

---

**Segmentação de imagens baseada em redes  
complexas e superpixels: uma aplicação ao censo  
de aves**

---

***Glenda Michele Botelho***

---

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

# Segmentação de imagens baseada em redes complexas e superpixels: uma aplicação ao censo de aves

**Glenda Michele Botelho**

**Orientador: Prof. Dr. João do Espírito Santo Batista Neto**

Tese apresentada ao Instituto de Ciências Matemáticas  
e de Computação - ICMC-USP, como parte dos  
requisitos para obtenção do título de Doutor em  
Ciências - Ciências de Computação e Matemática  
Computacional. *VERSÃO REVISADA*

**USP – São Carlos**  
**Novembro de 2014**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi  
e Seção Técnica de Informática, ICMC/USP,  
com os dados fornecidos pelo(a) autor(a)

B748s      Botelho, Glenda Michele  
                Segmentação de imagens baseada em redes complexas  
                e superpixels: uma aplicação ao censo de aves /  
                Glenda Michele Botelho; orientador João do Espírito  
                Santo Batista Neto. -- São Carlos, 2014.  
                82 p.

Tese (Doutorado - Programa de Pós-Graduação em  
Ciências de Computação e Matemática Computacional) --  
Instituto de Ciências Matemáticas e de Computação,  
Universidade de São Paulo, 2014.

1. Segmentação de Imagens. 2. Redes Complexas. 3.  
Superpixels. 4. Markov Random Fields. 5. Censo  
Demográfico de Aves. I. Batista Neto, João do  
Espírito Santo, orient. II. Título.



**Às três mulheres da minha vida:  
Helena, Elenita e Gleice...**



# Agradecimentos

---

À Deus, por todas as bençãos que vem me concedendo, dia após dia;

Ao Prof. Dr. João Batista Neto, sem o qual este trabalho não seria possível, pelos anos de orientação, pela atenção, paciência e competência;

Ao Prof. Dr. Francisco Rodrigues, pelas excelentes ideias e pela motivação constante;

À minha avó Helena e à minha mãe Elenita, pelo amor e apoio incondicionais, e por sempre acreditarem em mim;

À minha irmã, Gleice Lorena, pela companheirismo sem igual;

Ao meu colega, Oscar Cuadros, pelas horas de estudo e implementações;

Aos meus amigos do ICMC, pelos momentos de estudo e lazer;

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) ( $n^{\circ}$  do processo 2011/05802-2), pelo apoio financeiro;

Enfim, a todas as pessoas que, de uma forma ou de outra, vem contribuindo com meu crescimento profissional e pessoal.

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar.  
Mas, o mar seria menor se lhe faltasse uma gota.”

Madre Teresa de Calcutá

# Resumo

---

Uma das etapas mais importantes da análise de imagens e, que conta com uma enorme quantidade de aplicações, é a segmentação. No entanto, uma boa parte das técnicas tradicionais apresenta alto custo computacional, dificultando sua aplicação em imagens de alta resolução como, por exemplo, as imagens de ninhais de aves do Pantanal que também serão analisadas neste trabalho. Diante disso, é proposta uma nova abordagem de segmentação que combina algoritmos de detecção de comunidades, pertencentes à teoria das redes complexas, com técnicas de extração de *superpixels*. Tal abordagem é capaz de segmentar imagens de alta resolução mantendo o compromisso entre acurácia e tempo de processamento. Além disso, como as imagens de ninhais analisadas apresentam características peculiares que podem ser mais bem tratadas por técnicas de segmentação por textura, a técnica baseada em *Markov Random Fields* (*MRF*) é proposta, como um complemento à abordagem de segmentação inicial, para realizar a identificação final das aves. Por fim, devido à importância de avaliar quantitativamente a qualidade das segmentações obtidas, uma nova métrica de avaliação baseada em *ground-truth* foi desenvolvida, sendo de grande importância para a área. Este trabalho contribuiu para o avanço do estado da arte das técnicas de segmentação de imagens de alta resolução, aprimorando e desenvolvendo métodos baseados na combinação de redes complexas com *superpixels*, os quais alcançaram resultados satisfatórios com baixo tempo de processamento. Além disso, uma importante contribuição referente ao censo demográfico de aves por meio da análise de imagens aéreas de ninhais foi viabilizada por meio da aplicação da técnica de segmentação *MRF*.

**Palavras-chave:** Segmentação de Imagens, Detecção de Comunidades, Redes Complexas, *Superpixels*, *Markov Random Fields*, Textura, Censo Demográfico de Aves.

