. **Pattern Recognition**, v. 46, n. 3, 1020–1038, 2013.
- SAMET, H. The quadtree and related hierarchical structures. **ACM Computing Surveys**, v. 12, n. 2, 187–261, 1984.
- SARKAR, S. e BOYER, K.L. Quantitative measures of change based on feature organization: eigenvalues and eigenvectors. In **Proceedings of IEEE Conference on Computer Vision and Pattern Recognition**. 1996.
- SHI, J. e MALIK, J. Normalized cuts and image segmentation. In **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, pp. 731–737. 1997.
- SHI, J. e MALIK, J. Normalized cuts and image segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 1, 888–905, 2000.
- TORTORA, G.J.; FUNKE, B.R. e CASE, C.L. **Microbiologia**. Artmed, 8<sup>a</sup> ed. ed., 2005.
- WANG, S. e SISKIND, J.M. Image segmentation with minimum mean cut. In **Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on**, v. 1, pp. 517–524. 2001.
- WENBING, T. *et al.* Image thresholding using graph cuts. **IEEE Transactions on Systems, Man and Cybernetics**, v. 38, n. 11, 1181–1182, 2008.
- WILSON, R.J. e WATKINS, J.J. **Graphs, an introductory approach**. John Wiley & Sons, 1990.
- WU, J. Cluster analysis and k-means clustering: An introduction. **Advances in K-means Clustering**, pp. 1–16, 2012.
- WU, Z. e LEAHY, R. An optimal graph theoretic approach to data clustering. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 15, n. 11, 1101–1113, 1993.