



TIAGO WILLIAN PINTO

**SEGMENTAÇÃO DE IMAGENS DIGITAIS COMBINANDO  
WATERSHED E CORTE NORMALIZADO EM GRAFOS**

Limeira

2014





UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Tecnologia

TIAGO WILLIAN PINTO

SEGMENTAÇÃO DE IMAGENS DIGITAIS COMBINANDO WATERSHED E CORTE  
NORMALIZADO EM GRAFOS

Dissertação apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestre em Tecnologia, na Área de Tecnologia e Inovação.

Orientador: Prof. Dr. Marco Antonio Garcia de Carvalho  
Co-orientador: Prof. Dr. Paulo Sérgio Martins Pedro

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL  
DISSERTAÇÃO DEFENDIDA PELO ALUNO  
TIAGO WILLIAN PINTO, E ORIENTADO PELO  
PROF. DR. MARCO ANTONIO GARCIA DE CARVALHO

---

Limeira  
2014

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Faculdade de Tecnologia  
Vanessa Evelyn Costa - CRB 8/8295

P658s Pinto, Tiago Willian, 1985-  
Segmentação de imagens digitais combinando watershed e corte normalizado em grafos / Tiago Willian Pinto. – Limeira, SP : [s.n.], 2014.

Orientador: Marco Antonio Garcia de Carvalho.  
Coorientador: Paulo Sérgio Martins Pedro.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Tecnologia.

1. Segmentação de imagens. 2. Corte de grafos. 3. Corte normalizado. I. Carvalho, Marco Antonio Garcia de. II. Pedro, Paulo Sérgio Martins. III. Universidade Estadual de Campinas. Faculdade de Tecnologia. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Digital image segmentation combining watershed and normalized cut

Palavras-chave em inglês:

Image segmentation

Graph cuts

Normalized cut

Área de concentração: Tecnologia e Inovação

Titulação: Mestre em Tecnologia

Banca examinadora:

Marco Antonio Garcia de Carvalho [Orientador]

Francisco de Assis Zampirolli

Leticia Rittner

Data de defesa: 26-02-2014

Programa de Pós-Graduação: Tecnologia

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

Segmentação de imagens digitais combinando Watershed e corte normalizado em grafos.

**Tiago Willian Pinto**

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



---

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



---

Prof. Dr. Francisco de Assis Zampiroli  
UFABC



---

Profa. Dra. Leticia Rittner  
FEEC/UNICAMP



## Resumo

Em Visão Computacional, a importância da segmentação de imagens é comparável apenas à sua complexidade. Interpretar a semântica de uma imagem com exatidão envolve inúmeras variáveis e condições, o que deixa um vasto campo em aberto aos pesquisadores. O intuito deste trabalho é implementar um método de segmentação de imagens através da combinação de quatro técnicas de computação: A Transformação Watershed, o Watershed Hierárquico, o *Contextual Spaces Algorithm* e o Corte Normalizado.

A Transformação Watershed é uma técnica de segmentação de imagens do campo da Morfologia Matemática baseada em crescimento de regiões e uma forma eficiente de implementá-la é através da Transformada Imagem-Floresta. Esta técnica produz uma super-segmentação da imagem, o que dificulta a interpretação visual do resultado. Uma das formas de simplificar e reduzir essa quantidade de regiões é através da construção de um espaço de escalas chamado Watershed Hierárquico, que agrupa regiões através de um limiar que representa uma característica do relevo. O *Contextual Spaces Algorithm* é uma técnica de reclassificação utilizada no campo de Busca de Imagens Baseado em contexto, e explora a similaridade entre os diferentes objetos de uma coleção através da análise do contexto entre elas. O Corte Normalizado é uma técnica que explora a análise do grau de dissimilaridade entre regiões e tem suas bases na teoria espectral dos grafos. O Watershed Hierárquico é uma abordagem multiescala de análise das regiões do watershed, que possibilita a extração de métricas que podem servir de subsídio para aplicação do Corte Normalizado.

A proposta deste projeto é combinar estas técnicas, implementando um método de segmentação que explore os benefícios alcançados por cada uma, variando entre diferentes métricas do Watershed Hierárquico com o Corte Normalizado e comparando os resultados obtidos.

## Abstract

In computer vision, the importance of image segmentation is comparable only by its complexity. Interpreting the semantics of an image accurately involves many variables and conditions, which leaves a vast field open to researchers. The purpose of this work is to implement a method of image segmentation by combining four computing techniques: The Watershed Transform, the Hierarchical Watershed, Contextual Spaces Algorithm and Normalized Cut.

The Watershed Transform is a technique for image segmentation from the field of Mathematical Morphology based on region growing and an efficient way to implement it is through the Image Foresting Transform. This technique produces an over-segmented image, which makes the visual interpretation of the result be very hard. One way to simplify and reduce the quantity of regions is by constructing a space of scales called Hierarchical Watershed, grouping regions through a threshold that represents a characteristic of the relief. The Contextual Spaces Algorithm is a reranking

technique used in the field of Context Based Image Retrieval, and explores the similarity between different objects in a collection by analyzing the context between them. Normalized Cut is a technique that exploits the analysis of the degree of dissimilarity between regions and has its foundations in the spectral graph theory. The Hierarchical Watershed is a multiscale approach for analyzing regions of the watershed, which enables the extraction of metrics that can serve as a basis for applying the Normalized Cut.

The purpose of this project is to combine these techniques, implementing a segmentation method that exploits the benefits achieved by each one, varying between different metrics of Hierarchical Watershed with Normalized Cut and comparing the results.

# Sumário

<b>Resumo</b>	<b>vii</b>
<b>Sumário</b>	<b>ix</b>
<b>Dedicatória</b>	<b>xi</b>
<b>Agradecimentos</b>	<b>xiii</b>
<b>Lista de Figuras</b>	<b>xv</b>
<b>Lista de Tabelas</b>	<b>xvii</b>
<b>Lista de algoritmos</b>	<b>xviii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos . . . . .	2
1.2 Organização do texto . . . . .	2
<b>2 Fundamentação teórica</b>	<b>3</b>
2.1 Conceitos básicos em grafos . . . . .	3
2.2 Representação Matricial . . . . .	5
2.3 Representação Imagem-Grafo . . . . .	7
2.3.1 Imagem Digital . . . . .	7
2.3.2 Grafo baseado na Transformada de <i>Watershed</i> . . . . .	8
2.3.3 Transformada Imagem Floresta . . . . .	11

2.4	Particionamento em Grafos . . . . .	15
2.4.1	Particionamento em Grafos via Corte Normalizado . . . . .	16
2.5	Distância entre imagens e <i>ranking</i> . . . . .	17
2.5.1	Distância e conceitos fundamentais . . . . .	17
2.5.2	<i>Contextual Spaces Algorithm</i> . . . . .	19
<b>3</b>	<b>Segmentação de imagens usando corte normalizado</b>	<b>21</b>
3.1	Trabalhos relacionados . . . . .	21
3.2	Método de segmentação proposto . . . . .	22
3.3	Plano de trabalho executado . . . . .	23
<b>4</b>	<b>Experimentos e resultados</b>	<b>27</b>
4.1	Utilização de hardware e software . . . . .	27
4.2	Parametrização e configuração . . . . .	28
4.3	<i>Benchmark</i> para análise dos resultados . . . . .	31
4.4	Resultados obtidos e comparação . . . . .	33
4.5	Resultados para o método proposto . . . . .	34
4.6	Resultados com utilização da reclassificação . . . . .	37
4.7	Resultados de tempo de processamento . . . . .	41
<b>5</b>	<b>Conclusões</b>	<b>42</b>
5.1	Trabalhos futuros . . . . .	43
	<b>Referências Bibliográficas</b>	<b>44</b>
<b>A</b>	<b>Publicações</b>	<b>47</b>

# Dedicatória

Dedico este trabalho a minha filha Clara, que veio iluminar minha vida com o maior amor e a maior felicidade que alguém pode sentir.



# Agradecimentos

Agradeço a minha esposa, companheira e amiga Roberta, a quem sempre recorro nos momentos difíceis. Por estar comigo durante toda a minha trajetória, ajudando a enfrentar os desafios mesmo quando as dificuldades pareciam não ter fim.

Aos meus pais Ademar e Sonia, pelo amor e dedicação com que me educaram, fornecendo a base necessária para que eu pudesse encarar a vida.

Aos professores Dr. Paulo Sérgio Martins Pedro e Dr. Daniel Carlos Guimarães Pedronette, que me auxiliaram em tudo que precisei na reta final deste trabalho.

E agradeço também meu orientador Prof. Dr. Marco A. G. de Carvalho, a quem tenho grande respeito e admiração. Desde minha graduação ele vem me instruindo, indicando caminhos e, principalmente, motivando nos momentos decisivos.



# Lista de Figuras

2.1	Exemplo de grafo com três componentes conexos. . . . .	5
2.2	Exemplos de grafos: (a) Grafo $G$ , não direcionado com cardinalidade $n = 5$ e $m = 6$ , destacando o grau do vértice $V_4$ ; (b) Exemplo de caminho no grafo $G$ com início em $V_1$ e término em $V_5$ ; (c) Subgrafo conexo de $G$ (árvore geradora) com cardinalidade $n = 5$ e $m = 4$ . . . . .	5
2.3	Representação matricial: (a) Matriz de Adjacência; (b) Matriz de Similaridade; (c) Matriz de Graus; (d) Matriz Laplaciana. . . . .	7
2.4	Imagem Digital: (a) Imagem em escala de cinza; (b) representação matricial de (a), onde o valor de cada entrada corresponde ao nível de cinza do <i>pixel</i> de posição correspondente na imagem. . . . .	8
2.5	Tipos de vizinhança em relação a um <i>pixel</i> : (a) Vizinhança-4; (b) Vizinhança-8. . . . .	8
2.6	Exemplo de zonas planas e mínimos: (a) Algumas Zonas Planas; (b) Mínimos Regionais; (c) Mínimo Global. . . . .	9
2.7	Princípio da imersão de uma imagem: (a) imagem original e seus mínimos; (b) linhas de divisórias entre as regiões; (c) imagem imersa. . . . .	10
2.8	Exemplo de gradiente morfológico: (a) Imagem original em tons de cinza; (b) gradiente morfológico de (a). . . . .	10
2.9	Regiões <i>watershed</i> para a imagem da Figura 2.8: (a) <i>Watershed</i> clássico; (b) <i>Watershed</i> hierárquico reduzindo (a) para 25 regiões utilizando o volume como critério de extinção. . . . .	11
2.10	Exemplo de formação do grafo a partir das regiões do <i>watershed</i> . Em destaque as regiões associadas a um vértice e sua vizinhança . . . . .	11
2.11	(a) Grafo de uma imagem com vizinhança-4. Os números correspondem às intensidades $I(p)$ dos <i>pixels</i> e os pontos maiores denotam três sementes. (b) Floresta de caminhos de custo mínimo usando $c_{max}$ , com $\delta(p, q) =  I(p) - I(q) $ . Os números são os valores de custo e as setas apontam o predecessor no caminho ótimo. . . . .	13
3.1	Diagrama de blocos do método de segmentação proposto. . . . .	25

3.2	Exemplo de aplicação do método para uma imagem de teste. . . . .	26
4.1	Exemplo gráfico ilustrando os modelos testados no trabalho. . . . .	31
4.2	Imagens com maior nota na métrica medida-F para o experimento WHNCut-Volume-500: (a) Código da imagem; (b) Imagem Original; (c) <i>Ground truth</i> (1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Nota para medida-F . . . . .	36
4.3	Imagens com maior nota na métrica medida-F para o experimento WHNCut-Moda-1000: (a) Código da imagem; (b) Imagem Original; (c) <i>Ground truth</i> (1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Nota para medida-F . . . . .	37
4.4	Gráfico comparativo entre os resultados para o método proposto e com a utilização de <i>reranking</i> (segmentação final em 10 regiões). . . . .	38
4.5	Gráfico comparativo entre os resultados para o método proposto e com a utilização de <i>reranking</i> (segmentação final em 30 regiões). . . . .	38
4.6	Imagens com melhor resultado após <i>reranking</i> para pacote de teste: (a) Código da imagem; (b) Imagem original; (c) <i>Ground truth</i> (1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Segmentação utilizando <i>reranking</i> ; (g) Bordas de (f) . . . . .	39
4.7	Imagens com melhor resultado após <i>reranking</i> para pacote de teste: (a) Código da imagem; (b) Imagem original; (c) <i>Ground truth</i> (1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Segmentação utilizando <i>reranking</i> ; (g) Bordas de (f) . . . . .	40
4.8	Imagens com melhor resultado após <i>reranking</i> para pacote de treinamento: (a) Código da imagem; (b) Imagem original; (c) <i>Ground truth</i> (1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Segmentação utilizando <i>reranking</i> ; (g) Bordas de (f) . . . . .	40

# Lista de Tabelas

4.1	Nomenclatura dada aos modelos testados para o método de segmentação proposto.	34
4.2	Quadro comparativo dos modelos para os conjuntos de Validação e Teste (Segmentação final em 10 e 30 regiões). Ordenação decrescente pela métrica medida-F do pacote de teste e melhores notas destacadas em negrito. . . . .	35
4.3	Quadro com tempo médio de execução (em segundos) de cada etapa do método para segmentação final em 30 regiões. Ordenação decrescente pelo campo Total. .	41

# Lista de Algoritmos

1	Algoritmo para modelamento da Transformada <i>Watershed</i> via IFT. . . . .	14
---	--	----



# Capítulo 1

## Introdução

A Visão Computacional é a área da computação que tem por objetivo fornecer subsídios para que as máquinas tenham capacidade de visualizar uma imagem e interpretar os objetos que fazem parte dessa cena, simulando o sentido da visão de um ser humano. Esta disciplina está em desenvolvimento, acompanhando a crescente demanda de novas necessidades, o que proporciona inúmeras possibilidades de aplicações e um vasto campo de estudo ainda em aberto aos pesquisadores.

Dentre as áreas de estudo inerentes está o Processamento Digital de Imagens (PDI), que faz o tratamento e o reconhecimento de uma imagem com dois objetivos principais: Melhoria da informação visual para a interpretação humana e processamento dos dados para percepção automática através de máquinas.

Em PDI, confere-se grande importância ao estudo de Segmentação de Imagens, tarefa que consiste em subdividir uma imagem em regiões significativas, agrupando pixels de acordo com um critério pré-estabelecido para formar objetos onde seja possível extrair um conteúdo semântico, passível de interpretação num determinado contexto. Resumindo, o objetivo final da segmentação é simplificar a imagem a certo nível que seja possível realizar operações computacionais e/ou matemáticas no domínio de um problema a fim de solucioná-lo.

As técnicas de segmentação em geral se dividem de acordo com a estratégia utilizada para particionar a imagem, baseando-se em detecção de bordas/descontinuidades e similaridades/crescimento de regiões.

Contudo, a segmentação automática é uma das tarefas mais difíceis em PDI, dado o grande número de aplicações existentes, gerando uma contínua busca por soluções que consigam atender às premissas: Eficiência computacional, robustez e escalabilidade. Geralmente, soluções que atingem os melhores resultados em segmentação de imagens são específicas a um problema, falhando

quando aplicadas em cenários diversos. Desta forma, não é comum encontrar soluções genéricas, onde a preocupação é manter uma mesma qualidade no resultado independente do problema ao qual são aplicadas.

## 1.1 Objetivos

A meta principal deste trabalho é abordar duas técnicas conhecidas de segmentação de imagens: O *watershed* (implementado pela Transformada Imagem Floresta) e o Corte Normalizado, ambas solidamente documentadas na literatura científica. Com a combinação dessas técnicas, espera-se desenvolver um método de segmentação genérico, que obtenha resultados satisfatórios (semanticamente próximos da interpretação visual humana) na segmentação de imagens de propósito geral, utilizando como base de experimentos uma coleção de imagens bastante difundida para esta finalidade: o *Berkeley Image Dataset*, que também possui um *benchmark* para classificação dos métodos de segmentação que o utilizam. Adicionalmente, também será utilizada uma técnica chamada *Contextual Spaces Algorithm*. Esta técnica, originada da área de Recuperação de Imagens Baseada em Conteúdo, será combinada junto as demais técnicas anteriores com o objetivo de melhorar a qualidade de segmentação final das imagens.

## 1.2 Organização do texto

Este trabalho está organizado nos capítulos que seguem. O Capítulo 1 traz breve introdução sobre o tema abordado. O Capítulo 2 discorre sobre conceitos das técnicas e fundamentação teórica necessários para entendimento do trabalho. O capítulo 3 apresenta o método de segmentação proposto e trabalhos relacionados, enquanto o capítulo 4 traz os resultados e discussão obtidos pela implementação desse método. Finalmente, no capítulo 5 é apresentadas as conclusões e trabalhos futuros.

# Capítulo 2

## Fundamentação teórica

Neste capítulo serão abordados os fundamentos teóricos que serviram de base para a definição e implementação do método de segmentação proposto neste trabalho.

### 2.1 Conceitos básicos em grafos

A Teoria dos Grafos (TG) é uma importante área da matemática discreta, amplamente utilizada na análise, modelagem e resolução de problemas complexos em diversas áreas do conhecimento. Sua origem é atribuída aos estudos do matemático e físico suíço Leonhard Euler, que em 1735 propôs uma solução para um problema da época conhecido como *as sete pontes de Königsberg*. Desde então, diversos estudiosos e pesquisadores contribuíram para o crescimento da TG e, atualmente, sua utilização para resolução de problemas computacionais é muito comum, sendo que em Processamento de Imagens Digitais, sua aplicação em técnicas de segmentação é recorrente.

Pela grande abrangência da TG, nesta seção apresentamos apenas as definições e notações necessárias para entendimento do texto, baseado nos trabalhos de (GONDRAN E MINOUX, 1984; WILSON E WATKINS, 1990; THULASIRAMAN E SWAMY, 2011). Para um estudo mais detalhado em TG, recomenda-se a leitura completa deste material.

Um grafo  $G = (V, E)$ , ou simplesmente  $G$ , é uma estrutura algébrica, onde  $V$  é o conjunto discreto, cujos elementos são denominados de vértices de  $G$  e  $E$  é o conjunto de subconjuntos a dois elementos de  $V$ , denominados de arestas de  $G$ . Dois vértices  $i$  e  $j \in V$  são adjacentes (ou vizinhos), representado por  $i \sim j$ , se estiverem conectados por uma aresta  $(i, j) \in E$ . Um grafo ponderado é um grafo onde um valor (peso) é associado para cada aresta  $(i, j)$ , representado por  $w(i, j)$ . Neste caso, complementa-se a definição para  $G = (V, E, W)$ , onde  $W$  representa o conjunto de pesos entre

as arestas. A ordem representa a cardinalidade do conjunto de vértices do grafo e é denotada por  $|V(G)|$  ou simplesmente  $n$ . A cardinalidade do conjunto de arestas do grafo é denotado por  $|E(G)|$  ou simplesmente  $m$ .

Considera-se um grafo como completo, denotado por  $G = K_n$ , se todos os vértices de  $G$  são mutuamente adjacentes, alcançando o número máximo de arestas possível, dado por  $n * (n - 1)/2$ . Sendo assim, a densidade de um grafo é a razão entre a quantidade de arestas do grafo e a quantidade de arestas do grafo completo com a mesma quantidade de vértices, dado por  $2m/n$ . Um grafo é considerado denso quando a quantidade de arestas for próxima ao máximo e considerado esparsa caso contrário.

Para grafos não direcionais, uma aresta  $(i, j) \in E$  é considerada incidente a  $i$  e a  $j$ , então, dado um vértice  $i \in V(G)$ , o Grau desse vértice (ou valência), denotado por  $d(i)$  é o número de arestas incidentes em  $i$  ou  $|E(i)|$ . Esta definição satisfaz a necessidade deste trabalho, pois apenas grafos não direcionais serão utilizados. Quando existe necessidade de indicar um sentido para a aresta (orientação), o grafo é dito direcional (orientado) e o conceito de Grau deve ser expandido para considerar as informações de origem e destino das arestas.

Um caminho de  $i_1$  até  $i_n$  é uma lista  $(i_1, i_2, \dots, i_{n-1}, i_n)$  onde  $i_k \sim i_{k+1}$ ,  $k = 1, \dots, n - 1$ . Dois vértices  $i$  e  $j$  estão conectados se existir pelo menos um caminho entre  $i$  e  $j$ . Um grafo é dito conexo quando para quaisquer pares de vértices  $i, j \in V$  é possível determinar um caminho que começa em  $i$  e termina em  $j$ . Um ciclo é definido como um caminho que começa e termina no mesmo vértice, da forma  $i_1, i_2, \dots, i_{n-1}, i_n, i_1$ , ou  $i_1 \sim i_1$ .

Um grafo  $G'$  é um subgrafo de  $G$  se  $V(G') \subset V(G)$  e  $E(G') \subset E(G)$ . Seja  $G'$  um subgrafo de  $G$ . Componentes Conexos são subconjuntos disjuntos de vértices em  $G$  conectados entre si. Um componente conexo de  $G'$  é, por definição, o subgrafo gerado pelo conjunto de todos os outros vértices (incluindo o próprio) que lhe estão conectados, ou seja, o componente conexo que contém um vértice  $i$  é a união de todos os caminhos em  $G'$  que tem origem em  $i$ . Segue um exemplo na Figura 2.1:

Um grafo conexo sem ciclos é chamado de árvore. Por estas características específicas, pode-se designar um vértice para ser a raiz da árvore e estabelecer uma relação lógica de hierarquia entre os vértices, onde a distância entre cada um deles e a raiz é denominada nível. Seja  $T$  uma árvore com  $n$  vértices, então as seguintes proposições são válidas:

1.  $T$  é conexa, não contém ciclos e possui  $n - 1$  arestas;
2. Cada aresta de  $T$  é uma aresta de corte, ou seja, se uma aresta for removida o grafo deixará de ser conexo;

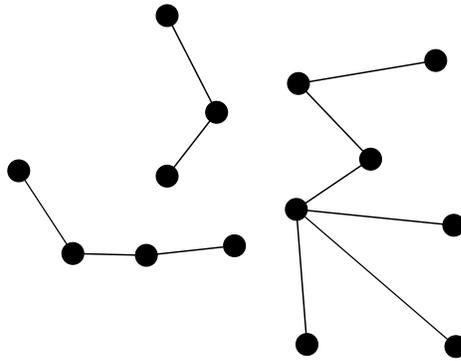


Figura 2.1: Exemplo de grafo com três componentes conexos.