# Abstract

---

Segmentation is one of the most important steps in image analysis with a large range of applications. However, some traditional techniques exhibit high computational costs, hindering their application in high resolution images such as the images of birds nests from Pantanal, one of Brazilian most important wetlands. Therefore, we propose a new segmentation approach that combines community detection algorithms, originated from the theory of the complex networks, with superpixels extraction techniques. This approach is capable of segmenting high resolution images while maintaining the trade-off between accuracy and processing time. Moreover, as the nest images exhibit peculiar characteristics that can be better dealt with texture segmentation techniques, the Markov Random Fields (MRF) technique is proposed, as a complement to the initial approach, to perform the final identification of the birds. Finally, due to the importance of the quantitatively evaluation of the segmentation quality, a new evaluation metric based on ground-truth was developed, being of great importance to the segmentation field. This work contributed to the state of art of high resolution images segmentation techniques, improving and developing methods based on combination of complex networks and superpixels, which generated satisfactory results within low processing time. Moreover, an important contribution for the birds census by the analysis of aerial images of birds nests was made possible by application of the MRF technique.

**Keywords:** Image Segmentation, Communities Detection, Complex Networks, Superpixels, Markov Random Fields, Texture, Birds Census.

# Sumário

<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Lista de Trabalhos Publicados</b>	<b>xv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização e Motivação . . . . .	1
1.2 Objetivos . . . . .	3
1.3 Contribuições Originais . . . . .	4
1.4 Organização do Trabalho . . . . .	4
<b>2 Segmentação de Imagens baseada em Textura</b>	<b>6</b>
2.1 Considerações Iniciais . . . . .	6
2.2 Segmentação por <i>Markov Random Fields - MRF</i> . . . . .	7
2.2.1 Segmentação EM/MPM utilizando <i>MRF</i> . . . . .	8
2.2.2 Modelo para o Campo de Rótulos . . . . .	9
2.2.3 Modelo para a Imagem Observada . . . . .	11
2.2.4 Algoritmo MPM . . . . .	12
2.2.5 Algoritmo EM . . . . .	13
2.2.6 Algoritmo EM/MPM . . . . .	14
2.3 Segmentação baseada em <i>Local Binary Pattern (LBP)</i> . . . . .	15
2.4 Considerações Finais . . . . .	17
<b>3 Redes Complexas e Segmentação de Imagens</b>	<b>19</b>
3.1 Considerações Iniciais . . . . .	19

3.2	Representando a imagem como um grafo . . . . .	20
3.2.1	Conceitos básicos sobre grafos . . . . .	21
3.2.2	Formas de representar imagens como grafos . . . . .	21
3.3	Abordagens para a detecção de comunidades . . . . .	22
3.3.1	Algoritmos baseados em <i>Clustering</i> Hierárquico . . . . .	23
3.3.2	Algoritmos baseados na medida de Modularidade . . . . .	24
3.3.3	Algoritmos baseados na Propagação de Rótulos . . . . .	28
3.4	Agrupamento de Pixels - <i>Superpixels</i> . . . . .	29
3.4.1	<i>Simple Linear Iterative Clustering (SLIC)</i> . . . . .	29
3.4.2	<i>Speeded-Up TurboPixel</i> . . . . .	30
3.5	Considerações Finais . . . . .	31
<b>4</b>	<b>Uma proposta para segmentação de imagens de alta resolução</b>	<b>33</b>
4.1	Considerações iniciais . . . . .	33
4.2	Segmentação baseada em redes complexas e <i>superpixels</i> . . . . .	34
4.2.1	Extração de <i>Superpixels</i> . . . . .	34
4.2.2	Geração do Grafo . . . . .	36
4.2.3	Aplicação dos Algoritmos de Detecção de Comunidades . . . . .	38
4.3	Abordagem de segmentação baseada em <i>Markov Random Fields (MRF)</i> . . . . .	38
4.3.1	Estimação inicial dos parâmetros . . . . .	39
4.3.2	Simulated Annealing . . . . .	41
4.4	Nova métrica de avaliação quantitativa da qualidade das segmentações . . . . .	42
4.4.1	Novo método de seleção da imagem de referência . . . . .	44
4.5	Considerações finais . . . . .	45
<b>5</b>	<b>Resultados</b>	<b>47</b>
5.1	Considerações Iniciais . . . . .	47
5.2	Experimentos . . . . .	47
5.2.1	Influência da variação do tamanho dos <i>Superpixels</i> . . . . .	48
5.2.2	Uso de características de textura extraídas via LBP . . . . .	50
5.2.3	Avaliação da qualidade da segmentação e definição dos melhores valores para os parâmetros . . . . .	52
5.2.4	Comparação com método de segmentação relacionado . . . . .	54
5.3	Considerações Finais . . . . .	57

<b>6 Análise das Imagens de Ninhais de Aves do Pantanal</b>	<b>59</b>
6.1 Considerações Iniciais . . . . .	59
6.2 Aquisição de imagens de ninhais no Pantanal . . . . .	60
6.3 Pré-segmentação: aplicando a abordagem de segmentação baseada em redes complexas e <i>superpixels</i> . . . . .	60
6.4 Segmentação final: aplicando a técnica de segmentação baseada em <i>Markov Random Fields</i> . . . . .	63
6.4.1 Experimento 1: Definição dos valores dos parâmetros da segmentação baseada em <i>MRF</i> para análise de imagens de ninhais . . . . .	64
6.4.2 Experimento 2: Comparação da segmentação de imagens de ninhais considerando diferentes escalas . . . . .	65
6.4.3 Experimento 3: Comparação da contagem automática de aves com a contagem manual . . . . .	66
6.5 Considerações Finais . . . . .	68
<b>7 Conclusões</b>	<b>73</b>
7.1 Limitações . . . . .	74
7.2 Contribuições . . . . .	74
7.3 Trabalhos Futuros . . . . .	75
<b>Bibliografia</b>	<b>76</b>



# Listas de Figuras

1.1	Amostras de imagens aéreas de ninhais de aves do Pantanal.	3
2.1	Esquema de vizinhanças para os campos de <i>Markov</i> . A vizinhança 4-conectada é dada pelos pixels numerados com 1 e a vizinhança 8-conectada é representada pelos pixels numerados com 1 e 2.	10
2.2	Exemplos de Cliques.	10
2.3	Conjunto de vizinhos circularmente simétricos para diferentes valores de $(P, R)$ .	15
2.4	36 padrões binários locais únicos invariantes à rotação referentes à $LBP_{8,R}^{ri}$ . Os círculos preto e branco correspondem, respectivamente, ao valores de bit 0 e 1 em um padrão de 8 bits do operador. A primeira linha contém os nove <b>padrões uniformes</b> e os números dentro deles correspondem aos seus códigos <i>LBP</i> únicos. Figura adaptada de [Ojala et al. 2002].	17
3.1	Exemplo de comunidades presentes em uma rede complexa. Figura adaptada de Newman [Newman and Girvan 2004].	20
3.2	Dendograma que descreve a rede. Linha tracejada representa a partição com maior valor de modularidade.	26
3.3	Vértices são atualizados um a um, da esquerda para a direita. Devido a alta densidade das arestas (a maior neste caso), todos os vértices adquirem o mesmo rótulo.	28
3.4	Pixels localizados nas bordas dos segmentos são atualizados em cada execução.	30
4.1	Metodologia referente à segmentação de imagens aéreas de ninhais do Pantanal.	34
4.2	Nova abordagem de segmentação de imagens baseada em redes complexas e <i>superpixels</i> .	35
4.3	Região circular (área amarela) de raio $R=5$ .	37

4.4	Duas imagens segmentadas $S$ e $S'$ divididas em $R = 4$ e $R' = 2$ regiões, respectivamente. . . . .	43
4.5	Diferentes segmentações manuais para a mesma imagem devido à subjetividade do usuário. . . . .	45
4.6	Seleção da imagem de referência. . . . .	46
5.1	Variando o tamanho inicial dos <i>superpixels</i> . . . . .	48
5.2	Resultado de segmentação (a) e tempo computacional (b) considerando a segmentação de uma imagem sintética. . . . .	49
5.3	Configuração dos vizinhos usados no experimento: $P = 8$ , $R = 1$ e interpolação 8-conectada. . . . .	50
5.4	Resultados da aplicação do método de segmentação proposto em imagens texturizadas. Função de peso, usada na geração do grafo, baseada na técnica <i>LBP</i> . . . . .	51
5.5	Segmentação de uma imagem real, cujo grafo foi montado usando uma função de peso baseada na técnica de extração de textura <i>LBP</i> . . . . .	52
5.6	Histogramas da qualidade da segmentação ( $I(S, S')$ ) para (a) $R = 1$ , (b) $R = 2$ , (c) $R = 3$ , (d) $R = 4$ e (e) $R = 5$ . O eixo $x$ representa o valor de qualidade e o eixo $y$ corresponde à quantidade de imagens. . . . .	54
5.7	Histogramas referente à relação entre a qualidade da segmentação $I(S, S')$ e o <i>threshold</i> , considerando (a) $R = 1$ , (b) $R = 2$ , (c) $R = 3$ , (d) $R = 4$ e (e) $R = 5$ . O eixo $x$ representa os valores do <i>threshold</i> adaptativo e o eixo $y$ representa a quantidade de imagens. Ressalta-se que considerou-se apenas as imagens com os maiores valores de qualidade. . . . .	55
5.8	Tempo de processamento (em segundos) de acordo com o raio ( $R$ ) para os <i>thresholds</i> estático ( $t = 0.75$ ) e adaptativo. . . . .	55
5.9	Exemplos de segmentações obtidas pela abordagem proposta, baseada em <i>superpixels</i> e detecção de comunidades em grafos. (a)-(c) são imagens originais e (d)-(f) são as respectivas segmentações com os valores de qualidade $I(S, S')$ . . . . .	56
5.10	A primeira coluna representa a imagem original, enquanto a segunda e a terceira representam as segmentações obtidas pela abordagem proposta e pelo método de Felzenwalb e Hunttenlocher [Felzenswalb and Huttenlocher 2004], respectivamente. O valor de $I(S, S')$ indica a qualidade da segmentação quantificada pelo método proposto na Seção 4.4. . . . .	58
6.1	Amostras de imagens de ninhais do Pantanal obtidas na expedição realizada em outubro de 2013. . . . .	61