3. Quaisquer dois vértices de  $T$  são conectados por somente um caminho;
4. A adição de qualquer nova aresta cria exatamente um ciclo.

Uma *Árvore Geradora*  $T$ , é um subgrafo de  $G$  tal que  $V(T) \subset V(G)$ . Uma árvore geradora cuja a soma dos pesos de seus arcos seja menor do que em qualquer outra situação é chamada de *árvore geradora mínima*. Uma floresta é um grafo onde as componentes são árvores.

A maneira mais simples de visualizar um grafo é descrevê-lo graficamente, representando cada vértice por um ponto num plano e suas arestas por linhas ligando os pontos correspondentes. Na Figura 2.2 são exibidos exemplos de grafos.

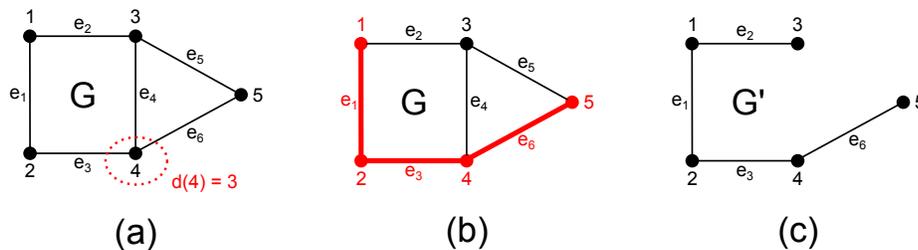


Figura 2.2: Exemplos de grafos: (a) Grafo  $G$ , não direcionado com cardinalidade  $n = 5$  e  $m = 6$ , destacando o grau do vértice  $V_4$ ; (b) Exemplo de caminho no grafo  $G$  com início em  $V_1$  e término em  $V_5$ ; (c) Subgrafo conexo de  $G$  (árvore geradora) com cardinalidade  $n = 5$  e  $m = 4$ .

## 2.2 Representação Matricial

Para que um grafo possa ser analisado computacionalmente ele precisa de uma representação matemática e para isso são utilizadas estruturas de dados que possam ser interpretadas pelos algoritmos. Em geral, a utilização de matrizes é muito útil nesta representação. A *Matriz de Adjacência*, ou

vizinhança, tem a finalidade de mapear as conexões entre todos os vértices de um grafo  $G$ , ela é definida conforme a Equação 2.1 :

$$A(i, j) = \begin{cases} 1, & \text{se } i \sim j, \\ 0, & \text{caso contrário.} \end{cases} \quad (2.1)$$

Quando um grafo é ponderado, a Matriz de Adjacência é preenchida com o valor do peso da aresta. Neste trabalho o peso das arestas representa uma função de similaridade entre os vértices. Sendo assim, deste ponto em diante vamos chamar esta matriz de *Matriz de Similaridade*. Dada a Matriz de Adjacência  $A$ , obtemos a Matriz de Similaridade  $S$  substituindo cada entrada  $a(i, j)$  pelo peso da aresta  $w(i, j)$ . Tal matriz é definida como se segue na Equação 2.2 :

$$S(i, j) = \begin{cases} w(i, j), & \text{se } i \sim j \\ 0, & \text{caso contrário.} \end{cases} \quad (2.2)$$

Lembrando que o grau de um vértice é o total de conexões com os vértices do grafo adjacentes a ele, e assumindo que  $d(i)$  é a somatória dos pesos das arestas incidentes a esse vértice, tem-se a Equação 2.3 :

$$d(i) = \sum w(i, j), \quad (2.3)$$

então a *Matriz de Graus* é uma matriz diagonal  $D$  com  $d$  em sua diagonal principal.

A *Matriz Laplaciana*, cuja importância está relacionada ao espectro do grafo e sua estrutura (segundo princípios da Teoria Espectral dos Grafos, Seção 2.2), é dada pela seguinte Equação 2.4 :

$$L(G) = (D - S), \quad (2.4)$$

onde  $D$  é a Matriz de Graus de um grafo  $G$  e  $S$  é a Matriz de Similaridade deste mesmo grafo.

Para o grafo exibido na Figura 2.2, dado conjunto de arestas  $E(G) = e_1, e_2, e_3, e_4, e_5, e_6$ , se considerarmos os seguintes pesos  $w = 5, 7, 2, 8, 1, 6$ , respectivamente, teremos as seguintes estruturas de dados ilustradas na Figura 2.3.

$$\begin{array}{cccc}
 \left[ \begin{array}{ccccc} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{array} \right] & \left[ \begin{array}{ccccc} 0 & 5 & 7 & 0 & 0 \\ 5 & 0 & 0 & 2 & 0 \\ 7 & 0 & 0 & 8 & 1 \\ 0 & 2 & 8 & 0 & 6 \\ 0 & 0 & 1 & 6 & 0 \end{array} \right] & \left[ \begin{array}{ccccc} 12 & 0 & 0 & 0 & 0 \\ 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 16 & 0 & 0 \\ 0 & 0 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 & 7 \end{array} \right] & \left[ \begin{array}{ccccc} 12 & -5 & -7 & 0 & 0 \\ -5 & 7 & 0 & -2 & 0 \\ -7 & 0 & 16 & -8 & -1 \\ 0 & -2 & -8 & 16 & -6 \\ 0 & 0 & -1 & -6 & 7 \end{array} \right] \\
 \text{(a)} & \text{(b)} & \text{(c)} & \text{(d)}
 \end{array}$$

Figura 2.3: Representação matricial: (a) Matriz de Adjacência; (b) Matriz de Similaridade; (c) Matriz de Graus; (d) Matriz Laplaciana.

## 2.3 Representação Imagem-Grafo

Uma das formas de se realizar a segmentação de uma imagem é através do particionamento de um grafo que represente esta imagem digital. Uma imagem digital  $I$  pode ser representada por um grafo  $G$ , através de um critério de agrupamento que associe cada *pixel* (ou grupo de pixels) de  $I$  a um vértice de  $G$ .

Nas subseções a seguir serão apresentadas as técnicas abordadas neste trabalho para se converter uma imagem digital em grafo e, posteriormente, o processo de particionamento necessário para se obter os objetos formadores desta imagem.

### 2.3.1 Imagem Digital

Uma imagem  $I$  pode ser representada por uma função bidimensional  $f(x,y)$ , onde  $x$  e  $y$  são coordenadas espaciais num plano e  $f$  a intensidade da imagem no ponto  $(x,y)$ , sendo que estas variáveis assumem valores finitos e discretos. Cada elemento de  $I$  é chamado de *pixel* (contração de *picture element*) e está associado a um ponto  $(x,y)$  e uma intensidade. (GONZALEZ E WOODS, 2008). Esta representação é feita a fim de permitir o armazenamento, transferência, impressão/reprodução e/ou processamento de uma imagem por meios computacionais.

As imagens utilizadas neste trabalho são originalmente coloridas, no sistema de cores RGB. Porém, para fins de processamento, passam por um processo de conversão para uma escala de cinza, utilizando estruturas de armazenamento de 8 bits (1 byte), o que permite intensidades entre 0 (preto) e 255 (branco). Um exemplo de imagem digital em escala de cinza é dado na Figura 2.4 .

Para um maior detalhamento em Processamento Digital de Imagens, recomenda-se a leitura de (GONZALEZ E WOODS, 2008).

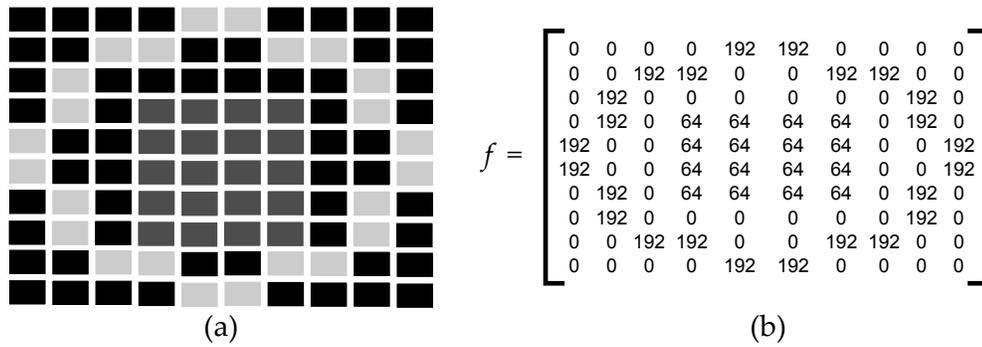


Figura 2.4: Imagem Digital: (a) Imagem em escala de cinza; (b) representação matricial de (a), onde o valor de cada entrada corresponde ao nível de cinza do *pixel* de posição correspondente na imagem.

### 2.3.2 Grafo baseado na Transformada de *Watershed*

Conceitualmente, a Transformada (ou transformação) de *Watershed* pode ser observada através das noções da imersão ou da inundação de um relevo topográfico, onde a altura da superfície será representada pelos tons de cinza formadores de uma imagem (BEUCHER, 1991). A seguir, o processo será exemplificado através da noção de imersão, mas antes, as seguintes definições são necessárias para se compreender o *watershed*:

- **Vizinhança:** Define uma relação entre os *pixels* de uma imagem. As relações de vizinhança (ou adjacência) mais utilizadas são do tipo vizinhança-4 ou vizinhança-8, conforme ilustrado na Figura 2.5.

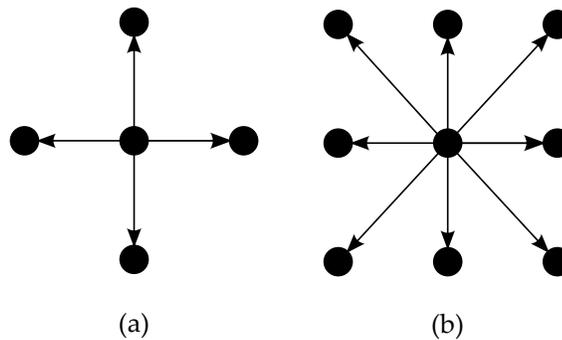


Figura 2.5: Tipos de vizinhança em relação a um *pixel*: (a) Vizinhança-4; (b) Vizinhança-8.

- **Zonas planas:** Zonas Planas são os componentes conexos da imagem que possuem mesma intensidade de nível de cinza, conforme ilustrado na Figura 2.6 (a).
- **Mínimo Regional:** Zona plana que possui vizinhos com maior intensidade em nível de cinza, conforme ilustrado na Figura 2.6 (b).

- **Mínimo Local:** É um *pixel* (ou zona plana) que não possui vizinhos de menor intensidade em nível de cinza.
- **Mínimo Global:** É um *pixel* (ou zona plana) que possui menor intensidade em nível de cinza dentre os mínimos regionais, conforme ilustrado na Figura 2.6 (c).

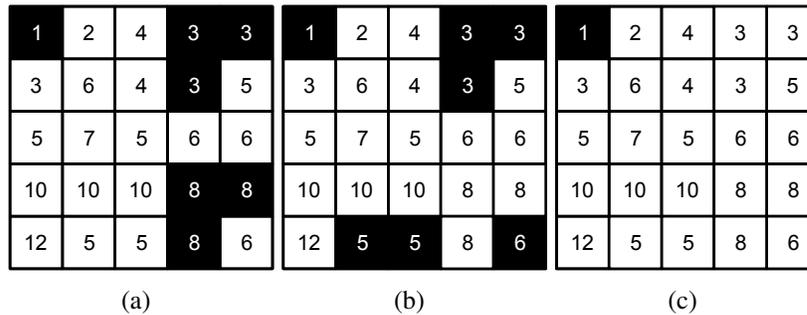


Figura 2.6: Exemplo de zonas planas e mínimos: (a) Algumas Zonas Planas; (b) Mínimos Regionais; (c) Mínimo Global.

O *Watershed*, segundo o conceito de imersão, pode ser explicado analogamente a um problema hidrológico, onde um relevo é inundado progressivamente a partir de fontes de água posicionadas nos vales. Conclui-se facilmente num contexto físico, que a água começaria a se acumular primeiro nas partes mais baixas deste relevo (Mínimos Regionais)<sup>1</sup>, formando várias bacias hidrográficas em diferentes lugares até chegarem a níveis mais altos, onde a água de várias bacias se juntaria. Os Mínimos Regionais, onde a água se acumula, formam a criação de bacias hidrográficas e cada bacia que for formada será chamada de Região. Enquanto a inundação acontece, a cada vez que uma região chegar à iminência de se fundir com outra, uma linha divisória é criada para evitar que as águas de regiões diferentes acabem se tocando. No final do processo, cada região terá formado um segmento da imagem, e as linhas divisórias o seu devido contorno.

A Figura 2.7, adaptado de (CARVALHO, 2004), exemplifica esse processo.

Esta é a abordagem clássica do *watershed* e um dos principais problemas que ela pode causar é o efeito de mosaico (super-segmentação), causado pela existência de muitos mínimos que na realidade não representam objetos semânticos.

Para atenuar esse efeito, é comum que a transformação *watershed* seja feita sobre o gradiente morfológico da imagem, uma vez que este operador ressalta o contorno dos objetos, produzindo bordas em pontos de descontinuidade de nível de cinza, efeito geralmente desejado em segmentação de imagens. O gradiente morfológico é composto de três outras operações básicas da morfologia:

<sup>1</sup>Neste trabalho (e em geral), a convenção é de que quanto menor for a intensidade do cinza, menor será a altura da superfície, e vice-versa. Portanto, pontos mínimos serão representados por menores intensidades do tom de cinza.

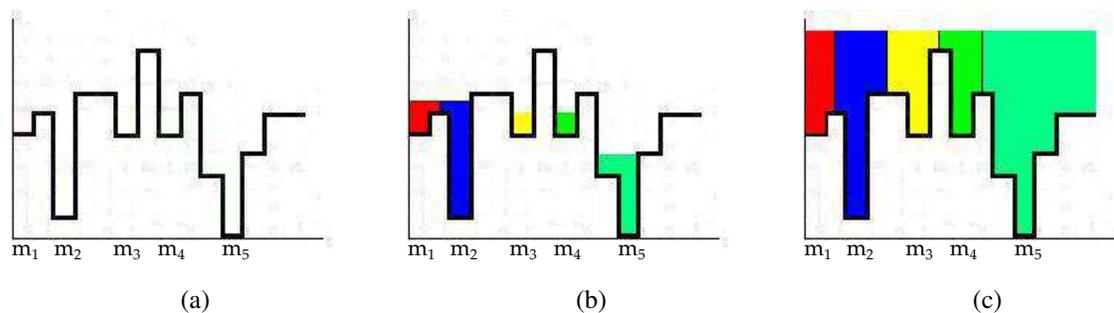


Figura 2.7: Princípio da imersão de uma imagem: (a) imagem original e seus mínimos; (b) linhas de divisórias entre as regiões; (c) imagem imersa.

a dilatação ( $\oplus$ ), a erosão ( $\ominus$ ) e a subtração como definido na Equação 2.5 (GONZALEZ E WOODS, 2008) e exemplificado na Figura 2.8:

$$grad(I) = (I \oplus b) - (I \ominus b) \quad (2.5)$$

onde  $I$  é a imagem original e  $b$  um elemento estruturante.



Figura 2.8: Exemplo de gradiente morfológico: (a) Imagem original em tons de cinza; (b) gradiente morfológico de (a).

Neste trabalho pretende-se utilizar uma abordagem chamada de *watershed* hierárquico (MEYER, 1996), que consiste em se definir uma função de extinção para os mínimos regionais, aproveitando apenas aqueles que satisfizerem essa função e descartando-se os demais (CARVALHO, 2004). Esta técnica reduz o número de regiões geradas pelo *watershed* clássico, reduzindo o problema de super-segmentação citado anteriormente, conforme exemplificado na Figura 2.9.

Após aplicação das técnicas de *watershed* descritas acima, a construção do grafo é feita pela associação de cada região obtida ao final do processo a um vértice no grafo. A adjacência desses vértices é definida por arestas que conectam regiões vizinhas, ou seja, são divididas por uma linha divisória que evita que elas se toquem. Um exemplo destacando alguns vértices obtidos da Figura 2.9 pode ser visto na Figura 2.10.

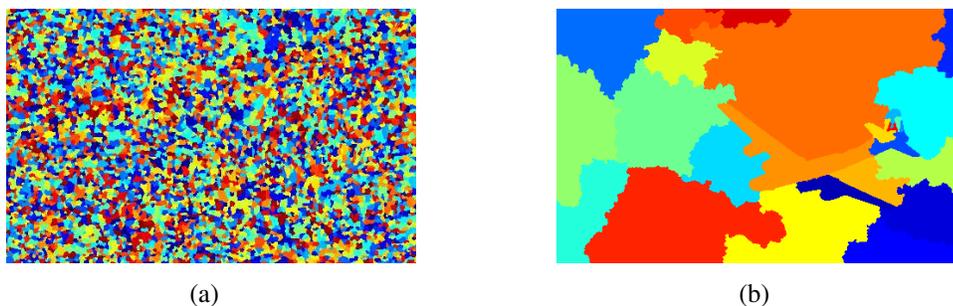


Figura 2.9: Regiões *watershed* para a imagem da Figura 2.8: (a) *Watershed* clássico; (b) *Watershed* hierárquico reduzindo (a) para 25 regiões utilizando o volume como critério de extinção.

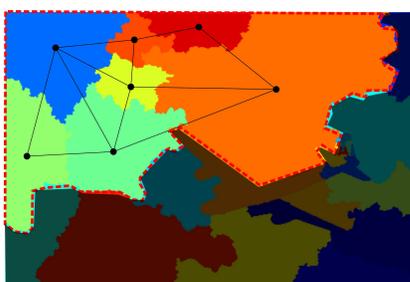


Figura 2.10: Exemplo de formação do grafo a partir das regiões do *watershed*. Em destaque as regiões associadas a um vértice e sua vizinhança

Conforme descrito na seção 2.2, este trabalho utiliza uma Matriz de Similaridade para representar a imagem analisada. Uma vez que o *watershed* hierárquico fornece critérios para definir o grau de semelhança entre as regiões, este trabalho irá explorar a métrica de volume das regiões e uma função de moda dos níveis de cinza de cada região.

### 2.3.3 Transformada Imagem Floresta

Conforme investigado por Roerdink e Meijster (2000) e mais recentemente por Körbes (2010), existem na literatura diversas formas de se definir e implementar a Transformada de *Watershed*. Neste trabalho optou-se pela utilização da Transformada Imagem Floresta, devido a sua simplicidade e eficiência na implementação, versatilidade para reutilização em outros operadores morfológicos e comprovada exatidão no resultado.

A Transformada Imagem Floresta (ou *Image Foresting Transform* – IFT), técnica proposta por (FALCÃO *et al.*, 2004), define uma ferramenta genérica para modelagem, implementação e avaliação de diversos operadores de processamento de imagem baseados em conectividade. Num grafo onde os vértices são derivados dos *pixels* da imagem, a IFT reduz o problema de processa-

mento de imagem ao cálculo de uma floresta de caminhos de custo mínimo.

**Definição:** Uma imagem pode ser vista como um grafo onde os vértices são os *pixels* e as arestas são definidas por uma relação de vizinhança entre eles (geralmente vizinhança-4 ou vizinhança-8), associando um peso a cada aresta.

Dado um conjunto de *pixels* sementes, cada semente recebe um rótulo que identifica a árvore. A IFT calcula então uma partição ótima dos vértices em árvores de custo mínimo, onde cada vértice recebe o rótulo específico da árvore à qual ele pertence, aponta de alguma maneira para seu predecessor nesta árvore e tem um custo de caminho.

Como o algoritmo calcula uma floresta ótima com árvores de custo mínimo, cada vértice tem, no final, o rótulo da semente a partir da qual existe um caminho até ele de custo mínimo.

**Função de custo:** As funções de custo na IFT são responsáveis por determinar a rotulação dos vértices, identificando qual das sementes será a raiz desse vértice. A avaliação desta função é que vai determinar o caminho de menor custo para um vértice alcançar sua semente.

**Watershed via IFT:** Para o modelamento da Transformada de *Watershed* através de uma IFT utiliza-se uma função de custo de caminho conhecida como função de máximo, como definida pela Equação 2.6 :

$$c_{max(\pi)} = \begin{cases} \max_{\forall(p,q) \in \pi} \{\delta(p,q)\}, & \text{se a origem de } \pi \in S, \\ +\infty, & \text{caso contrário,} \end{cases} \quad (2.6)$$

onde  $\pi$  é um caminho nesse grafo,  $S$  é um conjunto de vértices sementes e  $\delta(p,q)$  uma função que mede a dissimilaridade entre os vértices  $p$  e  $q$ , conforme exemplificado na Figura 2.11 (adaptado de (FALCÃO *et al.*, 2004)).

O algoritmo da IFT é essencialmente o algoritmo de Dijkstra para o cálculo de caminhos de custo mínimo a partir de uma única fonte, ligeiramente modificado para permitir fontes múltiplas e função mais geral de custo. Em (AUDIGIER, 2004) o algoritmo é didaticamente explicado em pseudocódigo, conforme Algoritmo 1.