6.2	Zoom digital de uma região da imagem apresentada na Figura 6.1(a). . . . .	61
6.3	Resultados da pré-segmentação de algumas imagens de ninhais do Pantanal utilizando o método proposto baseado em redes complexas e <i>superpixels</i> . . . . .	62
6.4	Resultados da pré-segmentação de algumas imagens de ninhais do Pantanal utilizando o método proposto baseado em redes complexas e <i>superpixels</i> . . . . .	63
6.5	Amostra dos resultados da aplicação da segmentação <i>EM/MPM</i> usando <i>MRF</i> em imagens de ninhais de aves. . . . .	65
6.6	Zoom de uma região da imagem apresentada na Figura 6.5(b). Aves identificadas pela cor cinza claro, marcada pelo círculo vermelho. . . . .	65
6.7	Imagen de um ninhal da Fazenda Sangradouro, no Pantanal. . . . .	66
6.8	Comparação dos resultados obtidos pela técnica <i>EM/MPM</i> com <i>MRF</i> , considerando a aplicação de zoom nas imagens. . . . .	67
6.9	Resultados das sub-imagens que representam a faixa superior da imagem apresentada na Figura 6.7. . . . .	69
6.10	Resultados das sub-imagens que representam a faixa central da imagem apresentada na Figura 6.7. . . . .	70
6.11	Resultados das sub-imagens que representam a faixa inferior da imagem apresentada na Figura 6.7. . . . .	71
6.12	Relação entre a contagem automática e a contagem manual. . . . .	71
6.13	Comparação da segmentação automática usando <i>MRF</i> com a segmentação manual obtida pela contagem das aves nas próprias imagens. Os valores entre parêntese representam a quantidade de aves. . . . .	72



# Lista de Tabelas

5.1	Parâmetros usados no processo de segmentação da imagem sintética e tempo computacional obtido, para os vários tamanhos dos <i>superpixels</i> (Tempo em segundos). . . . .	50
5.2	Tempo computacional (em segundos) para os experimentos cuja função de peso é baseada em textura. . . . .	52
6.1	Comparação da contagem <i>in loco</i> com a contagem automática das aves. . . . .	68



# Trabalhos Publicados

---

1. Cuadros, O.; BOTELHO, G. M.; Rodrigues, F.; BATISTA NETO, J. E. S. “Segmentation of Large Images with Complex Networks”. In: *XXV SIBGRAPI - Conference on Graphics, Patterns and Images*, 2012.
2. Cuadros, O.; BOTELHO, G. M.; Belizário, I. V.; Rodrigues, F.; BATISTA NETO, J. E. S. “Segmentation of large images based on super-pixels and community detection in graph”. In: *International Journal of Computer Vision*, 2014. Submetido para publicação.



# Introdução

## 1.1 Contextualização e Motivação

Um dos assuntos mais importantes da análise de imagens é a segmentação, cujo objetivo é identificar objetos e bordas nas imagens de acordo com a cor, forma ou textura [Fu and Mui 1981; Haralick and Shapiro 1985; Pal and Pal 1993; Zhang et al. 2008]. Esta tarefa tem várias aplicações práticas como, por exemplo, diagnóstico médico, recuperação de imagem baseada em conteúdo, observações astronômicas, rastreamento de segurança, monitoramento ambiental e outros [Costa and Cesar Parker 2010]. Como uma enorme quantidade de imagens são geradas destas aplicações, a extração de informação quantitativa destas imagens deve ser feita automaticamente, o que torna as técnicas de segmentação de imagens indispensáveis.

Atualmente, a possibilidade de fácil aquisição e armazenamento de imagens digitais em sistemas de baixo custo favoreceu o aumento da resolução das imagens. No entanto, o processamento destas imagens requer um aumento considerável no custo computacional. Logo, é necessário desenvolver técnicas de segmentação que lidem com imagens de alta resolução, mantendo, porém, o custo computacional o mais baixo possível. Por isso, este trabalho busca desenvolver uma nova abordagem de segmentação que combina algoritmos de detecção de comunidades, pertencentes à teoria das redes complexas, com técnicas de extração de *superpixels*, para segmentar grandes imagens mantendo o compromisso entre acurácia e custo computacional.

Uma das aplicações da metodologia de segmentação proposta será a análise de imagens de ninhais de aves do Pantanal mato-grossense, com o intuito de realizar o monitoramento biológico. Na prática, monitoramento biológico se refere ao uso de organismos para determinar a qualidade ambiental através da coleta de medidas e variáveis de processos ao longo do tempo [Cairns 1979]. Dessa forma, é possível diferenciar alterações naturais daquelas causadas

por poluição. Dentre os tipos de monitoramento, o de aves vem ganhando destaque devido à duas características principais: (1) as aves são mais facilmente visualizadas do que os demais integrantes da fauna e (2) são extremamente sensíveis à degradação ambiental. Assim, alterações no tamanho das populações de aves, quando detectadas, permitem registrar distúrbios eventuais que ocorrem no meio ambiente, inclusive aqueles decorrentes de mudanças climáticas [Lama et al. 2000].

O censo demográfico baseado na contagem da população é o parâmetro mais usado nos programas de monitoramento biológico, visto que a detecção precoce da redução da quantidade de indivíduos de uma população possibilita um melhor direcionamento das ações de manejo nos programas de conservação. A redução das populações naturais tem ocorrido por fatores como o isolamento das populações devido à diminuição e fragmentação dos seus ambientes. Este processo têm sido acelerado pelas mudanças climáticas, causando o declínio da biodiversidade global. Pesquisadores acreditam que alguns ecossistemas como, por exemplo, o Pantanal, terão dificuldades para se adaptar às mudanças climáticas [Loarie et al. 2009].

O Pantanal é um dos locais mais importantes para a reprodução de espécies de aves aquáticas no sul do continente americano, sendo considerado refúgio e área preferencial para o estabelecimento de ninhais e colônias deste tipo de aves. Estudos em populações de espécies chaves, distribuídas pelo Pantanal, podem ajudar a gerar respostas para as perturbações previstas no ecossistema. Neste contexto, destaca-se o **Projeto Monitoramento Biológico pelo Tamanho Populacional de Colônias de Aves Aquáticas de Topo de Cadeia Alimentar no Pantanal** (Proc. CNPQ n°563299/2010 – 0), cujo objetivo geral é a padronização de uma metodologia para monitorar populações das colônias reprodutivas da *Mycteria Americana* (ave popularmente conhecida como “cabeça-seca”) do Pantanal mato-grossense, baseada em estimativas do tamanho populacional e de tamanho efetivo da população.

Um grande interesse da equipe deste projeto é ter o monitoramento das colônias reprodutivas facilitado por um processo de contagem automática de ninhos/aves através de imagens aéreas, o que viria a atenuar as limitações inerentes do processo de contagem *in loco* sobre a colônia, diminuindo a intrusão no habitat natural das aves e reduzindo o tempo de aquisição dos dados. Com base nisso, o presente trabalho também visa contribuir com este projeto, aplicando a metodologia de segmentação proposta em imagens aéreas de ninhais do Pantanal, obtidas pelos integrantes do projeto citado, com o intuito de realizar a identificação das aves. Como as imagens de ninhais possuem alta resolução e apresentam uma característica peculiar, ou seja, a mesma cor (branca) para aves e os dejetos presentes no topo das árvores, conforme pode ser visto na Figura 1.1, é necessário tratar esta questão para garantir uma segmentação apropriada.

Para isso, propõe-se também a utilização da técnica de segmentação por textura baseada em



Figura 1.1: Amostras de imagens aéreas de ninhais de aves do Pantanal.

*Markov Random Fields (MRF)*. Tal técnica foi escolhida devido à sua capacidade de representar microtexturas (aves) adequadamente, o que permitirá diferenciar as aves das regiões com dejetos, possibilitando a segmentação e contagem da população. Na prática, será feita uma pré-segmentação das imagens de ninhais, usando a abordagem de segmentação baseada em redes complexas e *superpixels*, para eliminar áreas irrelevantes das imagens (lagos, estradas etc), seguida da segmentação final por *MRF*, permitindo a contagem automática das aves. Além disso, será feita uma comparação dos resultados obtidos automaticamente com os resultados obtidos pela contagem *in loco*, graças à parceria com os integrantes do **Projeto Monitoramento Biológico pelo Tamanho Populacional de Colônias de Aves Aquáticas de Topo de Cadeia Alimentar no Pantanal**.