Os seguintes parâmetros são utilizados:

*Variáveis de entrada:*

- a imagem  $I$  (conjunto de *pixels*);
- a vizinhança simétrica  $V : V(p)$  designa o conjunto dos vizinhos do pixel  $p$ ;

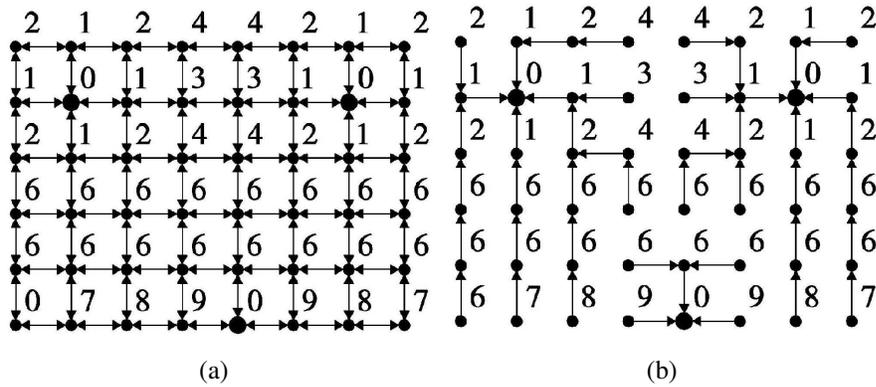


Figura 2.11: (a) Grafo de uma imagem com vizinhança-4. Os números correspondem às intensidades  $I(p)$  dos *pixels* e os pontos maiores denotam três sementes. (b) Floresta de caminhos de custo mínimo usando  $c_{max}$ , com  $\delta(p, q) = |I(p) - I(q)|$ . Os números são os valores de custo e as setas apontam o predecessor no caminho ótimo.

- o conjunto  $S$  de *pixels* sementes rotuladas;
- a função  $\lambda$  que associa um rótulo a cada semente de  $S$ ;
- uma função de custo de caminho  $f$  que associa ao caminho  $\pi_p$  o custo de percurso  $f(\pi_p)$ .

*Variáveis de saída:*

- o mapa de custos  $C$  (custos acumulados dos caminhos até cada *pixel*, em um dado instante);
- o mapa de rótulos  $R$ ;
- o mapa de predecessores  $P$ .

*Variáveis auxiliares:*

- a fila de prioridade  $Q$  vazia, gerenciada com as seguintes funções:
  - $RemoveMin(Q)$  remove da fila  $Q$ , o *pixel* de custo mínimo;
  - $Remove(p; Q)$  remove o *pixel*  $p$  da fila  $Q$ ;
  - $Inser(p; Q; c)$  insere o *pixel*  $p$  na fila  $Q$  com o custo  $c$ ;
  - $EstáVazia(Q)$  retorna um booleano indicando se a fila  $Q$  está vazia.
- a tabela  $e$  de estado dos *pixels* associando a cada *pixel*  $p$  da imagem  $I$  o valor  $e(p)$  igual a:

- 0 se o *pixel*  $p$  não foi tratado e não está na fila  $Q$ ;
  - 1 se  $p$  está em  $Q$ ;
  - 2 se  $p$  já foi tratado definitivamente (portanto, está fora da fila).
- a variável  $c$  para armazenar um custo (geralmente inteiro).

**Entrada:**  $I, V, S, \lambda, f$   
**Saída:**  $C, R, P$

```

1  $\forall p \in I, C[p] \leftarrow +\infty; R[p] \leftarrow nil; P[p] \leftarrow nil; e(p) \leftarrow 0;$ 
2  $\forall s \in S, C[s] \leftarrow 0; R[s] \leftarrow \lambda(s); Inse(re(s, Q, 0); e(s) \leftarrow 1;$ 
3 enquanto  $EstáVazia(Q) = FALSO$  faça
4    $p \leftarrow RemoveMin(Q); e(p) \leftarrow 2;$ 
5    $\forall q \in V(p),$  se  $e(q) \neq 2$  então
6      $c \leftarrow f(\pi_q \cdot \langle p, q \rangle);$ 
7     se  $c < C[q]$  então
8       se  $e(q) = 1$  então
9          $Remove(q, Q);$ 
10      fim
11       $Inse(re(q, Q, c); e(q) \leftarrow 1;$ 
12       $C[q] \leftarrow c; R[q] \leftarrow R[p]; P[q] \leftarrow p;$ 
13    fim
14  fim
15 fim

```

**Algoritmo 1:** Algoritmo para modelamento da Transformada *Watershed* via IFT.

O algoritmo começa inicializando variáveis e estruturas necessárias para seu funcionamento: todos os *pixels* da imagem, atribuindo  $+\infty$  ao custo e zerando os rótulos, predecessores e mapa de estados (linha 1) e todas as sementes, atribuindo zero ao seu custo, um rótulo e inserindo-as na fila de prioridade (linha 2).

Como a IFT trabalha com uma estrutura de dados do tipo fila hierárquica, na sequência é posto seu laço principal (linha 3) que finaliza somente com essa estrutura vazia, ou seja, todos os *pixels* processados.

Na linha 4 um *pixel*  $p$  de custo mínimo é removido da fila de prioridade para ser tratado e seu estado é marcado como processado definitivamente. Na linha 5 inicia-se um novo laço, processando todos os vizinhos de  $p$  ainda pendentes de processamento, enquanto que na linha 6 o custo para esses vizinhos é calculado.

Na linha 7 verifica-se se o custo calculado para o vizinho é menor do que já foi armazenado para

ele no mapa de custos, caso a condição seja satisfeita, ele é inserido na fila de prioridade com esse custo e seu estado é alterado na linha 10.

As linhas 8 e 9 servem apenas de controle para que um mesmo *pixel* não esteja na fila de prioridades com custos diferentes.

## 2.4 Particionamento em Grafos

Na teoria dos grafos, o particionamento de um grafo  $G$  pode ser realizado através da remoção de arestas neste grafo, tal que, ao final do processo sejam gerados conjuntos de subgrafos desconexos de  $G$ . Esse processo é conhecido como corte, e é baseado no grau de dissimilaridade entre os subgrafos resultantes conforme definido na Equação 2.7 :

$$\text{Cut}(A, B) = \sum_{i \in A, j \in B} w(i, j), \quad (2.7)$$

onde  $A$  e  $B$  são dois grupos desconexos no grafo  $G$ .

Diversas abordagens na literatura fornecem critérios de corte para particionar um grafo, tais como: o corte máximo, cuja finalidade é remover o máximo de arestas do grafo a fim de se formar dois subgrafos; o corte mais esparsa, que produz dois subgrafos minimizando o total de arestas removidas e o número de vértices do menor subgrafo; e o corte mínimo de (WU E LEAHY, 1993), que propõem a remoção mínima de arestas do grafo.

A abordagem que será descrita a seguir tem suas bases na *Teoria Espectral dos Grafos* (TEG). O estudo da TEG, em sua essência, preocupa-se com o relacionamento entre as propriedades algébricas do espectro de certas matrizes associadas com um grafo e as propriedades estruturais desse grafo (CVETKOVIĆ *et al.*, 2010).

O espectro de um grafo  $G$  é o conjunto de autovalores ( $\lambda$ ) que satisfazem a equação  $\lambda x = Ax$ , onde  $A$  é uma matriz que representa o grafo  $G$ . Para o cálculo dos autovalores deve-se encontrar as raízes da equação  $p_G(\lambda) = \det(\lambda I - A)$ , onde  $I$  é a Matriz Identidade e  $p(\lambda)$  é denominado polinômio característico (ABREU, 2005).

Dependendo da matriz  $A$  escolhida para representar o grafo  $G$ , o espectro desse grafo pode fornecer diferentes informações sobre sua estrutura, o que permite a aplicação da TEG em diversos problemas envolvendo grafos. Neste trabalho o foco é relacionado ao espectro da Matriz Laplaciana.

### 2.4.1 Particionamento em Grafos via Corte Normalizado

Um critério de corte proposto por (SHI E MALIK, 1997), chamado de Corte Normalizado (do inglês *Normalized Cut* - Ncut) usa as propriedades algébricas da Matriz Laplaciana para separar os vértices de um grafo de acordo com o grau de dissimilaridade entre eles. Este método calcula o custo do corte como uma fração de todas as conexões entre as arestas para todos os nós do grafo:

$$\text{NCut}(A, B) = \frac{\text{Cut}(A, B)}{\text{Assoc}(A, G)} + \frac{\text{Cut}(A, B)}{\text{Assoc}(B, G)}, \quad (2.8)$$

onde  $\text{Assoc}(A, G) = \sum_{i \in A, j \in G} w(i, j)$  é o total de conexões dos nós em  $A$  para o restante do grafo e  $\text{Assoc}(B, G)$  é similarmente definido.

Conforme descrito por Shi e Malik (1997) o Ncut pode ser minimizado resolvendo um sistema de autovalores genérico  $Lx = \lambda Dx$ , onde  $L$  é a Matriz de Laplaciana e  $D$  a Matriz de Graus do grafo  $G$  (ambas estruturas definidas na seção 2.2). Como o menor autovalor de  $L$  é sempre 0, o segundo menor autovalor é a solução real para o problema do Ncut e o autovetor correspondente determina exatamente como particionar o grafo.

Os demais autovetores também contém informações relevantes de como se particionar o grafo (e.g. O autovetor com terceiro menor autovalor é a solução real para subdividir as duas partições iniciais), porém quanto maior o autovetor, menor a confiabilidade que se tem na informação obtida para particionar o grafo (SHI E MALIK, 1997).

Shi e Malik (1997) propõe duas formas de se particionar um grafo através do NCut:

*Recursive 2-way NCut (2-way)*: Modo iterativo, onde cada região da imagem é recursivamente particionada utilizando o autovetor correspondente ao segundo menor autovalor do grafo em duas novas regiões. A cada nova iteração, este cálculo é feito a partir de uma nova matriz que representa os respectivos subgrafos. A vantagem deste método é a precisão obtida por se utilizar apenas do autovetor referente ao segundo menor autovalor calculado.

*Simultaneous k-way Cut with Multiple Eigenvectors (k-way)*: Neste modo o particionamento é realizado diretamente em um dado número  $k$  de regiões, através da utilização dos  $k$  autovetores calculados de uma única vez a partir da matriz que representa o grafo. A vantagem deste método é evitar o desperdício de esforço computacional, conseguindo particionar o grafo numa única iteração.

Para ambas as abordagens, os passos do algoritmo são os mesmos, diferenciando-se apenas no passo 4:

1. Dada uma imagem  $I$ , construir um grafo  $G = (V, E)$ , calcular o peso entre as arestas conectando dois vértices de acordo com alguma medida de similaridade e sumarizar as informações nas matrizes  $S$  e  $D$ ;
2. Resolver  $Lx = \lambda Dx$  para autovetores com os menores autovalores;
3. Usar o autovetor correspondente ao segundo menor autovalor para dividir o grafo em duas partições. Este particionamento é realizado separando os vértices representados neste autovetor através de um limiar  $t$  estabelecido. Por exemplo, para um limiar  $t = 0$ , o particionamento dos vértices se dará pela separação das entradas do autovetor entre valores positivos e negativos;
4. **2-way**: Decidir se as partições correntes devem ser subdivididas novamente e recursivamente particioná-las  
ou  
**k-way**: fazer uso dos demais autovetores para particionar o grafo em  $k$  regiões.

## 2.5 Distância entre imagens e *ranking*

Nesta seção será apresentada uma breve introdução sobre os conceitos envolvidos em Sistema de Recuperação de imagens e a técnica utilizada neste trabalho para reclassificar os pesos da Matriz de Similaridade utilizada no processo de segmentação.

### 2.5.1 Distância e conceitos fundamentais

A definição de medidas adequadas da distância entre imagens desempenha um papel fundamental em muitas aplicações multimídia, classificação, *clustering* e tarefas de recuperação. A escolha de uma medida boa de distância muitas vezes é fundamental para a construção de um Sistema de Recuperação de Imagens Baseado em Conteúdo (do inglês *Content Based Image Retrieval* – CBIR) (PEDRONETTE *et al.*, 2012).

Um sistema CBIR visa atender as necessidades do usuário expressos na especificação da consulta, por exemplo, através da definição de uma imagem de consulta como entrada. O método geralmente aplicado para alcançar o seu objetivo baseia-se em recuperar as imagens mais similares em uma coleção, tendo em conta as propriedades das imagens visuais, tais como forma, cor e textura. Uma coleção de imagens é classificada por ordem de similaridade decrescente, de acordo com um determinado descritor de imagem, formando o *ranking*. Em geral, os sistemas de busca

por conteúdo, muitas vezes executam a análise de imagens pares, ou seja, computação de similaridade (ou distância) considerando apenas pares de imagens, ignorando a informação codificada nas relações entre várias imagens.

Com o objetivo de recuperar as imagens semelhantes a uma imagem de consulta os sistemas de busca por conteúdo calculam uma medida predeterminada da distância entre esta imagem e uma imagem no banco de dados. As medidas tradicionais de distância, como a distância euclidiana, considera a semelhança entre dois pares de imagens. Em muitas situações, estas abordagens não conseguem apresentar resultados satisfatórios, principalmente em função do conhecido problema de *gap* semântico.

Segundo Pedronette *et al.* (2012) diversos estudos demonstram empiricamente e teoricamente que uma medida aprendida pode significativamente melhorar a classificação (*ranking*), *clustering* e recuperação de imagens. Cita o autor que em particular, os sistemas de busca por conteúdo, tem pesquisado sobre a melhoria das medidas de distâncias. Resultados promissores foram obtidos considerando-se várias abordagens e técnicas, uma das abordagens é chamada de re-classificação de imagem (*re-ranking*).

Uma tarefa de reclassificação (*re-ranking*) consiste em tomar um *ranking* inicial como entrada e, em seguida, considerando-se algumas informações adicionais, recalculando este *ranking* a fim de melhorar a eficácia dos sistemas de busca. Vários sistemas de recuperação de informação têm explorado diferentes tipos de informação, tais como as relações entre os itens e as preferências do usuário. Em busca de texto, tarefas de reclassificação se baseiam no *feedback* de uma pesquisa de texto inicial que é realizada ao longo do texto e os documentos mais bem classificados são considerados verdadeiros, pseudo-positivos. Termos adicionais descobertos nos documentos pseudo-positivos são então adicionados à consulta e a busca é executada novamente.

No cenário CBIR, o conceito de “*contextual information*”, informação contextual, tem sido usado para a concepção de métodos que exploram informações sobre as relações entre as imagens em tarefas de reclassificação. A noção de contexto refere-se aos vizinhos mais próximos de uma consulta, com uma medida de similaridade construída sobre a lista de *ranking*. Um método de agrupamento é usado para representar a informação contextual e em um método de pós-processamento considera-se a semelhança entre a lista do *ranking* para a caracterização de informação contextual. A principal motivação do algoritmo baseia-se na suposição de que as informações contextuais codificados na semelhança entre as demais imagens da lista de *ranking* podem fornecer recursos para a melhoria da eficácia dos descritores de busca por conteúdo. O algoritmo calcula a forma de similaridade de um par de imagens no contexto de outras imagens, em oposição ao considerar apenas as relações de imagens pares (PEDRONETTE *et al.*, 2012).

Torres *et al.* (2009) propuseram um método que explora as pontuações de similaridade de forma, este método usa um algoritmo de agrupamento não supervisionado, com o objetivo de capturar a estrutura de variedade das relações entre as imagens. Eles propõem um “*Distance Optimization Algorithm – DOA*”, algoritmo de otimização de distância, com o objetivo de agrupar formas, tendo em conta a semelhança entre listas de *ranking*. As distâncias entre as formas são atualizadas com base em *clusters* criados com o objetivo de melhorar a eficácia de recuperação.

Pedronette *et al.* (2012) explica que os métodos de pós-processamento propostos em Torres *et al.* (2009), Yang *et al.* (2008) e Yang *et al.* (2009) podem ser adaptados ou estendidos para serem usados em outras tarefas, como a agregação de classificação e recuperação multimídia.

Diferentes descritores CBIR produzem *rankings* e diferentes descritores podem fornecer diferentes, mas complementares, informações sobre as imagens. Para eles basicamente, as abordagens de agregação de classificação visam combinar diferentes classificações com o objetivo de obter uma classificação mais precisa. Uma das abordagens utiliza agregação de patente em novas aplicações, tais como filtragem de documentos, *spam*, detecção de página, meta-busca, busca por palavra associada, pesquisa múltipla e pesquisa por similaridade.

Abordagens não-supervisionadas considerando contagens de recuperação ou posições *ranking* têm sido amplamente utilizadas em tarefas de recuperação de informação. No domínio CBIR, o uso de técnicas de aprendizado de máquina têm sido empregadas com o objetivo de classificar com precisão o retorno de imagens. Uma abordagem comum é a utilização de métodos de aprendizagem para combinar informações provenientes de diferentes descritores. Estas abordagens incluem técnicas como “*Support Vector Machines*”, “*Genetic Programming*”, entre outros. Técnicas de classificação de agregação também podem ser exploradas para a recuperação multimídia, em especial em objetos que são compostos por diferentes mídias, como texto e imagem. Ocorreu um aumento exponencial deste tipo de conteúdos digitais nos últimos anos e as técnicas desenvolvidas se baseavam no contexto de três categorias diferentes, “*early fusion*”, “*late fusion*” e “*transmedia fusion*” ou seja fusão precoce, fusão tardia, e fusão transmídia, respectivamente.

## **2.5.2 Contextual Spaces Algorithm**

O *Contextual Spaces algorithm* redefine o relacionamento entre objetos de uma coleção respondendo a pergunta: “Que informação os vizinhos mais próximos de um objeto podem prover a respeito de outros objetos da coleção?”. Então, não apenas a similaridade entre dois objetos vizinhos é considerada, mas também a informação contextual contida em toda a vizinhança (PEDRONETTE *et al.*, 2012).

Um espaço contextual é um espaço bidimensional onde todos os objetos da coleção são representados de acordo com sua distância para um objeto arbitrário e cada um de seus  $k$  vizinhos mais próximos (do inglês *k-Nearest Neighbours* ou kNN). Para a tarefa de segmentação, cada objeto é considerado como uma região pré-segmentada de uma imagem.

Sendo  $r_i$  e  $r_l$  regiões pré-segmentadas de uma imagem  $I$  e  $\rho(r_i, r_l)$  a distância entre  $r_i$  e  $r_l$ , pode-se calcular os  $k$  vizinhos mais próximos da região  $r_i$  baseando-se na distância  $\rho$ . Para cada vizinho  $k$  de  $r_i$ , um espaço contextual é construído visando redefinir as distâncias de  $r_i$  para as outras regiões da imagem.

Novas distâncias  $\rho_{t+1}$  são calculadas agregando informação do espaço contextual de  $k$  num único espaço (onde  $t$  indica a iteração sendo processada). As novas distâncias são definidas de acordo com duas dimensões:  $d_i$  e  $d_j$ . Dadas duas regiões  $r_i, r_l$ , a dimensão  $d_i$  representa a informação de distância de  $r_i$  para  $r_l$ . A dimensão  $d_j$  representa a informação de distância dos vizinhos  $r_j \in kNN(r_i)$  para  $r_l$ .

A dimensão  $d_i$  é calculada como demonstrado na Equação 2.9:

$$d_i = \frac{\rho_t(r_i, r_l)}{k}, \quad (2.9)$$

A dimensão  $d_j$  considera um peso diferente para cada vizinho, onde vizinhos em posições mais altas do *ranking* apresentam maiores pesos. Neste trabalho foi utilizado o algoritmo na abordagem kNN Mútuo (PEDRONETTE *et al.*, 2012), que considera não apenas a informação dos vizinhos mais próximos mas também a mútua referência entre as regiões e o *ranking* de seus vizinhos. A dimensão  $d_j$  é calculada conforme a Equação 2.10:

$$d_j = \frac{\sum_{j=1}^k \rho_t(r_j, r_l) \times (k - j)}{\frac{k \times (k - 1)}{2}}, \quad (2.10)$$

Finalmente, a próxima iteração de distância  $\rho_{t+1}$  é calculada como se segue:

$$\rho_{t+1}(r_i, r_l) = \sqrt{d_i^2 + d_j^2}. \quad (2.11)$$

O processo é repetido iterativamente ao longo de  $t$  iterações, visando calcular uma nova e mais eficaz distância  $\hat{\rho}(r_i, r_l)$  entre as regiões  $r_i$  e  $r_l$ .

# Capítulo 3

## Segmentação de imagens usando corte normalizado

Neste capítulo será abordado o método de segmentação proposto neste trabalho, ilustrando através de um diagrama de blocos, os passos necessários para atingir os resultados esperados. Uma breve discussão sobre trabalhos relacionados também será exposta.

### 3.1 Trabalhos relacionados

Observa-se na literatura acadêmica recente que o método proposto nesta pesquisa assemelha-se a pesquisas de outros autores. Isto se deve à versatilidade do Corte Normalizado, que proporciona um vasto campo para aplicação de métodos híbridos de segmentação, uma vez que o grafo de similaridade pode ser obtido através de diferentes técnicas e particionado da mesma forma pelo NCut.

Uma abordagem utilizada como exemplo por Shi e Malik (1997) utiliza uma estrutura onde cada *pixel* é representado por um vértice no grafo e a adjacência/similaridade é estabelecida por uma função de brilho e posição espacial de acordo com a distância entre os *pixels*. Esta abordagem, neste trabalho denominada de *pixel grid*, exige muito esforço computacional e pode tornar-se inviável para aplicações de visão computacional, uma vez que o desempenho é diretamente relacionado com o tamanho da imagem (quantidade de *pixels*).