## 1.2 Objetivos

O objetivo principal deste trabalho de doutorado é propor um método que permita a segmentação de imagens de alta resolução de forma precisa, com custo computacional reduzido. Tal método consiste numa nova abordagem de segmentação baseada na teoria das redes complexas (mais especificamente, algoritmos de detecção de comunidades) combinada com o uso de técnicas de extração de *Superpixels*. Além disso, propõe-se como aplicação principal a segmentação de imagens aéreas de ninhais de aves do Pantanal, permitindo a identificação automática das aves/ninhos. Para isso, o uso da técnica de segmentação por textura baseada em *Markov Random Fields* será considerada como complemento à abordagem de segmentação principal.

## 1.3 Contribuições Originais

Este trabalho traz as seguintes contribuições originais para a área de segmentação de imagens de alta resolução:

- Proposição de uma nova abordagem de segmentação que combina algoritmos de detecção de comunidades, pertencentes à teoria das redes complexas, com técnicas de extração de *superpixels*. Como a maioria dos algoritmos de detecção de comunidades apresentam custo computacional elevado, sua aplicação é limitada à imagens de baixa resolução. No entanto, com o uso das técnicas de extração de *superpixels*, a imagem passa a ser representada por um grafo de *superpixels*, e não de pixels, reduzindo a cardinalidade do grafo e, consequentemente, o custo computacional da aplicação de algoritmos de detecção de comunidades (seção 4.2).
- Amplo estudo para definição de valores adequados para os parâmetros do processo de extração de *superpixels* e geração do grafo, o qual representa a imagem. Neste contexto, destaca-se (1) a definição de novas funções de peso, responsáveis pelo estabelecimento das conexões entre os vértices do grafo, considerando as características de cor e textura dos *superpixels*, e (2) a proposição do *threshold adaptativo*, parâmetro fundamental do processo de geração do grafo, responsável pela eliminação do problema de vértices desconectados (seção 4.2).
- Desenvolvimento de uma nova métrica quantitativa, baseada em *ground-truth*, para avaliação da qualidade das imagens segmentadas. Tal métrica também serviu de base para a criação de um novo método de seleção de uma imagem de referência, dentre as diferentes imagens segmentadas manualmente pelos usuários, para ser comparada com a segmentação automática (seção 4.4).
- Desenvolvimento de um método de contagem automática de aves por meio da análise de imagens aéreas de ninhais usando técnicas de segmentação. Cabe ressaltar que a contagem automática ainda é um assunto pouco abordado na literatura [Balan 2003; Gerhardinger 2006].

## 1.4 Organização do Trabalho

Os demais capítulos desta tese estão organizados da seguinte forma:

- O Capítulo 2 descreve conceitos relacionados à segmentação de imagens digitais, destacando a segmentação baseada em textura, devido à sua aplicação nas imagens de ninhais. Duas técnicas focadas na identificação de microtexturas são detalhadas: segmentação por *Markov Random Fields* e por *Local Binary Pattern*.
- O Capítulo 3 aborda a teoria das redes complexas, relacionado-a com segmentação de imagens. Explica-se como é feita a representação da imagem como um grafo e apresentam-se diferentes técnicas de detecção de comunidades em grafos. Por fim, a técnica de extração de *superpixels* é detalhada. Tal técnica permite a redução da cardinalidade do grafo, o qual representa a imagem, por meio do agrupamento de pixels.
- O Capítulo 4 apresenta a abordagem proposta para a realização da segmentação de imagens de alta resolução, com aplicação na segmentação de imagens aéreas de ninhais de aves. Além disso, apresenta-se a nova métrica quantitativa de avaliação da qualidade das imagens segmentadas, juntamente com o método de seleção da imagem de referência.
- O Capítulo 5 apresenta os experimentos realizados, ilustrando os principais aspectos da abordagem de segmentação baseada em redes complexas e *superpixels*.
- O Capítulo 6 destaca os resultados da segmentação de imagens aéreas de ninhais de aves do Pantanal.
- Por fim, as considerações finais, as limitações e propostas de trabalhos futuros são relatadas no Capítulo 7.

# Capítulo 2

## Segmentação de Imagens baseada em Textura

### 2.1 Considerações Iniciais

A segmentação é uma das principais etapas em aplicações de análise de imagens e, geralmente, é a de maior complexidade computacional. Tem como objetivo a separação das regiões de uma imagem que são associadas aos objetos nela representados, considerando as propriedades de descontinuidade e similaridade destas regiões. Na literatura existem diversas abordagens para a segmentação de imagens, sendo normalmente classificadas em: limiarização, segmentação baseada em bordas e segmentação baseada em regiões [Fu and Mui 1981; Gonzalez and Woods 2010; Tripathi et al. 2012]. A partir desta classificação, diferentes técnicas foram desenvolvidas como, por exemplo, as técnicas baseadas em cores [Gonzalez et al. 1990; Lucchese and Mitra 2001] e em texturas [Reed and Hans 1993; Materka and Strzelecki 1998; Maenpaa 2003], redes neurais e lógica *fuzzy* [Bezdeck and Pal 1992; Karasulu and Balli 2010; Torbati et al. 2014], algoritmos genéticos [Ankenbrandt et al. 1990; Binh et al. 2012] e particionamento de grafos [Shi and Malik 2000; Xi 2010].

Considerando a aplicação final deste trabalho, ou seja, segmentação de imagens de ninhais de aves, duas importantes propriedades precisam ser analisadas: cor e textura. A cor será utilizada nas técnicas de segmentação baseada em algoritmos de detecção de comunidades, apresentados detalhadamente no Capítulo 3. Já a textura será abordada neste capítulo, com foco nas técnicas capazes de detectar microtexturas (aves). Inicialmente, é importante ressaltar que a definição de textura ainda não é um consenso para os pesquisadores, mas pode ser entendida como um padrão aleatório ou regular que se repete em uma determinada região ou objeto da imagem e que apresenta características como rugosidade,

regularidade, suavidade ou aspereza [Gonzalez and Woods 2010]. Existem diversos métodos de segmentação que utilizam características de textura, os quais podem ser classificados em [Reed and Hans 1993; Materka and Strzelecki 1998]: (1) estatísticos, (2) estruturais e (3) baseados em modelos.

Os métodos estatísticos analisam a distribuição espacial dos tons de cinza, calculando características locais em cada ponto da imagem e derivando um conjunto de estatísticas a partir destas distribuições. Já os métodos estruturais são usados para texturas de comportamento bem definido (determinísticos), não sendo adequados para a maioria das imagens do mundo real [Sonka 1999]. Por fim, os métodos baseados em modelos tem como vantagem a possibilidade de utilização em variados tipos de textura. Eles são classificados em modelos estocásticos e fractais [Reed and Hans 1993]. Ressalta-se que os modelos estocásticos vêm sendo cada vez mais estudados devido aos resultados bastante promissores e à ampla gama de aplicações.

Na segmentação baseada em modelos estocásticos, as classes da imagem são modeladas como campos aleatórios bidimensionais e o problema de segmentação é proposto como um problema de otimização estatística. Como exemplo deste tipo de técnica cita-se a segmentação baseada em *Markov Random Fields (MRF)* [Cross and Jain 1983], devido à sua capacidade de modelar microtexturas, visto que define a função de probabilidade do mapa de classes por meio de características puramente locais (características de vizinhança) e gera resultados satisfatórios [Balan 2003; Gerhardinger 2006]. Esta técnica será utilizada neste trabalho para realizar a segmentação final das imagens de ninhais de aves.

Outra técnica que também trabalha com características locais é a *Local Binary Pattern (LBP)* [Ojala et al. 2002; Qing et al. 2005]. Esta técnica emprega padrões binários locais invariantes à escala de cinza e à rotação da imagem para representar características de textura e, será utilizada para definir uma nova função de peso, a qual é responsável pelo estabelecimento das conexões entre as arestas do grafo que representa a imagem (detalhes na seção 4.2). Cabe ressaltar que, geralmente, as funções de peso são baseadas em cores ou intensidade de cinza e o uso da técnica *LBP* vai permitir analisar o processo de montagem do grafo por meio da extração de características de textura. Nas seções 2.2 e 2.3 são descritas, respectivamente, as técnicas de segmentação baseada em *Markov Random Fields* e em *Local Binary Pattern*.

## 2.2 Segmentação por *Markov Random Fields* - *MRF*

Na segmentação de imagens utilizando *MRF*, as classes dos pixels das imagens são modeladas usando campos aleatórios de *Markov* e o problema de segmentação é proposto como um problema de otimização estatística. Ressalta-se que este tipo de segmentação pode ser supervi-

sionada ou não-supervisionada. Na primeira abordagem, há uma etapa anterior à segmentação que visa obter os parâmetros do modelo a partir de um conjunto de dados de treinamento gerado de imagens já segmentadas de mesma natureza. No entanto, às vezes não é possível obter estes parâmetros ou pode-se desejar ter um sistema completamente autônomo. Nestes casos, recorre-se à segmentação não-supervisionada, onde os parâmetros são estimados durante a segmentação da imagem observada.

As abordagens não-supervisionadas são modeladas como um problema de dados incompletos (*incomplete data problem*), onde a imagem a ser segmentada são os dados observados; o resultado que se deseja obter, geralmente denominado mapa de classes ou campo de rótulos, são considerados os dados ausentes e os parâmetros do modelo constituem um ítem a ser determinado. Com isso, a tarefa de segmentação passa a ser uma tarefa de otimização estatística de um dado critério. A implementação desta abordagem segue um padrão comum dividido em duas partes: a) estimativa dos parâmetros e b) otimização estatística.