Desta forma, é comum encontrar métodos que fazem um agrupamento inicial dos *pixels* através de alguma técnica específica e posteriormente agrupam os vértices utilizando o NCut.

Os trabalhos publicados por De Bock *et al.* (2004), Monteiro e Campilho (2008) e Fabijanska (2012) são exemplos onde a Transformada de *Watershed* é aplicada como um passo inicial de pré-segmentação com utilização do NCut para obter a segmentação final, atestando o ganho computacional em relação ao método *pixel grid*.

A diferença do método proposto neste trabalho é a utilização de duas técnicas adicionais para modelar a imagem em grafo: o *Watershed Hierárquico* e o *Contextual Spaces Algorithm*.

Em (COSTA, 2013) o desempenho de quatro modelos de representação de imagem-grafo foi comparado através de métricas do *benchmark* da Berkeley BSDS500 (MARTIN *et al.*, 2001):

- Modelo *pixel grid* (SHI E MALIK, 1997);
- Modelo Multiescala (COUR *et al.*, 2005);
- Modelo Árvore dos componentes;
- Modelo *Quadtree*.

A pesquisa de Costa (2013) foi desenvolvida paralelamente ao método proposto neste trabalho, e a utilização do *benchmark* é implementada da mesma forma, o que permite uma comparação direta entre os modelos, no que diz respeito à qualidade da segmentação.

## 3.2 Método de segmentação proposto

Conforme anunciado nos objetivos propostos por este projeto, a estratégia de segmentação pretendida envolve a combinação dos métodos abordados no Capítulo 2. Desta forma, pretende-se estabelecer um método que possa ser utilizado em diferentes tipos de imagem sem que haja necessidade de alteração dos processos envolvidos.

A sequência de operações realizadas tem cinco estágios principais: Pré-processamento, Segmentação Inicial, Criação Grafo de Similaridade, Reclassificação (*Reranking*) e Particionamento/Segmentação. O diagrama ilustrando estes estágios pode ser visto na Figura 3.1.

A seguir, cada etapa do diagrama é abordada separadamente de maneira sucinta.

### **Etapas do método de segmentação:**

- **Pré-Processamento:** É feita a obtenção da imagem, através de qualquer dispositivo eletrônico (*scanner*, máquina fotográfica, microscópio, etc.) e seu armazenamento é feito em

formato digital. Caso a imagem seja colorida, é feita sua conversão para uma imagem em Escala de Tons de Cinza. Conforme descrito no Capítulo 2, a aplicação do gradiente morfológico é necessária para evidenciar os contornos dos objetos da imagem. Caso seja necessário, neste passo também podem ser aplicados outros operadores na imagem (em caso de ruídos, por exemplo). Neste trabalho não foi aplicado nenhum operador adicional;

- **Segmentação Inicial:** Nesta etapa, é aplicada a segmentação inicial da imagem, obtendo as regiões de *watershed* através da aplicação da IFT. A escolha das sementes é feita de maneira automática, rotulando todos os mínimos regionais presentes na imagem e marcando-os como sementes. Após extração do *watershed*, é feita a construção de um espaço de escalas (*Watershed* hierárquico), reduzindo o número de regiões a serem analisadas;
- **Criação Grafo de Similaridade:** Neste passo é feito o cálculo das métricas relacionadas a cada região do *Watershed* Hierárquico: Volume e Moda de Cinza. Uma das métricas é escolhida (ou uma combinação delas) e aplicada como peso das arestas do grafo, obedecendo a uma função que estabeleça a similaridade entre regiões adjacentes. Calculam-se então as estruturas de matrizes descritas no Capítulo 2;
- **reranking:** O *reranking* dos pesos é feita através da aplicação do *Contextual Spaces Algorithm*, visando obter uma melhor distribuição entre as similaridades das regiões;
- **Particionamento e Segmentação:** A etapa final é a aplicação do Corte Normalizado, que vai particionar os nós do grafo de similaridade em regiões (subgrafos) de acordo com a afinidade entre elas.

A Figura 3.2 exemplifica de forma mais didática o processo de segmentação proposto e descrito acima.

### 3.3 Plano de trabalho executado

Este trabalho abordou técnicas bastante consolidadas na bibliografia acadêmica: Transformada *Watershed* e o Corte Normalizado. Em contrapartida, tendo em vista o crescente número de publicações relacionadas ao Corte Normalizado, uma constante revisão bibliográfica deste tópico foi necessária. As principais etapas do projeto foram:

- **Revisão bibliográfica:** Foi realizada uma atualização da bibliografia disponível. Esta etapa foi revisitada periodicamente para acompanhar novas publicações relacionadas ao tema;

- **Obtenção de imagens:** Aquisição das imagens do *Berkeley Image Dataset* BSDS500 que foram utilizadas nos processos de segmentação e montagem do ambiente para execução do *benchmark* disponibilizado;
- **Estudo das técnicas abordadas:** Foi aprofundado o conhecimento na Transformada *Watershed*, *Watershed* Hierárquico e IFT, assim como o tópico de teoria espectral dos grafos e Corte Normalizado. Adicionalmente, foi introduzido o estudo da técnica de espaços contextuais e *reranking*.
- **Implementação do Método:** O método de segmentação foi implementado quase que integralmente em *Matlab*. No passo de Pré-Segmentação, a leitura da imagem, conversão para escala de tons de cinza e cálculo do gradiente morfológico foram implementados através de funções do pacote de processamento gráfico embutido no aplicativo *Matlab*. O algoritmo da IFT, a simplificação com *watershed* hierárquico, o cálculo das métricas e a construção do grafo de similaridade foram implementados neste trabalho utilizando a linguagem de programação do *Matlab*, assim como o particionamento via NCut, que foi adaptado da implementação disponibilizada por (SHI E MALIK, 1997). Já o método de *reranking* foi desenvolvido em linguagem *Java* em (PEDRONETTE *et al.*, 2012), adaptado e reutilizado neste trabalho.
- **Aplicação do Método:** O método foi aplicado integralmente para as 500 imagens disponíveis no BSDS500. Para isso foi necessária uma interface via arquivo de texto para que a Matriz de Similaridade obtida em Matlab fosse reprocessada no processo de *reranking* e novamente processada no Matlab para aplicação do NCut.
- **Comparação dos resultados:** Os resultados obtidos foram submetidos ao *benchmark* do BSDS500, também implementado em Matlab e posteriormente planilhados para análise;
- **Documentação dos resultados:** Após avaliação dos resultados, começou-se a escrita da dissertação para documentar e discutir o método e apresentar os resultados obtidos.

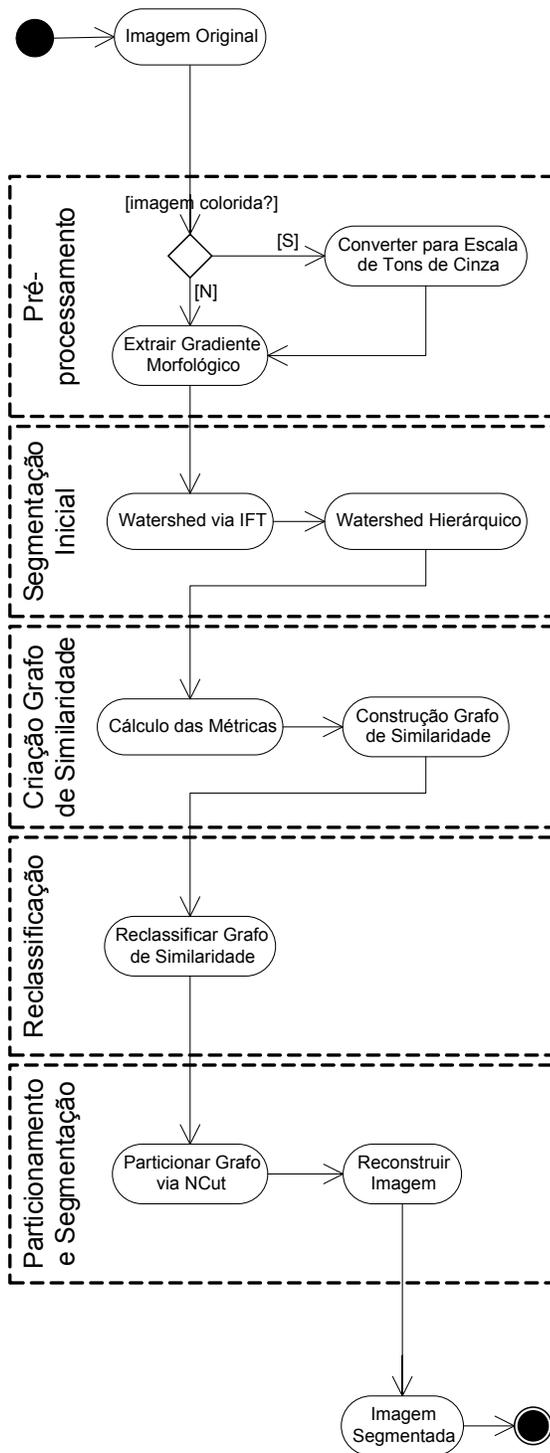


Figura 3.1: Diagrama de blocos do método de segmentação proposto.

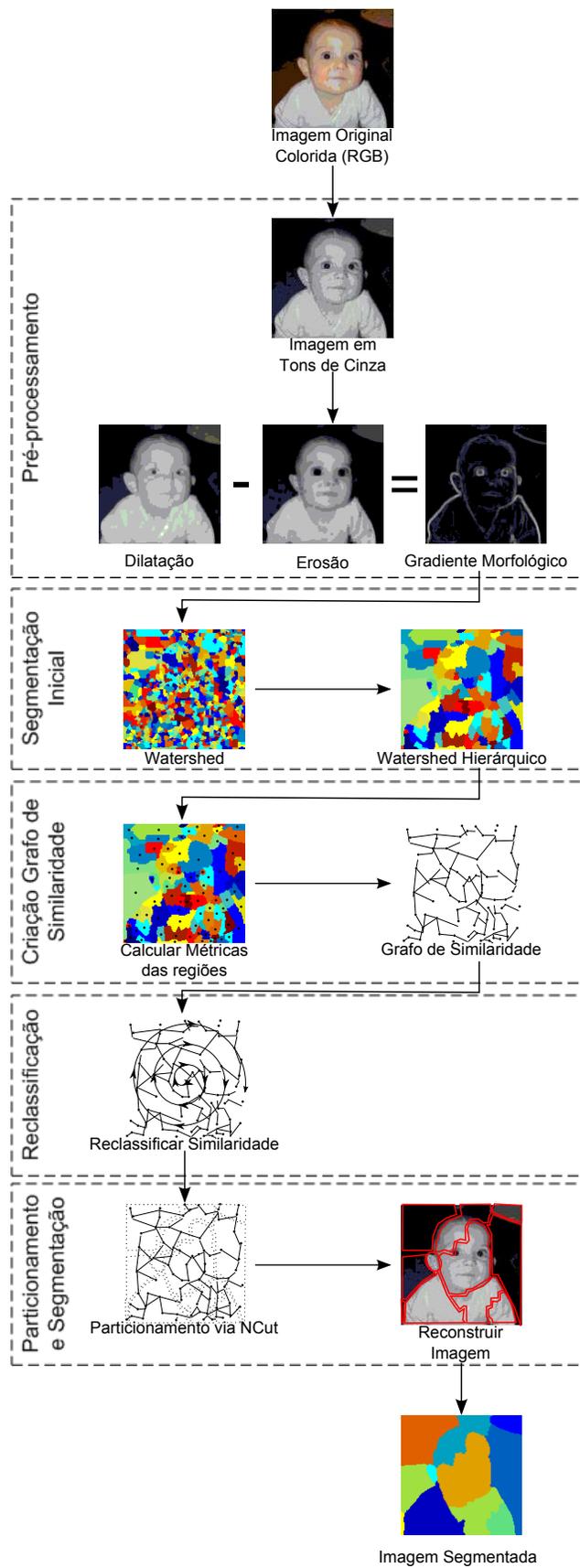


Figura 3.2: Exemplo de aplicação do método para uma imagem de teste.

# Capítulo 4

## Experimentos e resultados

Neste capítulo serão exibidos os resultados alcançados com a aplicação do método de segmentação proposto neste trabalho, partindo da configuração e parametrização de cada etapa até a construção dos modelos testados.

### 4.1 Utilização de hardware e software

Para a implementação e teste do método de segmentação, foram utilizadas as seguintes ferramentas:

- **Software:**

*Matlab 7.0.1 R14:* Para implementação do método, geração dos resultados e execução do *benchmark* de comparação;

*Netbeans IDE 7.4:* Para implementação e teste do processo de reclassificação utilizando o *Contextual Spaces Algorithm*.

- **Hardware:**

*Processador:* Core 2 Duo T8100 2.1GHz

*Memória RAM:* 3Gb

*Sistema Operacional:* Windows Vista 32bits

## 4.2 Parametrização e configuração

O diagrama de atividades descrito no Capítulo 3 expõe de forma macro os passos utilizados no método de segmentação proposto. Para sua aplicação prática, é necessário definir parâmetros e configurações para cada uma das etapas. A seguir serão detalhadas as configurações utilizadas neste trabalho:

1. **Pré-Processamento:** Conforme descrito anteriormente, a implementação das operações utilizadas nesta etapa são nativas do pacote gráfico do Matlab, seguindo os seguintes parâmetros:

*Leitura da Imagem* - Imagens coloridas são armazenadas no sistema de cores RGB e sua leitura é feita através de um comando que armazena a imagem de entrada numa estrutura de matriz com capacidade de armazenar 1 byte por pixel, o que permite uma variação de 0 a 255 para o valor de cada entrada. Neste sistema de cores uma matriz de 3 dimensões é utilizada para armazenar o valor relativo a cada canal de cor (*Red-Green-Blue*).

*Conversão para Escala de Tons de Cinza* - O método de conversão utilizado neste trabalho é o mesmo da NTSC (*National Television System Committee*) para transmissão de sinal de TV monocromática. Trata-se de uma função que converte a intensidade dos canais de cor RGB de cada pixel  $i$  através de uma média ponderada entre eles:  $Gray(i) = a \times R(i) + b \times G(i) + c \times B(i)$ , onde  $a$ ,  $b$  e  $c$  indicam as contribuições de peso de cada canal de cor. Na técnica da NTSC os valores de  $a$ ,  $b$  e  $c$  associados aos componentes R, G, e B são 0,2989, 0,5870, and 0,1140 respectivamente. Isso se deve ao fato de que para quantidades iguais de cor, o olho humano é mais sensível ao verde, seguido do vermelho e por último o azul (GONZALEZ E WOODS, 2008).

*Aplicação do Gradiente Morfológico* - As operações de dilatação e erosão são aplicadas com um elemento estruturante no formato 3x3, que após testes realizados apresentou geração de contornos mais suaves.

2. **Segmentação Inicial:** Funções implementadas em linguagem Matlab com base em trabalhos relacionados na literatura acadêmica:

*Watershed Clássico* - Aplicação do *watershed* via IFT utilizando como parâmetro vizinhança do tipo 4, pois os resultados obtidos nos testes iniciais foram melhores aos obtidos com vizinhança do tipo 8. Resultados obtidos com vizinhança do tipo 8 tiveram uma tendência a criar

regiões pouco similares entre si.

*Watershed Hierárquico* - Para testar a influência da simplificação das regiões obtidas no *watershed* clássico pelo *watershed* hierárquico, foram gerados três grupos de resultados para aplicação do método de segmentação proposto: 1000 regiões, 500 regiões e 100 regiões. Essas quantidades distintas de regiões foram utilizadas a fim de avaliar a influência que este parâmetro exerce no resultado final da segmentação. Foram avaliados também resultados obtidos via *watershed* hierárquico reduzindo o número de regiões diretamente para as quantidades que foram testadas posteriormente com o método: 10 regiões e 30 regiões.

3. **Criação do Grafo de Similaridade:** Funções implementadas em linguagem Matlab com base em trabalhos relacionados na literatura acadêmica:

*Calculo das Métricas* - Para cada região do *watershed* hierárquico foram calculadas duas métricas: Volume e Moda de Cinza. O Volume é uma característica morfológica da região extraída do *watershed* hierárquico e sua utilização se faz interessante pois por natureza esta característica é uma combinação entre a área da região e seus tons de cinza. A Moda é uma medida matemática que atribui a cada região o valor de cinza que mais se repete dentro dela, podendo evidenciar qual é sua tonalidade de cinza predominante.

*Construção do Grafo* - Cada região do *watershed* hierárquico foi considerado como um vértice no grafo. Dois vértices são considerados vizinhos e conectados por uma aresta se a distância entre eles for menor que um raio  $r$  pré-estabelecido, que neste trabalho assume o valor de 15% da maior distância possível na imagem. Este parâmetro foi escolhido pois é o limiar onde observou-se o melhor resultado na relação densidade do grafo x qualidade da segmentação. As arestas são ponderadas com a similaridade entre as regiões, através da Equação 4.1:

$$W_I(i, j) = e^{(-\frac{\|x_i - x_j\|^2}{\sigma_X} - \frac{\|I_i - I_j\|^2}{\sigma_I})} \quad (4.1)$$

onde  $I$  assume a métrica escolhida (Volume ou Moda),  $X$  é a posição do pixel central da região,  $\sigma_X = 4$  e  $\sigma_I = 0,01$ .

Adicionalmente, foi adicionado no cálculo da similaridade a Equação 4.2 proposta em (COUR *et al.*, 2005) e que utiliza o conceito de Contorno Interferente (do inglês *Intervening Contour* - IC) para medir a similaridade de regiões que estão separadas por um contorno:

$$W_C(i, j) = e^{-\frac{\max_{x \in \text{line}(i, j)} \|Edge(x)\|^2}{\sigma^C}} \quad (4.2)$$

onde  $\text{line}(i, j)$  é uma linha reta entre os pontos  $i$  e  $j$ ,  $Edge(x)$  é a força do contorno nesse ponto.

As duas equações são então combinadas na Equação 4.3 conforme segue:

$$W_M(i, j) = \sqrt{W_I(i, j) - W_C(i, j)} + \alpha W_C(i, j) \quad (4.3)$$

onde  $\alpha = 1, 5$ .

4. **Reclassificação:** Função implementada em linguagem Java (PEDRONETTE *et al.*, 2012) e utilizada através de uma interface por arquivos de texto, sendo o arquivo de entrada a Matriz de Similaridade calculada anteriormente e o arquivo de saída a Matriz de Similaridade reclassificada através do *Contextual Spaces Algorithm* na abordagem kNN mútuo. Foram utilizados como parâmetros:  $kNN = 30$ ,  $t = 20$  e  $\lambda = 30$ .
5. **Particionamento e Segmentação:** Função implementada em Matlab (SHI E MALIK, 1997) e adaptada para receber como entrada a Matriz de Similaridade obtida nos passos anteriores. Para fins de comparação com os modelos testados em (COSTA, 2013), a quantidade de regiões estabelecida para o particionamento foram 10 e 30 regiões, utilizando a abordagem *k-way*. Com o intuito de verificar a eficácia da utilização da reclassificação, foram gerados particionamentos com e sem a utilização do método. Após particionamento, o resultado da função é um vetor mapeando à qual partição cada região pertence, permitindo assim a reconstrução da imagem segmentada.

A Figura 4.1 ilustra de forma compacta os diferentes parâmetros utilizados para teste do método de segmentação, destacando em quadros mais escuros as principais atividades:

Conforme visto nas parametrizações acima, pode-se considerar que foram testados 26 modelos ao longo deste trabalho:

- Dois modelos de *watershed* hierárquico (10 e 30 regiões);
- Doze modelos combinando *watershed* hierárquico (100, 500 e 1000 regiões) com duas métricas (moda e volume) e Corte Normalizado (10 e 30 regiões);
- Doze modelos combinando *watershed* hierárquico (100, 500 e 1000 regiões) com duas métricas (moda e volume), Reclassificação e Corte Normalizado (10 e 30 regiões).

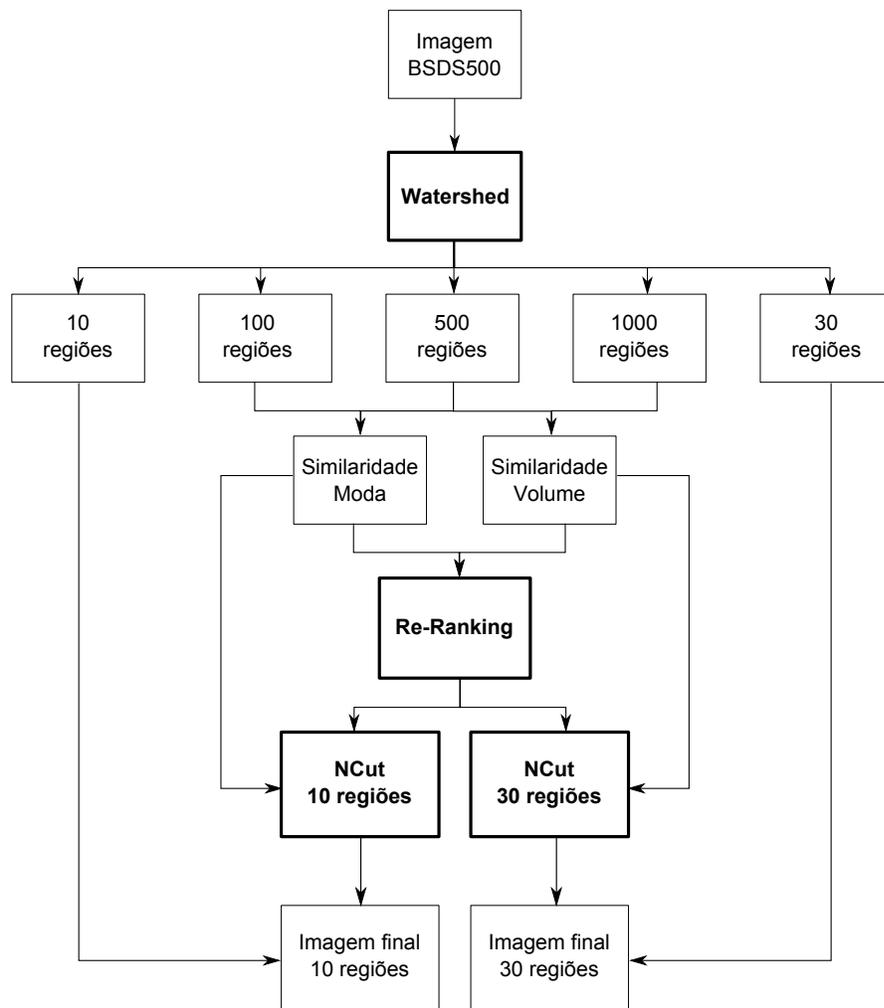


Figura 4.1: Exemplo gráfico ilustrando os modelos testados no trabalho.

### 4.3 *Benchmark* para análise dos resultados

Para avaliação dos resultados obtidos neste trabalho, foi utilizado o *benchmark* de segmentação de imagens BSDS500 da Berkeley (MARTIN *et al.*, 2001). Esta 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). 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 e foi utilizado neste trabalho para ajuste dos parâmetros;

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 foi feito neste trabalho;
3. **de teste, composto por 200 imagens:** Esse conjunto deve ser utilizado para definir a pontuação definitiva do algoritmo, não devendo ser base para ajuste de código ou parâmetros.

Para verificar a eficiência do método proposto, foram utilizados índices de avaliação disponibilizadas pelo BSDS500 para os pixels de borda dos objetos segmentados. A decisão desta avaliação pode ser representada em uma estrutura chamada matriz de confusão onde 4 categorias podem ser encontradas:

- Verdadeiros-Positivos (VP): elementos corretamente rotulados como positivos;
- Falsos-Positivos (FP): elementos incorretamente rotulados como positivos;
- Verdadeiros-Negativos (VN): elementos corretamente rotulados como negativos;
- Falsos-Negativos (FN): elementos incorretamente rotulados como negativos;

A Precisão (P) é a probabilidade de que um pixel de borda de uma região gerado pelo método ser de fato um pixel de borda. Tal métrica é calculada como:

$$P = \frac{VP}{VP + FP}, \quad (4.4)$$

Recuperação (R) é a probabilidade de que um pixel de borda de uma região verdadeiro seja encontrado e é calculado como:

$$R = \frac{VP}{VP + FN}, \quad (4.5)$$

O desempenho do método será avaliado de acordo com a medida-F, uma medida que calcula a média harmônica ponderada dos valores de Precisão e Recuperação, dado pelo cálculo:

$$F = 2 \cdot \frac{P \cdot R}{P + R}, \quad (4.6)$$

Também foram utilizadas métricas que avaliam a segmentação com base nas regiões. Uma delas é a cobertura de região (MARTIN *et al.*, 2001) definida como:

$$C(S' \rightarrow S) = \frac{1}{N} \sum_{R \in S} |R| \cdot \max_{R' \in S'} O(R, R'), \quad (4.7)$$

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) (MARTIN *et al.*, 2001), definido como:

$$PRI(S, \{G_k\}) = \frac{1}{T} \sum_{i < j} c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij}), \quad (4.8)$$

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.

## 4.4 Resultados obtidos e comparação

As combinações de modelos foram testadas com todas as imagens do BSDS500, obtendo uma grande base de resultados que serviu de suporte para análise da eficácia do método testado. Na Tabela 4.1 são demonstrados nomenclaturas dadas a cada modelo implementado neste trabalho e também demais modelos utilizados para comparação.

Os modelos WH-10/WH-30 e Pixel Grid equivalem, respectivamente, aos métodos de *Watershed* Hierárquico e Corte Normalizado aplicados isoladamente, o que permite uma comparação objetiva para validar se a combinação proposta pelo método estudado neste trabalho obteve ganho na qualidade de segmentação.

Uma das formas de medir o ganho de qualidade na segmentação é pela utilização das métricas disponibilizadas pelo BSDS500. Todas as métricas tem grande importância para esta finalidade, porém, neste trabalho, o foco voltou-se para a geração de bordas das regiões finais da segmentação. Sendo assim, a métrica de medida-F (que mede a qualidade na geração das bordas) é que norteia a análise dos resultados.

Tabela 4.1: Nomenclatura dada aos modelos testados para o método de segmentação proposto.

Nome do Modelo	Regiões <i>Watershed</i> Hierárquico	Métrica	Utiliza Reclassificação?
WHNCut-Moda-1000	1000	Moda	Não
WHNCut-Volume-1000	1000	Volume	Não
WHRNCut-Moda-1000	1000	Moda	Sim
WHRNCut-Volume-1000	1000	Volume	Sim
WHNCut-Moda-500	500	Moda	Não
WHNCut-Volume-500	500	Volume	Não
WHRNCut-Moda-500	500	Moda	Sim
WHRNCut-Volume-500	500	Volume	Sim
WHNCut-Moda-100	100	Moda	Não
WHNCut-Volume-100	100	Volume	Não
WHRNCut-Moda-100	100	Moda	Sim
WHRNCut-Volume-100	100	Volume	Sim
WH-10	10	NA	NA
WH-30	30	NA	NA
Pixel Grid (SHI E MALIK, 1997)	NA	NA	NA
Multiescala (COUR <i>et al.</i> , 2005)	NA	NA	NA
Árvore dos Componentes (COSTA, 2013)	NA	NA	NA
Quadtree (COSTA, 2013)	NA	NA	NA

## 4.5 Resultados para o método proposto

Para avaliar a qualidade de segmentação produzida pelos modelos testados, na Tabela 4.2 os resultados deste trabalho são comparados com resultados obtidos anteriormente em (COSTA, 2013) em dois cenários: Segmentação final em 10 e 30 regiões. A ordenação da tabela é decrescente pela nota da métrica medida-F para o pacote de teste, uma vez que este é o pacote de imagens do BSDS500 utilizado para validação final dos resultados. As melhores notas alcançadas em cada métrica também são destacadas em negrito.

Observa-se que os resultados obtidos com a aplicação do método proposto não obteve as maiores notas se comparado com modelos testados em (COSTA, 2013), porém manteve-se próximo dos melhores resultados em ambos os cenários testados.

Também pode-se observar que o método proposto obteve maior pontuação em grande parte dos modelos testados para ambos cenários se comparado com os modelos WH-10/WH-30 e Pixel Grid, ratificando que a combinação das técnicas alcança notas superiores do que a utilização das mesmas de forma isolada.

Tabela 4.2: Quadro comparativo dos modelos para os conjuntos de Validação e Teste (Segmentação final em 10 e 30 regiões). Ordenação decrescente pela métrica medida-F do pacote de teste e melhores notas 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	Árvore dos Componentes	<b>0,4997</b>	<b>0,4499</b>	0,6522	<b>0,5053</b>	0,4255	0,6377
	WHNCut-Volume-500	0,4729	0,3796	0,7259	0,4810	0,4015	0,7445
	Multiescala	0,4756	0,3920	<b>0,7415</b>	0,4805	0,4128	<b>0,7649</b>
	WHNCut-Moda-500	0,4737	0,3839	0,7280	0,4804	0,4013	0,7443
	WHNCut-Volume-1000	0,4843	0,4011	0,7334	0,4783	0,4085	0,7487
	WHNCut-Moda-1000	0,4852	0,4010	0,7339	0,4777	0,4083	0,7486
	WHNCut-Volume-100	0,4632	0,3490	0,7101	0,4735	0,3564	0,7293
	WHNCut-Moda-100	0,4626	0,3479	0,7095	0,4734	0,3552	0,7281
	WH-10	0,4591	0,4181	0,7167	0,4680	<b>0,4376</b>	0,7417
	Quadtree	0,4476	0,3841	0,7273	0,4566	0,3897	0,7469
	Pixel Grid	0,4610	0,3571	0,7236	0,4538	0,3629	0,7430
30 regiões	Multiescala	<b>0,5805</b>	0,3029	0,7263	<b>0,5819</b>	0,3218	0,7501
	WHNCut-Moda-1000	0,5547	0,3332	0,7274	0,5634	0,3555	0,7528
	WHNCut-Volume-1000	0,5547	0,3344	0,7279	0,5629	0,3545	0,7525
	WHNCut-Moda-500	0,5548	0,3360	0,7291	0,5602	0,3434	0,7496
	WHNCut-Volume-500	0,5540	0,3335	0,7287	0,5591	0,3419	0,7495
	Quadtree	0,5584	0,2986	0,7224	0,5584	0,3080	0,7448
	WHNCut-Moda-100	0,5391	0,3179	0,7235	0,5577	0,3325	0,7470
	Pixel Grid	0,5556	0,2455	0,7141	0,5572	0,2541	0,7356
	WHNCut-Volume-100	0,5395	0,3198	0,7235	0,5569	0,3312	0,7466
	WH-30	0,5411	0,3624	<b>0,7324</b>	0,5492	0,3774	<b>0,7558</b>
	Árvore dos Componentes	0,5347	<b>0,4651</b>	0,6832	0,5276	<b>0,4439</b>	0,6661

Para ilustrar a qualidade de segmentação obtida pelo método proposto nos dois cenários testados, a Figura 4.2 exibe as 5 imagens com melhor pontuação para o modelo WHNCUT-VOLUME-500 (melhor modelo testado com 10 regiões) e a Figura 4.3 as 5 imagens com melhor pontuação para o modelo WHNCUT-MODA-1000 (melhor modelo testado com 30 regiões).

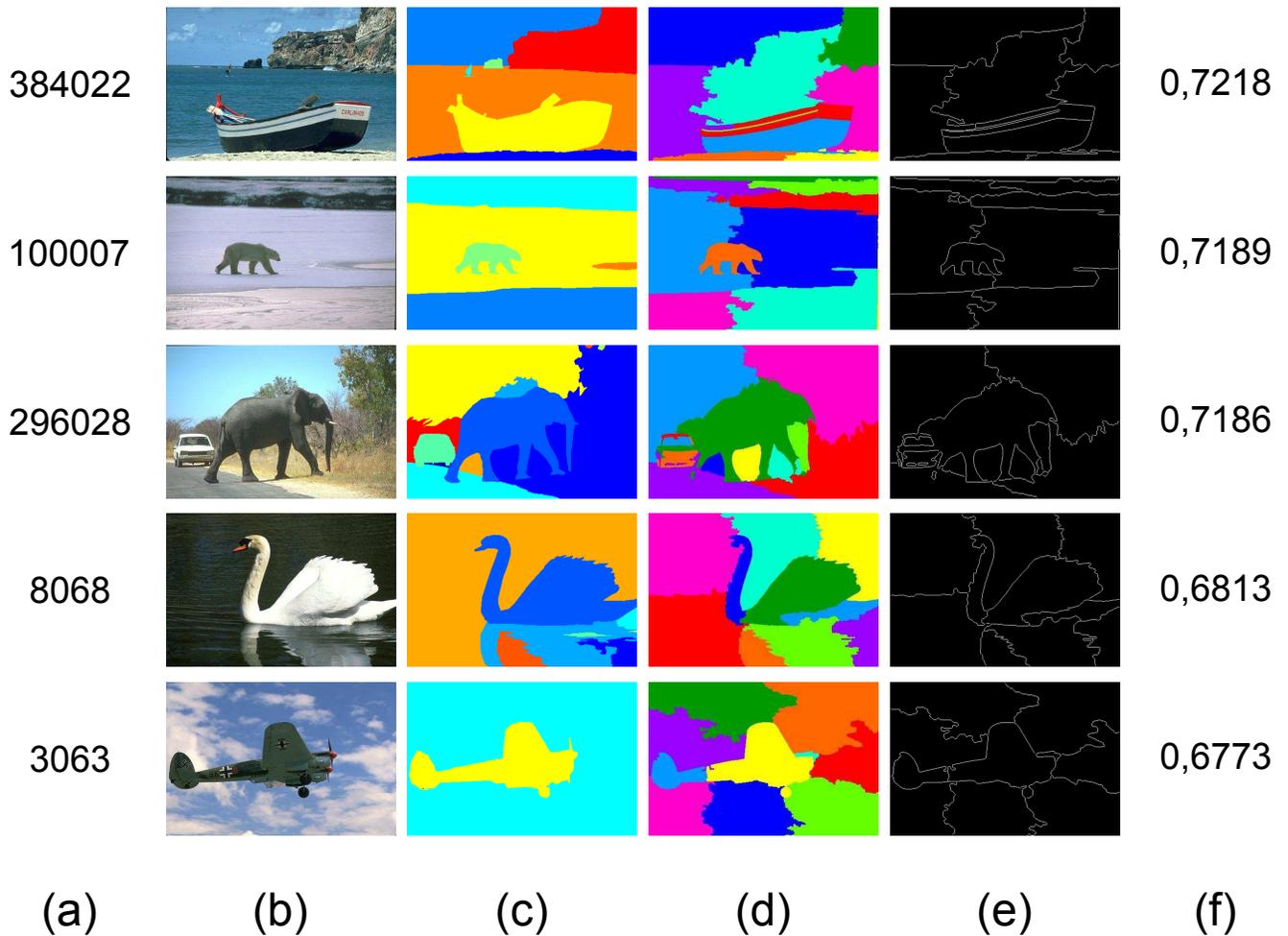


Figura 4.2: Imagens com maior nota na métrica medida-F para o experimento WHNCut-Volume-500: (a) Código da imagem; (b) Imagem Original; (c) *Ground truth*(1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Nota para medida-F

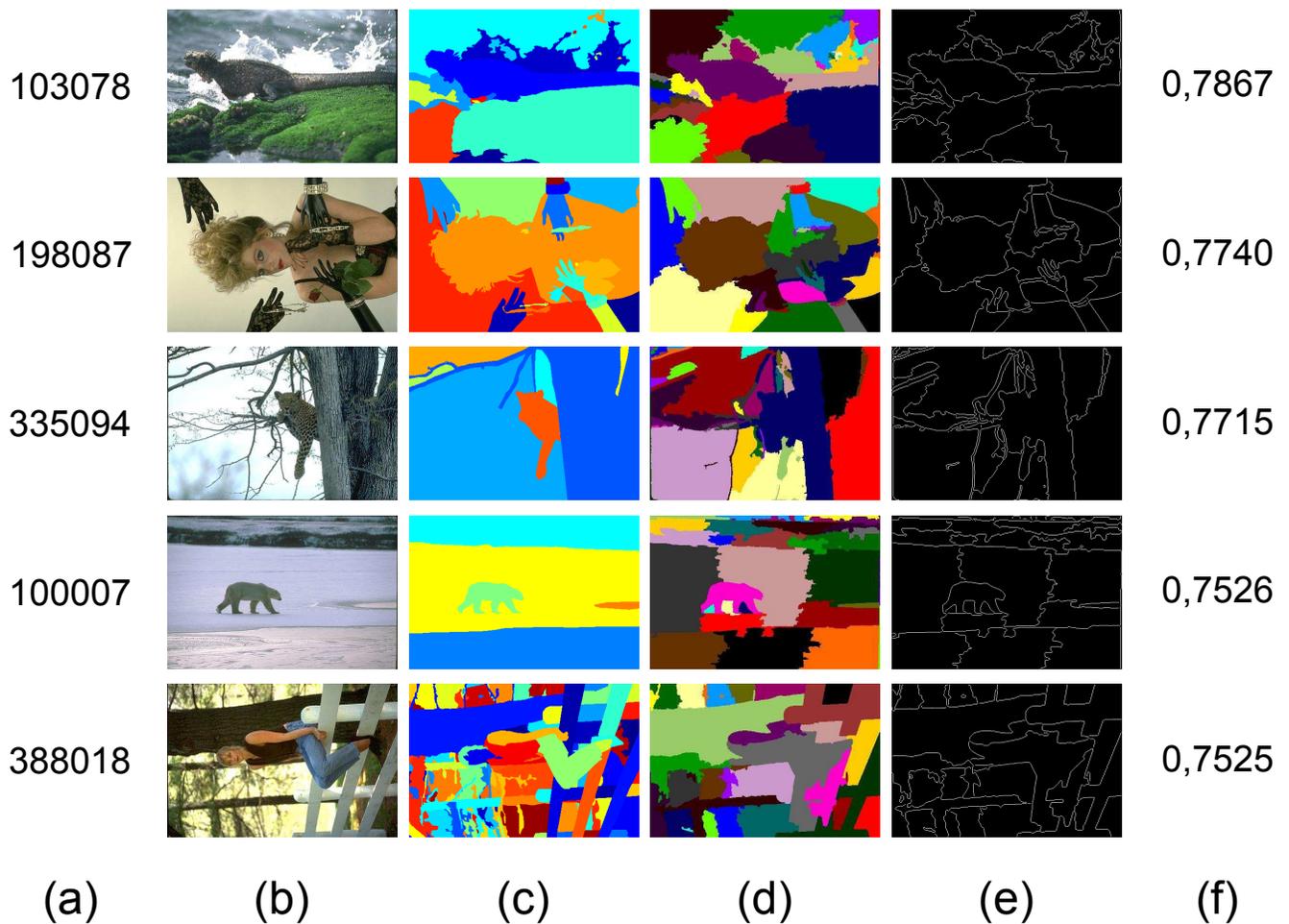


Figura 4.3: Imagens com maior nota na métrica medida-F para o experimento WHNCut-Moda-1000: (a) Código da imagem; (b) Imagem Original; (c) *Ground truth*(1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Nota para medida-F

## 4.6 Resultados com utilização da reclassificação

Para verificar a influência que a Reclassificação (*Reranking*) da Matriz de Similaridade gera na qualidade da segmentação final, a Figura 4.4 e a Figura 4.5 mostram a comparação da métrica medida-F entre os resultados obtidos com o método proposto e com a utilização da técnica de *reranking* na pasta de teste do BSDS500. No gráfico, verifica-se que os modelos que obtiveram maiores índices de melhora com a utilização do *reranking* foram WHNCUT-MODA-1000 e WHNCUT-VOLUME-1000, chegando a aproximar-se de 50% de imagens com maior nota após *reranking* para o cenário de segmentação final em 10 regiões.

Para as medidas de revocação e cobertura de região a quantidade de imagens que obtiveram melhora não foi tão significativa, mas ainda assim houve ganho em algumas situações.

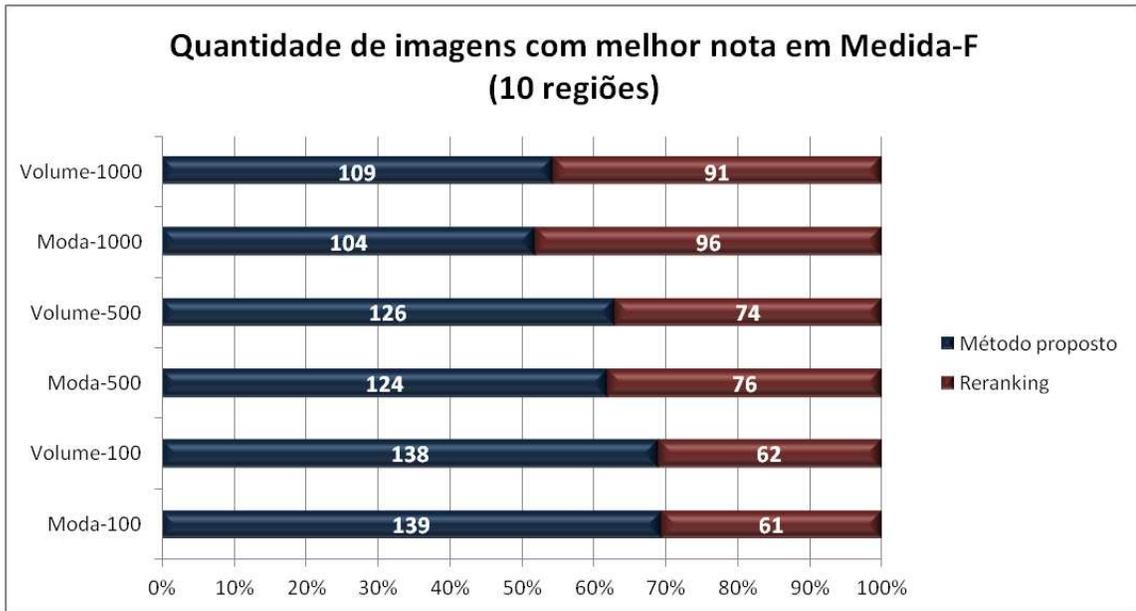


Figura 4.4: Gráfico comparativo entre os resultados para o método proposto e com a utilização de *reranking* (segmentação final em 10 regiões).

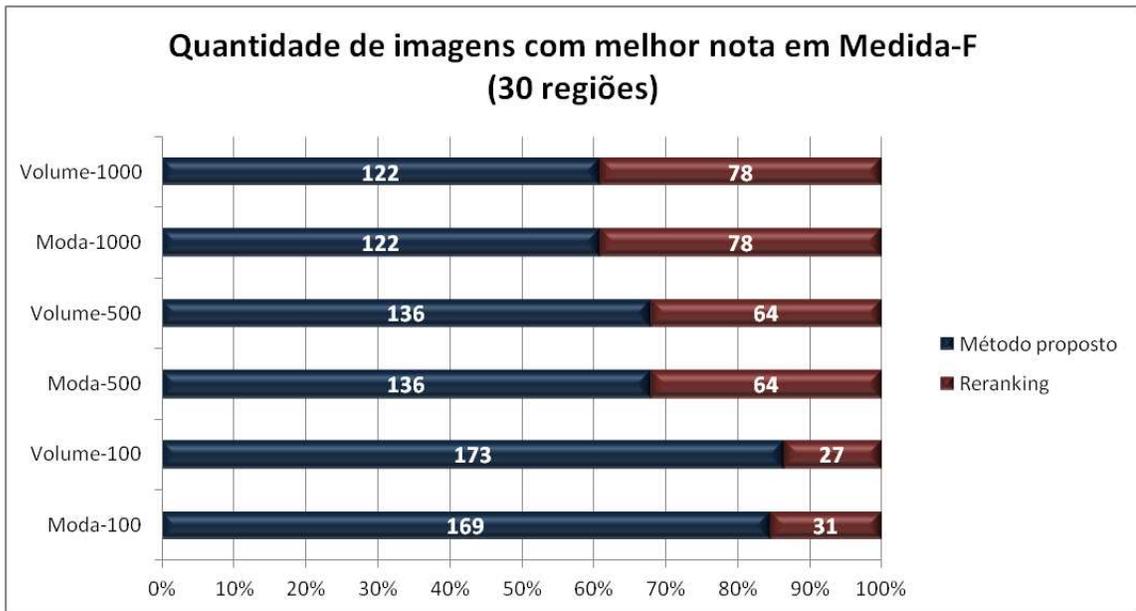


Figura 4.5: Gráfico comparativo entre os resultados para o método proposto e com a utilização de *reranking* (segmentação final em 30 regiões).

Nota-se nos gráficos mostrados, que a métrica para cálculo da Matriz de Similaridade não variou o resultado obtido após *reranking* de forma significativa. Em contrapartida, a quantidade de regiões utilizadas no *watershed* hierárquico exerceu forte influência na quantidade de imagens com aumento de qualidade na segmentação.

Para ilustrar a qualidade de segmentação obtida pela utilização da técnica de *reranking*, nos dois cenários testados, a Figura 4.6 e a Figura 4.7 exibem as 5 imagens com melhor pontuação para o modelo WHRNCUT-MODA-1000, que apresentou maior índice de evolução para as imagens processadas.

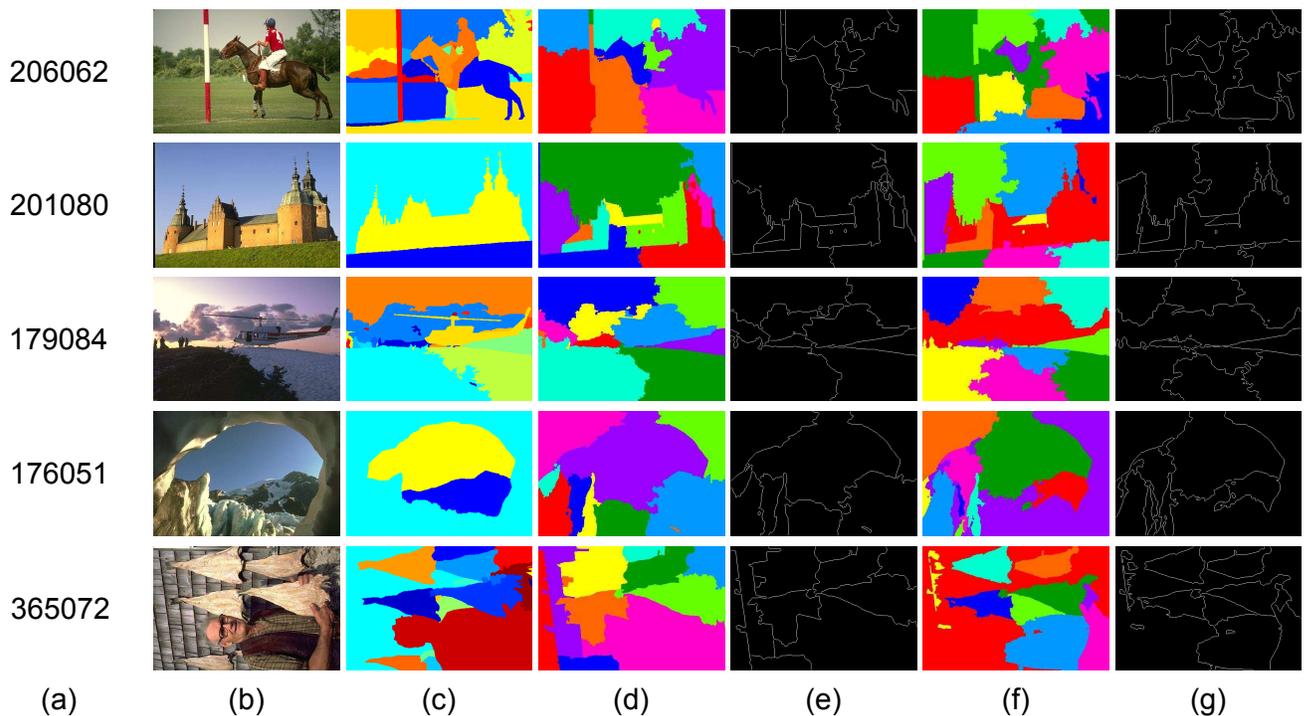


Figura 4.6: Imagens com melhor resultado após *reranking* para pacote de teste: (a) Código da imagem; (b) Imagem original; (c) *Ground truth*(1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Segmentação utilizando *reranking*; (g) Bordas de (f)

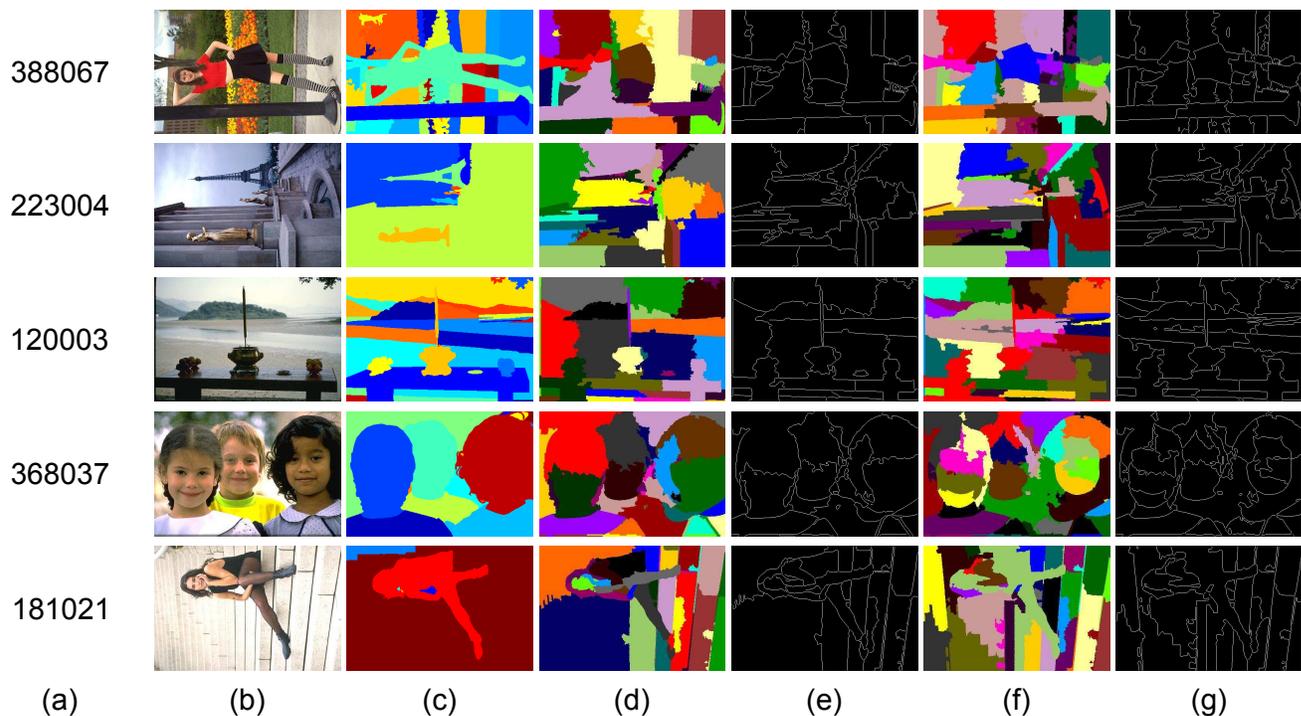


Figura 4.7: Imagens com melhor resultado após *reranking* para pacote de teste: (a) Código da imagem; (b) Imagem original; (c) *Ground truth*(1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Segmentação utilizando *reranking*; (g) Bordas de (f)

Outro indício de que a técnica de *reranking* pode obter resultados ainda mais significativos é que a maior nota obtida por uma imagem nas métricas medida-F e cobertura de região são de imagens processadas com utilização do *reranking*, são elas 55067 e 106020, respectivamente, ilustradas na Figura 4.8.

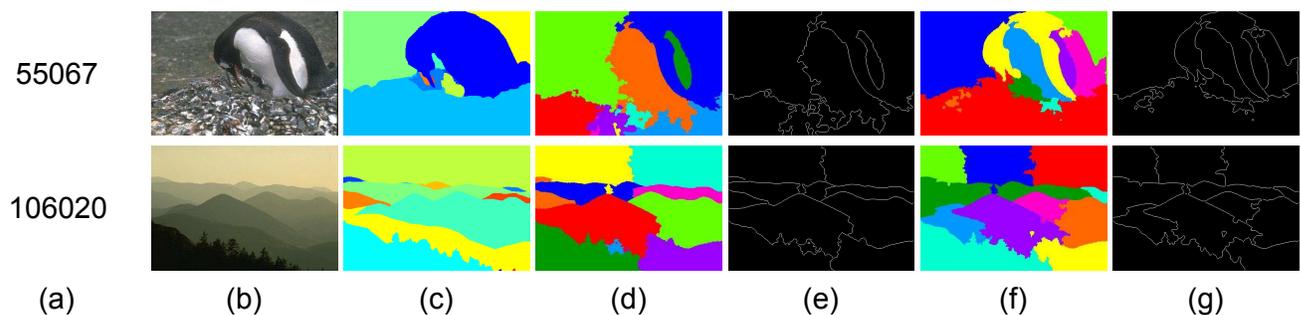


Figura 4.8: Imagens com melhor resultado após *reranking* para pacote de treinamento: (a) Código da imagem; (b) Imagem original; (c) *Ground truth*(1) do BSDS500; (d) Segmentação final pelo método proposto; (e) Bordas de (d); (f) Segmentação utilizando *reranking*; (g) Bordas de (f)

## 4.7 Resultados de tempo de processamento

Para avaliar a quantidade de tempo gasto em cada experimento processado, a Tabela 4.3 exibe os resultados em segundos para os modelos processados com segmentação final em 30 regiões.

A tabela tem o intuito de mostrar uma visão superficial do esforço gasto por cada modelo, não tendo o propósito de se fazer uma comparação objetiva ou análise de *performance* dos mesmos, uma vez que não foi utilizado *hardware* e *software* adequados para esta finalidade e cada experimento pode ter sofrido leves alterações no resultado final decorrentes da falta de rigidez no controle das execuções.

Tabela 4.3: Quadro com tempo médio de execução (em segundos) de cada etapa do método para segmentação final em 30 regiões. Ordenação decrescente pelo campo Total.

<b>Modelo</b>	<b><i>Watershed</i> Clássico</b>	<b><i>Watershed</i> Hierár- quico</b>	<b>Matriz de Similari- dade</b>	<b><i>Reranking</i></b>	<b>NCut</b>	<b>Total</b>
WHRNCut-Moda-1000	248,18	10,95	438,68	330,15	2,87	1030,83
WHNCut-Moda-1000	248,18	10,95	438,68	-	0,32	698,13
WHRNCut-Moda-500	248,18	11,50	120,04	29,79	0,60	410,11
WHRNCut-Volume-1000	248,18	10,95	19,02	116,74	2,70	397,59
WHNCut-Moda-500	248,18	11,50	120,04	-	0,14	379,86
WHRNCut-Volume-500	248,18	11,50	5,86	25,72	0,53	291,79
WHNCut-Volume-1000	248,18	10,95	19,02	-	0,38	278,53
WHRNCut-Moda-100	248,18	12,60	7,38	0,78	0,06	269,00
WHNCut-Moda-100	248,18	12,60	7,38	-	0,06	268,22
WHNCut-Volume-500	248,18	11,50	5,86	-	0,11	265,65
WHRNCut-Volume-100	248,18	12,60	2,67	1,45	0,06	264,96
WHNCut-Volume-100	248,18	12,60	2,67	-	0,06	263,51

# Capítulo 5

## Conclusões

A implementação do método proposto foi concluída com sucesso e o objetivo inicial foi atingido, uma vez que a combinação das técnicas gerou resultados superiores em sua grande maioria se comparado com os obtidos pela utilização isolada das técnicas abordadas, ratificando que é viável a utilização deste método.

Os resultados obtidos neste trabalho, pela aplicação do *benchmark* de segmentação BSDS500, demonstraram que o método de segmentação proposto pode obter segmentações com boas notas nas métricas utilizadas (Precisão, Recuperação, Medida-F, Cobertura de Região e PRI).

Apesar de a nota geral não ter superado modelos testados anteriormente, resultados para imagens específicas mostram que é possível, através ajustes nos parâmetros do método, aumentar as notas em todas as métricas e conseqüentemente refletir em segmentações que extraem com mais exatidão o conteúdo semântico das imagens.

O modelo é robusto pois independe do tamanho e tipo das imagens ao qual é submetido. Conforme mostrado nos resultados dos experimentos realizados, a pontuação do método manteve-se estável ao ser aplicado em imagens de diversos tipos, indicando que o mesmo pode atuar como agente de solução para problemas em cenários diversos.

Conclui-se que a fase de extração de métricas e construção do grafo de similaridade são de extrema importância para a qualidade final da segmentação, uma vez que a métrica escolhida está diretamente ligada com a homogeneidade no agrupamento dos pixels e a estrutura da grafo de similaridade terá grande influência no particionamento via Corte Normalizado.

A aplicação do método de *reranking* se mostrou eficaz em algumas situações. Além dos resultados exibidos no capítulo anterior, notou-se que o método conseguiu melhorar 101 dos 200

resultados obtidos para a pasta de treinamento no que diz respeito a métrica de Precisão.

Isto encoraja um aprofundamento em sua utilização e uma melhor escolha dos parâmetros de execução, uma vez que a parametrização desta etapa reaproveitou valores utilizados em (PEDRONETTE *et al.*, 2012), onde a finalidade da utilização da técnica era divergente do que está sendo abordado nesta pesquisa.

Desta forma, é seguro afirmar que a qualidade de segmentação nos resultados com a utilização do *reranking* tem ainda um grande potencial a ser explorado.

Alguns resultados alcançados durante o período desta pesquisa podem ser vistos nos trabalhos publicados e disponíveis no apêndice desta dissertação.

## 5.1 Trabalhos futuros

Diversos pontos observados no método proposto podem sofrer alterações para verificar uma melhora na qualidade final de segmentação de imagens. O método é robusto e maleável, sendo que cada um dos passos pode receber novas perspectivas para sua aplicação:

1. **Pré-Processamento:** O método de conversão da imagem colorida para tons de cinza pode utilizar outras abordagens, como *Average* (GONZALEZ E WOODS, 2008), *NECG / SNECG* (LIM E ISA, 2011) ou ainda não haver conversão da imagem. Outros tipos de gradiente também podem ser utilizados: Gradiente Matemático, Gradiente Externo, Gradiente Interno.
2. **Segmentação Inicial:** Este trabalho focou na utilização do *watershed*, mas outro tipo de abordagem para agrupamento dos pixels também poderia ser utilizado em seu lugar;
3. **Criação Grafo de Similaridade:** As métricas utilizadas, a fórmula para calcular o peso das arestas e o tipo de estrutura do grafo tem um vasto número de opções existentes. A simples variação de uma destas características pode gerar resultados totalmente diferentes;
4. **Reclassificação:** A parametrização desta etapa ainda pode ser aprofundada para gerar melhores resultados e ainda há muito campo para adaptação deste método para a tarefa de segmentação de imagens;
5. **Particionamento e Segmentação:** Neste trabalho utilizamos quantidades de partições pré-determinadas como entrada do Corte Normalizado, porém esta escolha pode ter um método automático ou ainda assistido. Pode-se também utilizar a abordagem *2-way*, para verificar se as partições geradas culminam em segmentações de mais qualidade.

# 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.
- AUDIGIER, Romaric. **Segmentação e Visualização Tridimensional Interativa de Imagens de Ressonância Magnética**. 2004. Dissertação de mestrado. Universidade Estadual de Campinas, Campinas, São Paulo, Brasil.
- BEUCHER, S. The watershed transformation applied to image segmentation. In **The Pfefferkorn Conference on Signal Image Processing in Microscopy and Microanalysis**, pp. 299–314. 1991.
- CARVALHO, M.A.G. **Análise hierárquica de imagens através da árvore dos lagos críticos**. 2004. Tese de doutorado. Universidade Estadual de Campinas, Campinas, São Paulo, Brasil.
- COSTA, André Luís da. **Corte Normalizado em Grafos: um Algoritmo Aglomerativo para Segmentação de Imagens de Colônias de Bactérias**. 2013. Dissertação de mestrado. Universidade Estadual de Campinas, Campinas, São Paulo, Brasil.
- 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**, pp. 1124–1131. 2005.
- CVETKOVIĆ, D.; ROWLINSON, P.; SIMIĆ, S. e SOCIETY, L.M. **An Introduction to the Theory of Graph Spectra**. London Mathematical Society Student Texts. Cambridge University Press, 2010. ISBN 9780521134088.
- DE BOCK, J.; DE SMET, P. e PHILIPS, W. Watersheds and normalized cuts as basic tools for perceptual grouping. In **Proc. of ProRISC 2004**, pp. 238–245. 2004.
- FABIJANSKA, A. Normalized cuts and watersheds for image segmentation. In **Image Processing (IPR 2012), IET Conference on**, pp. 1–6. 2012.
- FALCÃO, A.X.; STOLFI, J. e LOTUFO, R.A. The image foresting transform: theory, algorithms and applications. In **IEEE Transactions on Pattern Analysis and Machine Intelligence**, pp. 19–29. 2004.

- GONDRAN, M. e MINOUX, M. **Graphs and algorithms**. Discreet Mathematics Series: 1-484. John Wiley & Sons Canada, Limited, 1984. ISBN 9780471103745.
- GONZALEZ, R. e WOODS, R. **Digital Image Processing**. Pearson/Prentice Hall, 2008. ISBN 9780131687288.
- KÖRBES, André. **Análise de Algoritmos da Transformada Watershed**. 2010. Dissertação de mestrado. Universidade Estadual de Campinas, Campinas, São Paulo, Brasil.
- LIM, W.H. e ISA, N.A.M. Color to grayscale conversion based on neighborhood pixels effect approach for digital image. **International Conference on Electrical and Electronics Engineering**, pp. 1–4, 2011.
- 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 **Proc. 8th Int’l Conf. Computer Vision**, v. 2, pp. 416–423. July 2001.
- MEYER, F. The dynamics of minima and contours. In P. Maragos; R. Schafer e M. Butt, editores, **Mathematical Morphology and its Applications to Image and Signal Processing**, v. 5 de *Computational Imaging and Vision*, pp. 329–336. Springer US, 1996.
- MONTEIRO, F. e CAMPILHO, A. Watershed framework to region-based image segmentation. In **Pattern Recognition, 2008. ICPR 2008. 19th International Conference on**, pp. 1–4. 2008.
- PEDRONETTE, D.; DA SILVA TORRES, R. e CALUMBY, R. Using contextual spaces for image re-ranking and rank aggregation. **Multimedia Tools and Applications**, pp. 1–28, 2012.
- ROERDINK, J. e MEIJSTER, A. The watershed transform: Definitions, algorithms and parallelization strategies. In **Fundamenta Informaticae**, pp. 187–228. 2000.
- 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.
- THULASIRAMAN, K. e SWAMY, N. **Graphs: Theory and Algorithms**. Wiley, 2011. ISBN 9781118030257.
- TORRES, R.S.; FALCÃO, A.X.; GONÇALVES, M.A.; PAPA, J.P.; ZHANG, B.; FAN, W. e FOX, E.A. A genetic programming framework for content-based image retrieval. **Pattern Recognition**, pp. 283–292, 2009.
- WILSON, R. e WATKINS, J. **Graphs: an introductory approach : a first course in discrete mathematics**. Wiley, 1990. ISBN 9780471615545.
- WU, Z. e LEAHY, R. An optimal graph theoretic approach to data clustering. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, pp. 1101–1113, 1993.
- YANG, X.; BAI, X.; LATECKI, L. e TU, Z. Improving shape retrieval by learning graph transduction. **Computer Vision – ECCV 2008**, pp. 788–801, 2008.

YANG, X.; KOKNAR-TEZEL, S. e LATECKI, L. Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In **IEEE Conference on Computer Vision and Pattern Recognition**, pp. 357–364. 2009.

# **Apêndice A**

## **Publicações**

A seguir serão apresentados os artigos originais dos autores. Tais artigos foram publicados em eventos internacionais e são relacionados à pesquisa desenvolvida nessa dissertação.

# Image Segmentation Using Watershed and Normalized Cut

Marco Antonio Garcia de Carvalho  
Tiago William Pinto  
Anselmo Castelo Branco Ferreira  
School of Technology  
University of Campinas - UNICAMP  
Limeira/SP, Brazil  
{magic,t025323,a023169}@ft.unicamp.br

Roberto Marcondes César Júnior  
Department of Computer Science - IME  
University of São Paulo - USP  
São Paulo/SP, Brazil  
cesar@ime.usp.br

**Abstract**—Several publications have studied the graph spectra (a concept extracted from linear algebra) as a partitioning tool. This work studies a method called Normalized Cut, introduced by Shi and Malik [1] and proposes an image segmentation strategy utilizing two ways to convert images into graphs: Pixel affinity and watershed transform. Both ways provide us as result a similarity matrix that is used to calculate the spectral graph properties (eigenvalues and eigenvectors) and then apply the grouping algorithm. Images of yeast cells were used to generate the preliminary results obtained and provide important knowledge for future research.

**Keywords**-image segmentation; watershed transform; graph partitioning; spectral graph.

## I. INTRODUCTION

The relationship between some linear algebra concepts and graph theory provides an way to segment an image into meaningful regions and extract its semantic information. Spectral graph theory has evolved and encouraged numerous works on digital image processing domain[2].

Graph cuts may be explored to analyze the degree of dissimilarity between parts of a graph (or its corresponding image). In this paper, we study an approach based on graph cuts called Normalized Cut (Ncut) [1] and its association with concepts extracted from the study of eigenvalues and eigenvectors problems.

Our experiments use two approaches in order to convert images into graphs: The brightness formula showed in [1] (Pixel Affinity) and the Watershed Transform [3],[4].

## II. IMAGE-GRAPH CONVERSION METHODS

Firstly, we need to convert the image into a graph. Given an image  $I$ , this work uses the following methods in order to implement this conversion.

### A. Pixel affinity

The overall quality of the segmentation depends on the pairwise pixel affinity graph. The chosen properties were the intensity and position approach presented in [1]. In this formulation, each pixel corresponds to one node in a graph. Because the graph must be not cyclic, the affinity between

a pixel and itself is 0. Equation (1) shows the formula to calculate the affinity matrix  $W_i(i, j)$ .

$$W_i(i, j) = e^{-\frac{\|X_i - X_j\|^2}{\sigma_x} - \frac{\|C_i - C_j\|^2}{\sigma_c}} \quad (1)$$

where  $X_i$  and  $X_j$  are vectors containing the xy positions of the pixels  $i$  and  $j$ ;  $C_i$  and  $C_j$  are vectors containing the RGB values (or graylevel values) of these same pixels.

### B. Watershed Transform

We consider the gradient image as a topographic surface. In the watershed method, an image is segmented by constructing the catchment basins of its gradient image. The gradient image is flooded starting from selected sources (regional minima) until the whole image has been flooded. A dam is erected between lakes that meet with others lakes. At the end of the flooding process, we obtain one region for each catchment basin of the gradient image.

Hierarchical Watershed creates a set of nested partitions, i.e., a hierarchy. In this case, a partition at a coarse level is obtained by merging regions of the fine partition[4].

The watershed problem can be modeled using graphs. The gradient image is represented by a weighted neighborhood graph, where a node represents a catchment basin of the topographic surface. We use Hierarchical Watershed in order to reduce the number of nodes (supersegmentation problem) in the correspondent graph.

After the conversion, we use the area and average grayscale level of each region to set the edges weights between them.

## III. GRAPH REPRESENTATION

We assume  $G = (V, E)$  as an undirected graph where  $V$  is the set of nodes and  $E$  is the set of edges  $(i, j)$ . Two nodes  $i, j$  are adjacent, represented by  $i, j$ , if there exists an edge linking  $i$  and  $j$ , and the weight associated to each edge  $(i, j)$  is represented by  $w(i, j)$ . The mathematical representation for this graph is given as follows:

- Similarity matrix: A similarity matrix  $A$  is a representation for an undirected weighted graph where each entry value  $a(i, j)$  is the edge weight  $w(i, j)$  linking a pair of nodes. The weights are given by a function that maximize the similarity between nodes  $i$  and  $j$ .

- Weighted degree matrix: Let  $d(i) = \sum_j w(i, j)$  be the total connection from node  $i$  to all its neighbor nodes. Then the weighted degree matrix  $D$  is the diagonal matrix with  $d$  on its diagonal.
- Laplacian matrix: The Laplacian matrix of a graph  $G$  is computed from  $L(G) = (D - A)$ , where  $D$  is the weighted degree matrix and  $A$  is the similarity matrix.

#### IV. GRAPH PARTITIONING

The Ncut approach uses the algebraic properties of the Laplacian matrix to separate the nodes according to the dissimilarity between them. In graph theoretic language, it is called cut:

$$\text{Cut}(S_1, S_2) = \sum_{u \in S_1, v \in S_2} w(u, v) \quad (2)$$

where  $S_1$  and  $S_2$  are two disjoint sets in a graph. Instead of using the total edge weight connection, this method computes the cut cost as a fraction of the total edge connections to all the nodes in the graph:

$$N \text{Cut}(S_1, S_2) = \frac{\text{Cut}(S_1, S_2)}{\text{SumCon}(S_1, V)} + \frac{\text{Cut}(S_2, S_1)}{\text{SumCon}(S_2, V)} \quad (3)$$

where  $\text{SumCon}(S, V)$  is the total connection from nodes in the set  $S$  with all nodes in  $V$ . Expanding this equation the following equation can be found:

$$\min_x N \text{Cut}(x) = \min_y \frac{y^T (D - A) y}{y^T D y} \quad (4)$$

This equation, called Rayleigh quotient, has a property that it can be minimized by the smallest eigenvector  $x_0$  of the Rayleigh quotients matrix (in this case, the Laplacian matrix) and its minimum value is the corresponding eigenvalue  $\lambda_0$ . So, the Normalized cut can be minimized by solving a generalized eigenvalue system as shown below.

$$(D - A) y = \lambda D y \quad (5)$$

Because of the first Laplacian matrix smallest eigenvalue is 0, the second smallest eigenvalue is the real valued solution to the normalized cut problem and its corresponding eigenvector can tell exactly how to partition the graph just by separating the nodes represented by the positive values in the eigenvector from the negative ones.

#### V. EXPERIMENTAL RESULTS

We applied the conversion methods and the segmentation strategy presented in yeast cells images, thanks to its importance in several chemical processes and previous related works [5]. Applying the pixel affinity method to generate a graph the results are showed in Figure 1. The original image, Figure 1(a) is converted into a graph using the pixel affinity method. After, the eigenvalues and eigenvectors are calculated. The sign of the second smallest eigenvectors values provides the partition presented on Figure 1(b).

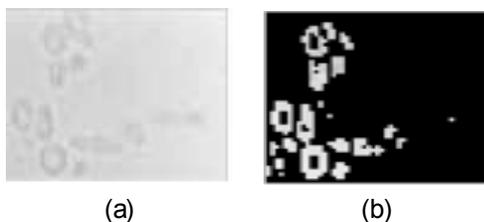


Figure 1. (a)Yeast original image; (b) yeast segmentation by N-Cut (pixel affinity).

Figure 2 shows the results of the segmentation process using the Watershed Transform (WT) and Normalized Cut. Figure 2(b)

shows the cells separated from the background (first iteration); Figures 2(c) and (d) show the cells grouped by their size similarities (second iteration).

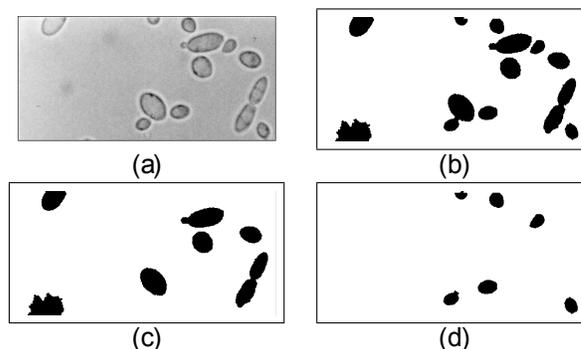


Figure 2. (a)Yeast original image; (b) yeast segmentation by WT and N-Cut; (c), (d) yeast segmentation grouped by area.

#### VI. CONCLUSIONS AND PERSPECTIVES

Our experiments were satisfactory to group the image regions that represent semantically the yeast cells, when WT was used. Minimizing the Normalized Cut is a NP-complete [1], and the comparison between the two presented methods of image conversion showed that the performance is higher using the Hierarchical Watershed Transform, because it simplifies the graph in a lower number of nodes (different from [2]). The studies about the Ncut and the spectral graph theory allow numerous combinations of methods due to its versatility. Once you can represent the problem using a graph and determine the similarity matrix, the grouping algorithm will be similar. In this way, the properties chosen to compute the edge's weights are important for the final result. In order to improve our results, future works can combine the image conversion methods presented and consider the region's shape as a similarity criterion.

#### REFERENCES

- [1] J. Shi, J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI*, vol. 22, no. 8, pp. 888-905, 2000.
- [2] F. C. Monteiro, A. Campilho. Watershed framework to region-based image segmentation. In: *Proc. Of IEEE 19th International Conference on Pattern Recognition- ICPR*, pp. 1-4, 2008.
- [3] F. Meyer, P. Maragos. Multiscale Morphological Segmentations based on Watershed, Flooding and Eikonal PDE. *Scale-Space*, pp. 1-12, 1999.
- [4] M. A. G. Carvalho. Hierarchical image analysis through the tree of critical lakes (in portuguese). In *PhD Thesis, Electrical and Computer Engineering Faculty, University of Campinas UNICAMP, Brazil, 2004.*
- [5] M. A. G. Carvalho, T. W. Pinto. Comparing yeast cells segmentation through hierarchical trees. In: *Proc. Of International Conference on Computer Vision Theory and Applications - VISAPP, Setbal - Portugal. v. 1. p. 515-518, 2006.*

# Combining Hierarchical Watershed Metrics and Normalized Cut For Image Segmentation

Tiago Willian Pinto & Marco Antonio Garcia de Carvalho

School of Technology - FT  
University of Campinas - UNICAMP  
Limeira - SP, Brazil

Combining partitioning techniques became a promising approach in order to implement image segmentation and has produced good results on different applications. Some works has studied the graph spectrum as a partitioning tool, by means of the so-called Normalized Cut method (NCut) as a final partitioning process for images modeled by different types of graphs. This work explores the Watershed Transform as a modeling tool, using different criteria of the hierarchical Watershed (Area, Volume and Dynamics), in order to convert an image into a graph, followed by the use of the NCut to perform the final segmentation. The main goal is to compare the image segmentation results obtained from graphs modeled by Watershed Transform with others relevant results, using images from Berkeley Database.

Keywords: image segmentation; watershed transform; graph partitioning; normalized cut.

## 1 INTRODUCTION

The complexity of Image Segmentation makes it still in constant research for performance improvement and results enhancement, since there is not exists a general solution for every computational application. Some early publications made combinations of segmentation techniques to achieve better results and these hybrid approaches has brought promising results, i.e. (Monteiro & Campilho 2008), (Carvalho & Ferreira & Costa & Cesar Jr 2010). This paper will combine the Watershed Transform, a classic method used for image segmentation purposes, and the Normalized Cut (NCut), an early technique that explores linear algebra concepts and graph theory to segment an image into meaningful regions and take its semantic information, i.e. (Shi & Malik 2000). The Watershed Transform has the inconvenience of over-segmentation problems and NCut is known as a NP-hard complexity algorithm, but these weakness are treated naturally with the combination of both. The same process was done in (Monteiro & Campilho 2008) and (Bock & Smet & Philips 2005), but our approach will use different criteria of watershed and neighborhood structure for graph definition, in order to compare the results obtained

with the original formula of NCut and another results in the benchmark found in (Ferreira, 2010). The next sections are organized as follows: In section 2 we give a general review of graph concepts that will be used in this work. Section 3 shows a review of the Watershed Transform and the criteria used for the graph modeling and section 4 shows the NCut theory basis. Experiments and partial results are showed in section 5 and finally in section 6 the conclusions and perspectives for future works are exploited.

## 2 GRAPH REPRESENTATION

Assuming  $G = (V, E)$  as an undirected graph where  $V$  is the set of nodes and  $E$  is the set of edges  $(i, j)$ . Two nodes  $i, j$  are adjacent, represented by  $i \sim j$ , if there exist an edge linking  $i$  and  $j$ , and the weight associated to each edge  $(i, j)$  is represented by  $w(i, j)$ . The mathematical representation for this graph is given as follows:

- Similarity matrix: A similarity matrix  $A$  is a representation for a undirected weighted graph where each entry value  $a(i, j)$  is an edge weight  $w(i, j)$  linking a pair of nodes. The weights are given by a function that maximize the similarity between nodes  $i$  and  $j$ .

- **Weighted degree matrix:** Let  $d(i) = \sum_j w(i, j)$  be the total connection from node  $i$  to all its neighbor nodes. Then the weighted degree matrix  $D$  is the diagonal matrix with  $d$  on its diagonal.
- **Laplacian matrix:** The Laplacian matrix of a graph  $G$  is computed from  $L(G) = (D - A)$ , where  $D$  is the weighted degree matrix and  $A$  is the similarity matrix.

### 3 WATERSHED TRANSFORM

We consider the gradient image  $I$  as a topographic surface. In the watershed method, an image is segmented by constructing the catchment basins of its gradient image. The gradient image is flooded starting from selected sources (regional minima) until the whole image has been flooded. A dam is erected between lakes that meet with others lakes. At the end of flooding process, we obtain one region for each catchment basin of the gradient image.

Hierarchical watershed creates a set of nested partitions. A partition  $P$  of an image  $f$  is a set of disjoint regions  $R_i$ ,  $i = 1, 2, \dots, n$ , where the union of regions is the whole image. Let  $(P_k)$  be a sequence of partitions  $P_1, P_2, \dots, P_n$ , of an image  $f$ .  $(P_k)$  is a hierarchy, also called nested sequence of partitions, if a partition at a fine level is obtained by merging regions of the coarse partition. Some criteria exploited in this paper are:

- **Volume and Area:** Geometric operators given for each region;
- **Dynamics:** Relationship between regional minima altitude, (Meyer 1996).

The watershed problem can be modeled using graphs. The gradient image is represented by a weighted neighborhood graph, where a node represents a catchment basin (region) of the topographic surface and two nodes are linked by an edge when its regions are neighbours. The use of Hierarchical Watershed can reduce the number of nodes (oversegmentation problem) in the correspondent graph enhancing the segmentation process performance.

### 4 NORMALIZED CUT

The Normalized Cut (Ncut) approach uses the algebraic properties of the Laplacian matrix to separate the nodes according to the dissimilarity between them. In graph theoretic language, it is called cut and defined by:

$$\text{Cut}(S_1, S_2) = \sum_{u \in S_1, v \in S_2} w(u, v) \quad (1)$$

where  $S_1$  and  $S_2$  are two disjoint sets in a graph. Instead of using the total edge weight connection, this method computes the cut cost as a fraction of the total edge connections to all nodes in the graph:

$$N\text{Cut}(S_1, S_2) = \frac{\text{Cut}(S_1, S_2)}{\text{SumCon}(S_1, V)} + \frac{\text{Cut}(S_1, S_2)}{\text{SumCon}(S_2, V)} \quad (2)$$

where  $\text{SumCon}(S, V)$  is the total connection from nodes in the set  $S$  with all nodes in  $V$ . Expanding this equation the following equation can be found:

$$\min_x N\text{Cut}(x) = \min_y \frac{y^T (D - A) y}{y^T D y} \quad (3)$$

This equation, called Rayleigh quotient, has a property that it can be minimized by the smallest eigenvector  $x_0$  of the Rayleigh quotients matrix (in this case, the Laplacian matrix) and its minimum value is the corresponding eigenvalue  $\lambda_0$ . So, the Normalized cut can be minimized solving a generalized eigenvalue system as shown below.

$$(D - A)y = \lambda D y \quad (4)$$

Because of the first Laplacian's matrix smallest eigenvalue is 0, the second smallest eigenvalue is the real valued solution to the normalized cut problem. The corresponding eigenvector can tell exactly how to partition the graph only by separating the nodes represented by the positive values in the eigenvector from the negative ones for instance.

### 5 EXPERIMENTS

We obtained some partial results in our experiments using images for general purposes from the Berkeley Database, in Figure 1 we show the original images; in Figure 2 is shown the segmentation using the original Ncut method. The main goal was to analyse the proposed method and compare the parameters used in the modeled graphs. Firstly, we modeled the images into graphs using hierarchical watershed, reducing the image into 5 regions and building a fully connected graph. The metric used to calculate the similarity matrix were the average grayscale between every pair of nodes. These results are given in Figure 3. Other experiments were done using the region's area as edge weight in the similarity matrix. The same process was done, but now using 15 regions in the hierarchical watershed, the results are in Figure 4.

The next step is to consider the other metrics proposed (volume and dynamics) to model the weighted graph generated by the watershed transform and compare the final results.

## 6 CONCLUSIONS AND FUTURE WORKS

In this paper our main goal is to propose a combination of segmentation methods with high relevance in the related literature, in order to generate and compare some segmentations results for images of general purposes from Berkeley segmentation dataset. Our perspectives are that this method can achieve good results with reduced computation time (compared with the original techniques) and then use that for images of specific applications in future works. Also, we wish to evaluate the quality of results obtained, and compare them, through the implementation of a benchmark.

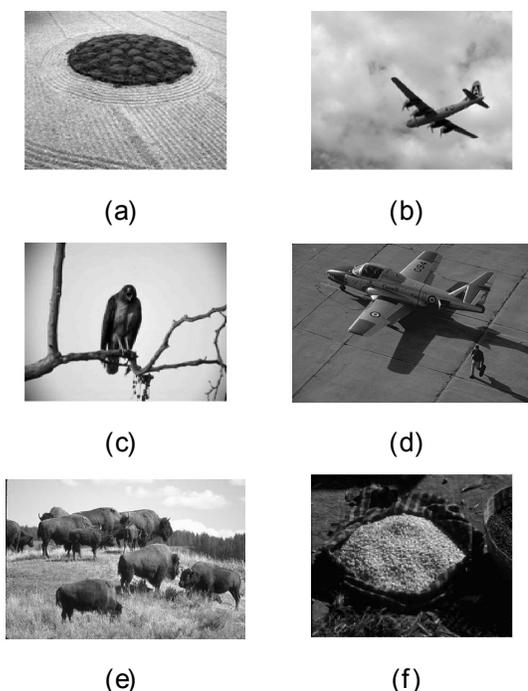


Figure 1: Images from Berkeley Database: (a) 86016; (b) 3096; (c) 42049; (d) 37073; (e) 38092; (f) 58060.

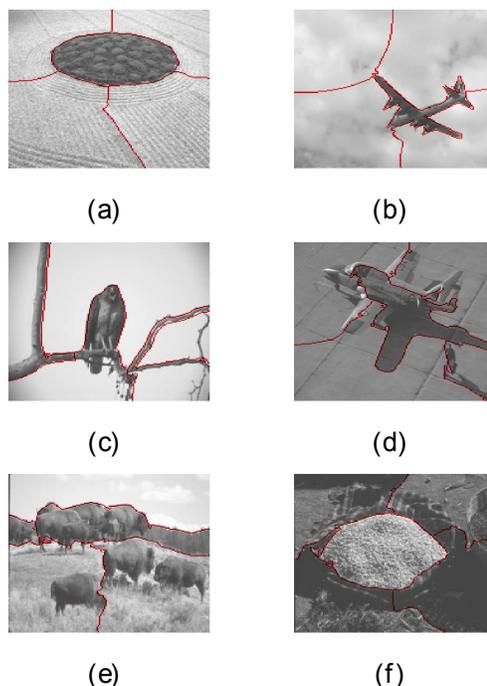


Figure 2: Segmentation using the NCut method proposed in Shi & Malik (2000). Images: (a) 86016; (b) 3096; (c) 42049; (d) 37073; (e) 38092; (f) 58060.

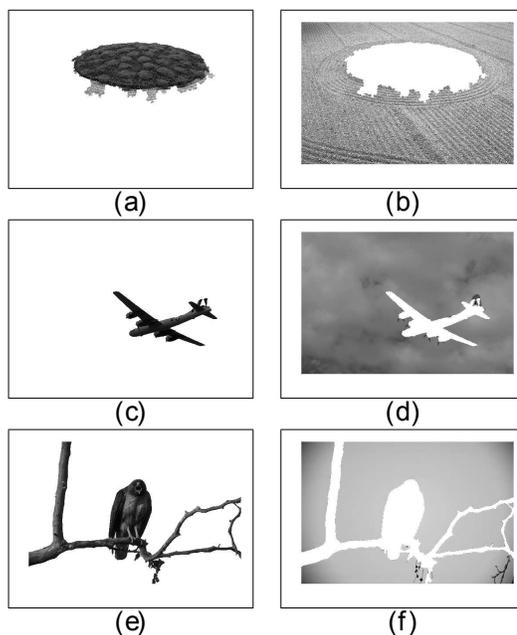


Figure 3: Segmentation using proposed method. Hierarchical Watershed with 5 regions and similarity graph modeled using region average grayscale as metric. Images: (a,b) 86016; (c,d) 3096; (e,f) 42049.

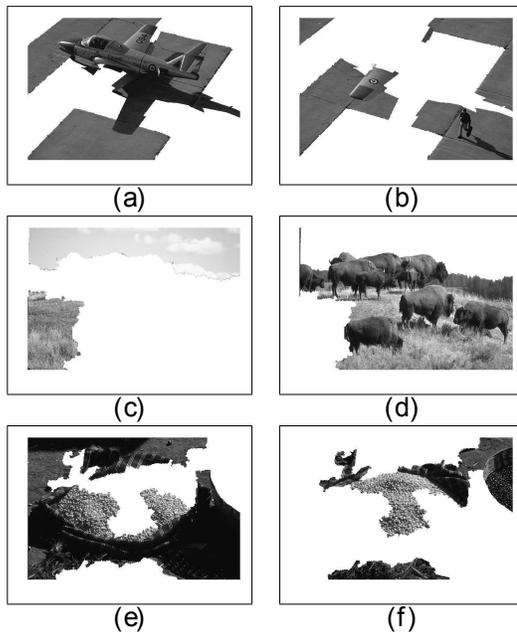


Figure 4: Segmentation using proposed method. Hierarchical Watershed with 15 regions and similarity graph modeled using region area as metric. Images: (a,b) 37073; (c,d) 38092; (e,f) 58060.

## 7 REFERENCES

BOCK, J. De, SMET, P. D., and PHILIPS, W. Image segmentation using watersheds and normalized cuts. In: Proc. of the IS&T/SPIE Electronic Imaging, volume 5675, pp. 164173, 2005.

CARVALHO, M.A.G.; FERREIRA, A.C.B.; COSTA, A.L.; CESAR-JR., R.M. Image segmentation using component tree and normalized cuts. In: Proc. of 23th Brazilian Symposium on computer graphics and image processing - SIBGRAPI, pp. 317-320, 2010.

FERREIRA, A.C.B. Um estudo comparativo de segmentacao de imagens por aplicacoes do Corte Normalizado em Grafos. 2010. 94 p. Dissertacao (Mestrado em Tecnologia e Inovacao) - Faculdade de Tecnologia, Universidade Estadual de Campinas, Limeira, 2010.

MARTIN, D.; FOWLKES C.; TAL, D.; MALIK, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: Proc. of 8th Int'l Conf. Computer Vision, vol. 2, pp. 416-423, 2001.

MONTEIRO, F. C.; CAMPILHO, A. Watershed framework to region-based image segmentation. In: Proc. Of IEEE 19th International Conference on Pattern Recognition- ICPR, pp. 1-4, 2008.

SHI, J.; MALIK, J. Normalized cuts and image segmentation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI, vol. 22, no. 8, pp. 888-905, 2000.

MEYER, F.; MARAGOS, P. The dynamics of minima and contours. In: P. Maragos, R. W. Schafer and M. A. Butt Editors, Mathematical Morphology and its Applications to Image and Signal Processing, pp.329-336, 1996.

# Image Segmentation through Combined Methods: Watershed Transform, Unsupervised Distance Learning and Normalized Cut

Tiago W. Pinto, Marco A. G. de Carvalho, Daniel C. G. Pedronette and Paulo S. Martins

**Abstract**—Research on image processing has shown that combining segmentation methods may lead to a solid approach to extract semantic information from different sort of images. Within this context, the Normalized Cut (NCut) is usually used as a final partitioning tool for graphs modeled in some chosen method. This work explores the Watershed Transform as a modeling tool, using different criteria of the hierarchical Watershed to convert an image into an adjacency graph. The Watershed is combined with an unsupervised distance learning step that redistributes the graph weights and redefines the Similarity matrix, before the final segmentation step using NCut. Adopting the Berkeley Segmentation Data Set and Benchmark as a background, our goal is to compare the results obtained for this method with previous work to validate its performance.

**Index Terms**—image segmentation; watershed transform; graph partitioning; normalized cut; unsupervised distance learning.

## I. INTRODUCTION

Image segmentation is a fundamental tool for image understanding and a challenging problem in computer vision. Although many different segmentation methods have been proposed, the problem is still far from being completely solved due to its convoluted and somehow subjective range space.

Research work on the NCut and the spectral graph theory allows numerous combinations of methods due to their versatility. Once a problem is represented using a graph, along with its similarity matrix, the grouping algorithm will be similar, and the properties selected to compute the edge's weights are critical to the final result.

Monteiro and Campilho [6] proposed a method that combines flooding watershed with NCut. In their work, each node is connected to all other nodes in the graph. The weights are assigned according to a contour continuity function with origin at the central pixel in each region. A pre-processing step is used to reduce the spatial resolution without losing important image information. An initial partitioning of the image into primitive regions is set by applying a rain-falling watershed algorithm on the image gradient magnitude. This initial partition is the input to a computationally efficient region segmentation process which produces the final segmentation. The latter segmentation uses a region-based similarity graph representation

of the image regions. Experimental results clearly demonstrate the effectiveness of the proposed approach to produce simpler segmentations that yet compare favorably with state-of-the-art methods.

Bock, Smet and Philips proposed a combination of rain-falling watershed and NCut [3]. Both the similarity graph and the weights on edges follow the same approach of Shi and Malik [9], using the central pixel and the brightness average of each region for each segmented region.

More recently, contextual distance learning methods have been proposed for content-based image retrieval tasks, aiming at improving the effectiveness of their systems [4], [7]. Contextual distance measures take into account the local distribution of the sample objects and iteratively estimates distance updates, thereby modifying the neighborhood structure [4].

In this work, we apply the recently proposed Contextual Spaces Algorithm (by Pedronette et al [7], [8]) to perform a distance learning step for image segmentation tasks. The goal is to improve distance measures among pre-segmented regions of the image by considering contextual information encoded in the neighborhood structure. To our knowledge, this is the first attempt to construct and validate a segmentation framework that includes the contextual spaces algorithm in order to rebalance the edges of the similarity matrix.

The method under analysis in this paper is denoted combined method, as opposed to the method used for comparison, the original method. The original method consists of a watershed transform followed by N-Cut. It differs from the combined method in that it does not include the Unsupervised Context Distance step (reranking) (Section II-C).

The remainder of this paper is organized as follows: In Section II we discuss the methods and algorithms that are integrated in the proposed segmentation framework. Section III introduces the combined approach, and the experiments and the results are addressed in Section IV. Finally, in Section V we present the conclusion.

## II. BACKGROUND

This section introduces all the elements that compose the segmentation framework proposed. Therefore, it briefly addresses the watershed transforms and the normalized cut. Since these are quite popular methods, the reader is referred to the literature for more details [3]. It also discusses in a bit more detail the unsupervised distance learning and its contextual spaces algorithm.

D. C. G. Pedronette is with the Department of Statistics, Applied Mathematics and Computing, UNESP, Rio Claro - 13506-900, São Paulo, Brazil e-mail: (daniel@rc.unesp.br).

T. Pinto, M. Carvalho and P. Martins are with College of Technology, UNICAMP, Limeira - 13484-332, São Paulo, Brazil, e-mail: (t025323@dac.unicamp.br, { magic, paulo }@ft.unicamp.br)

D. C. G. Pedronette would like to thank FAPESP (grant 2013/08645-0)

### A. Watershed Transform

Considering a greyscale image as a topographic surface, in the watershed method an image is segmented by constructing the catchment basins of its image. The greyscale image is flooded starting from selected sources (regional minima) until the whole image has been flooded. A dam is erected between lakes that meet with others lakes. At the end of the flooding process, we obtain one region for each catchment basin of the greyscale image.

Hierarchical watershed creates a set of nested partitions. A partition  $P$  of an image  $f$  is a set of disjoint regions  $R_i$ ,  $i = 1, 2, \dots, n$ , where the union of regions is the whole image. Let  $(P_k)$  be a sequence of partitions  $P_1, P_2, \dots, P_n$ , of an image  $f$ .  $(P_k)$  is a hierarchy, also called nested sequence of partitions, if a partition at a fine level is obtained by merging regions of the coarse partition.

The watershed problem can be modeled using graphs. The greyscale image is represented by a weighted neighborhood graph, where a node represents a catchment basin (region) of the topographic surface and two nodes are linked by an edge when its regions are neighbors. The use of Hierarchical Watershed can reduce the number of nodes (over-segmentation problem) in the corresponding graph, enhancing the performance of the segmentation process.

### B. Normalized Cut

The Normalized Cut (NCut) [9] approach uses the algebraic properties of the Laplacian matrix to separate the nodes according to the dissimilarity between them. In graph-theoretic language it is called a cut and defined by:

$$\text{Cut}(S_1, S_2) = \sum_{u \in S_1, v \in S_2} w(u, v) \quad (1)$$

where  $S_1$  and  $S_2$  are two disjoint sets in a graph. Instead of using the total edge weight connection, this method computes the cut cost as a fraction of the total edge connections to all nodes in the graph:

$$\text{NCut}(S_1, S_2) = \frac{\text{Cut}(S_1, S_2)}{\text{SumCon}(S_1, V)} + \frac{\text{Cut}(S_1, S_2)}{\text{SumCon}(S_2, V)} \quad (2)$$

where  $\text{SumCon}(S, V)$  is the total connection from nodes in the set  $S$  with all nodes in  $V$ . The following equation can be found by expanding equation (2):

$$\min_x \text{NCut}(x) = \min_y \frac{y^T (D - A) y}{y^T D y} \quad (3)$$

This equation, called Rayleigh quotient, can be minimized by the smallest eigenvector  $x_0$  of the Rayleigh quotients matrix (in this case, the Laplacian matrix) and its minimum value is the corresponding eigenvalue  $\lambda_0$ . Therefore, the Normalized cut can be minimized solving a generalized eigenvalue system as shown below:

$$(D - A)y = \lambda D y \quad (4)$$

Because of the first Laplacian's matrix smallest eigenvalue is 0, the second smallest eigenvalue is the real valued solution to the normalized cut problem. The corresponding eigenvector can tell exactly how to partition the graph, e.g. by separating the nodes represented by the positive values in the eigenvector from the negative ones.

### C. Unsupervised Distance Learning

Several real-world problems are modeled by a set of objects (or data points) sampled from a data manifold and represented as a graph. The edge weights are determined by distances (or similarities) between pairs of objects. In general, the pairwise distance measures are unreliable due to noise or due to intrinsic difficulties in estimating similarity values between objects [10]. More effective distance measures can be computed if the original distances are replaced by more global measures, which are able to encode information of the entire collection.

A central problem in the vision and learning community [5] lies in how to capture and utilize the intrinsic manifold structure of a collection without the need of any training data. In this scenario, manifold learning methods were proposed, with the goal of explicitly construct a new embedding space with a corresponding metric that is more faithful to the manifold structure - and hence induces a better distance/similarity measure [5].

### D. Contextual Spaces Algorithm

The Contextual Spaces algorithm redefines the relationships between objects in a collection by answering the question "What information can my nearest neighbors provide about other collection objects?". Therefore, not only the pairwise similarity between two samples is considered, but also the contextual information encoded in the whole neighborhood [7], [8].

A contextual space is a bi-dimensional space where all collection objects are represented according to their distances to an arbitrary object and each one of its  $k$ -nearest neighbors. For the segmentation task, each object is considered as a pre-segmented region of an image.

Let  $r_i$  and  $r_l$  be regions given by a pre-segmentation of an image  $I$ , and let  $\rho(r_i, r_l)$  be a distance between  $r_i$  and  $r_l$ . We can compute the  $k$ -nearest neighbors of the region  $r_i$  based on the distance  $\rho$ . For each  $k$ -nearest neighbor of  $r_i$ , a contextual space is constructed aiming at redefining distances from  $r_i$  to other regions of the image.

New distances  $\rho_{t+1}$  are computed merging information of  $k$  contextual spaces in an unique space (where  $t$  indicates the iteration being processed). The new distances are defined according to two dimensions:  $d_i$  and  $d_j$ . Given two regions  $r_i, r_l$ , the dimension  $d_i$  represents the distance information from  $r_i$  to  $r_l$ . The dimension  $d_j$  represents the distance information from neighbors  $r_j = \text{kNN}(r_i)$  to  $r_l$ . The dimension  $d_i$  is computed as follows:

$$d_i = \frac{\rho_t(r_i, r_l)}{k} \quad (5)$$

The dimension  $d_j$  considers a different weight to each neighbor, where neighbors at top positions present higher weights. In this work, we used the Mutual-kNN approach [8], which considers not only the information of nearest neighbors, but also the mutual reference between images and their neighbors in their ranked lists. The dimension  $d_j$  is computed as follows:

$$d_j = \frac{\sum_{i=1}^k \rho_i(r_j, r_i) \times (k-j)}{\frac{k \times (k-1)}{2}} \quad (6)$$

Finally, the next iteration distance  $\rho_{t+1}$  is computed as follows:

$$\rho_{t+1}(r_i, r_l) = \sqrt{d_i^2 + d_l^2} \quad (7)$$

The process is iteratively repeated along  $T$  iterations, aiming at computing a new and more effective distance  $\hat{\rho}(r_i, r_l)$  between regions  $r_i, r_l$ .

### III. COMBINED APPROACH

The proposed method consists of four major steps as illustrated in Fig. 1 and described as follows:

- First Step: Watershed Transform: This step involves the following operations: 1) Preprocessing, i.e. the conversion of original image to greyscale and extraction of the image gradient from the greyscale; 2) Watershed with Image Foresting Transform (IFT) processing on the image gradient; and 3) Simplification of primitive regions of Watershed through Hierarchical Watershed.

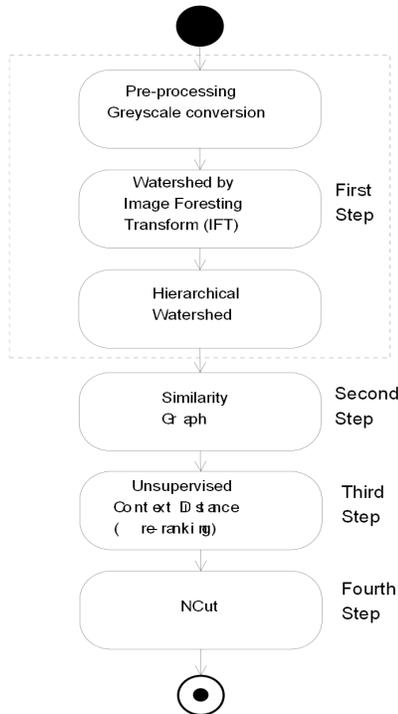


Fig. 1: UML Activity Diagram of the Combined Approach

- Second step: Similarity Matrix: This step computes the similarity graph. It employs the equations from Cour et al [2] as follows:

$$W_{IP}(i, j) = e \exp - \frac{\|X_i - X_j\|^2}{\sigma x} - \frac{\|I_i - I_j\|^2}{\sigma l} \quad (8)$$

$$W_C(i, j) = e \exp - \frac{\max(x) | \text{ine}(i, j) |^2}{\sigma C} \quad (9)$$

If the distance between the center of regions is less than the set radius, we have:

$$W_M(i, j) = \overline{W_I(i, j) - W_C(i, j)} + \alpha W_C(i, j) \quad (10)$$

where  $I$  is a mode function of the region (it takes the most frequent gray value);  $X_i - X_j$  is the distance between pixels  $i$  and  $j$ ;  $x$  is the center of a region;  $\sigma x = 4.0$ ,  $\sigma l = 0.01$ ,  $\sigma C = 1.5$  and radius = 15% of the largest possible distance within the image.

- Third step: Contextual Spaces Algorithm: The contextual spaces algorithm is applied once the similarity matrix is constructed and it rebalances the weights of the graph making it more suitable for the NCut algorithm. It was run using the following parameters:  $K=8$ ,  $T=13$ , and  $\lambda=23$ .
- Fourth Step: Segmentation via NCut: This step comprises the segmentation in 10 regions using the similarity graph obtained in Step 2; application of the re-ranking algorithm to the similarity graph; and segmentation in 10 regions using the similarity graph obtained from re-ranking.

### IV. EXPERIMENTS AND RESULTS

In this section we apply the proposed method to the Berkeley Segmentation Data Set (BDS500 Benchmark [1]) to obtain the Precision, Recall, F-measure and Region Coverage for all images in the training set and for both methods being compared. Table I summarize these results.

Regarding Precision, we found that 101 images (out of 200) presented better results after re-ranking, with an average of 15% of improvement and peaks above 40%. On the other hand, for the remaining images, there is an average of 30% of quality loss with peaks above 100%.

As for Recall, F-measure, Region Coverage the number of images that showed an improvement was not significant. Although there was still a gain in some cases, most images lost their quality regarding these measures. The best scores regarding the F-measure and Region Coverage were obtained in 5 images processed by re-ranking (i.e. images 22013, 46076, 55067, 106020, and 374067). Due to limitations of space, in Fig. 2 we show the results for the top two segmentations (i.e. 55067 and 106020). The top frame illustrates the BDS500 image, the middle one refers to the segmentation carried out by the original method, and the bottom image is the resulting segmentation from the combined method. Fig. 3 illustrates the F-measure for a sample of images from the BDS500 Data Set.

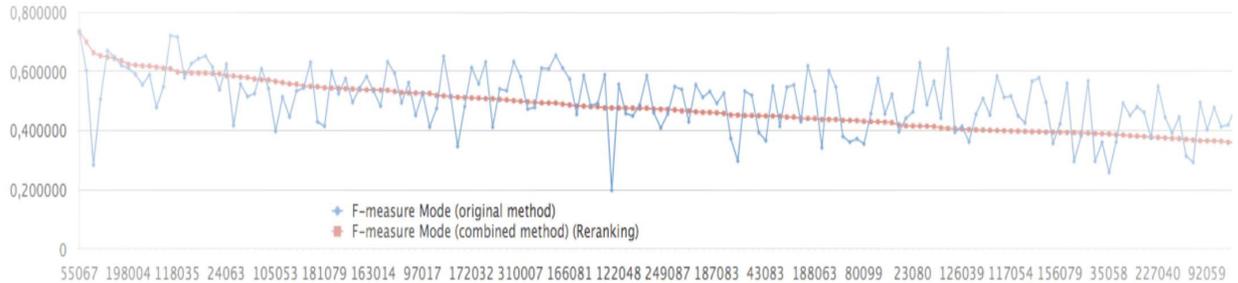


Fig. 3: F-Measure

TABLE I: Performance Assessment of Combined Method

Performance attributes	Original Method	Combined Method
Precision	0.613689	0.535201
Recall	0.405416	0.394238
F-Measure	0.488270	0.454030
Region Covering	0.411791	0.353255

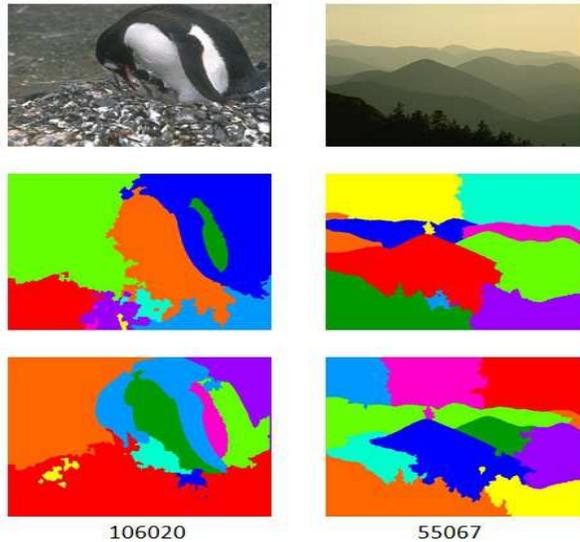


Fig. 2: BSDS500 Segmentation

## V. CONCLUSIONS

This paper examined a segmentation method that combines three major approaches: The Watershed transform, the Unsupervised Distance Learning Algorithm (Contextual Spaces Algorithm) and the Normalized Cut. The novelty lies in the introduction of the Unsupervised Distance Learning before the Normalized cut, in order to re-rank the similarity matrix before the actual cut.

We analyzed the performance of the method using conventional metrics such as precision, recall, F-measure and region coverage. The results show an improvement in the quality of a set of images, but not in all images from the data set.

Furthermore, while an improvement is observable regarding precision, this improvement does not systematically extend to all performance parameters (i.e. recall and region coverage). However, the initial results obtained in particular with five images are encouraging, and we believe that further work on this approach will allow us to fine tune both the algorithm and approach in order to make it completely feasible.

Future work aims at performing more experiments with the model, looking at the adjustment of the performance of the combined approach. The next step should also look at other metrics (volume, area and dynamics). Our perspective is that this method can achieve good results with potentially reduced computation time (compared with the original techniques) so that it can be used for images of specific applications in future work.

## REFERENCES

- [1] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intel.* 33(5), 898–916 (May 2011), <http://dx.doi.org/10.1109/TPAM.2010.161>
- [2] Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. vol. 2, pp. 1124–1131 vol. 2 (2005)
- [3] De Bock, J., De Smet, P., Philips, W.: A fast sequential rainfalling watershed segmentation algorithm. In: *Proceedings of the 7th International Conference on Advanced Concepts for Intelligent Vision Systems*. pp. 476–482. ACIVS'05, Springer-Verlag, Berlin, Heidelberg (2005)
- [4] Jegou, H., Schmid, C., Harzallah, H., Verbeek, J.: Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(1), 2–11 (2010)
- [5] Jiang, J., Wang, B., Tu, Z.: Unsupervised metric learning by self-smoothing operator. In: *IEEE International Conference on Computer Vision (ICCV'2011)*. pp. 794–801 (2011)
- [6] Monteiro, F., Campilho, A.: Watershed framework to region-based image segmentation. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. pp. 1–4 (2008)
- [7] Pedronette, D.C.G., da S. Torres, R.: Exploiting contextual spaces for image re-ranking and rank aggregation. In: *ACM International Conference on Multimedia Retrieval (ICMR'11)*. pp. 13:1–13:8 (2011)
- [8] Pedronette, D.C.G., da S. Torres, R., Tripodi, R.C.: Using contextual spaces for image re-ranking and rank aggregation. *Multimedia Tools and Applications* (2012), to appear, On-Line
- [9] Shi, J., Malik, J.: Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22(8), 888–905 (2000)
- [10] Yang, X., Prasad, L., Latecki, L.: Affinity learning with diffusion on tensor product graph. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35(1), 28–38 (2013)