Para a estimativa de parâmetros os algoritmos iterativos conhecidos são: EM (*Expectation-Maximization*), SEM (*Stochastic Estimation-Maximization*) e ICE (*Iterative Conditional Estimation*). Para a etapa de otimização, as duas técnicas conhecidas são: MAP (*Maximization a Posterior*) e MPM (*Maximization of the Posterior Marginals*). Algumas combinações são possíveis com estes conjuntos de métodos. Um estudo sobre *ICE/MPM* é apresentado em [Giordana and Pieczynski 1997] e os resultados gerados pelo método *EM/MPM* podem ser encontrados em [Comer et al. 1996; Comer and Delp 2000]. Após considerar os resultados apresentados em diferentes trabalhos, optou-se pela combinação *EM/MPM*, a qual será apresentada com detalhes nas subseções seguintes.

### 2.2.1 Segmentação EM/MPM utilizando *MRF*

O método de segmentação *EM/MPM* foi proposto por Comer e Delp [Comer and Delp 2000] e combina o algoritmo *EM* para estimativa dos parâmetros da distribuição da imagem com o algoritmo *MPM* para otimização estatística do campo de rótulos (ou mapa de classes). A seguinte notação é utilizada na modelagem do problema:

- $X$  - campo de rótulos (ou mapa de classes)
- $Y$  - imagem observada
- $S$  - espaço 2D (látice retangular) onde  $X$  e  $Y$  estão definidos
- $N$  - número de elementos (pixels) em  $S$

- $X_s$  e  $Y_s$  - variáveis aleatórias de  $X$  e  $Y$  na posição  $s \in S$
- $L$  - número de rótulos supostamente conhecidos
- $x = (x_1, \dots, x_N)$  - realização de  $X = (X_1, \dots, X_N)$
- $y = (y_1, \dots, y_N)$  - realização de  $Y = (Y_1, \dots, Y_N)$
- $\Omega_X = \{x : x_1, \dots, x_N\}$  - espaço das possíveis realizações de  $X$ , onde  $x_s \in \{1, \dots, L\}$
- $\Omega_Y = \{y : y_1, \dots, y_N\}$  - espaço das possíveis realizações de  $Y$ , onde  $y_s \in \{0, \dots, 255\}$

Como em outros métodos estocásticos, a modelagem utilizada para a resolução do problema de segmentação permite a formulação como um problema de dados incompletos. Neste caso, a imagem original é o dado observado e a imagem segmentada (campo de rótulos) é o dado ausente. Segundo o teorema de *Bayes*, a probabilidade condicional  $p_{X|Y}(x|y, \theta)$  de ocorrer um determinado campo de rótulos dada a imagem original e o conjunto de parâmetros ( $\theta$ ) pode ser escrita como:

$$p_{X|Y}(x|y, \theta) = \frac{p_X(x) \times f_{Y|X}(y|x, \theta)}{f_Y(y|\theta)}, \quad (2.1)$$

onde a função  $p_{X|Y}$  é chamada de distribuição de probabilidade *a posteriori*, enquanto  $p_X$  é denominada distribuição de probabilidade *a priori*. Ressalta-se que  $p_{X|Y}$  é a probabilidade de ocorrer um campo de rótulos dada a imagem observada,  $p_X$  é a distribuição do campo de rótulos e  $f_{Y|X}$  é a probabilidade de ocorrer a imagem observada dado o campo de rótulos. O método de segmentação busca, portanto, encontrar uma estimativa do campo de rótulos  $x$  utilizando o método MPM, que tem como critério a minimização do número de pixels rotulados erroneamente. Como se trata de uma segmentação não-supervisionada, os parâmetros  $\theta$  do modelo são também desconhecidos, sendo necessária a utilização do algoritmo EM para estimá-los por máxima verossimilhança. Os modelos  $p_X$  (campo de rótulos) e  $f_{Y|X}$  (imagem observada), necessários para formulação da probabilidade *a posteriori*, são descritos nas subseções 2.2.2 e 2.2.3, respectivamente. Por fim, os algoritmos MPM e EM são apresentados nas subseções 2.2.4 e 2.2.5, respectivamente.

## 2.2.2 Modelo para o Campo de Rótulos

O campo de rótulos  $X$  é modelado por um campo aleatório de *Markov* (*Markov Random Fields* - *MRF*). Um campo aleatório é basicamente uma malha 2D formada por variáveis aleatórias. Neste caso as variáveis aleatórias são  $X_1, X_2, \dots, X_N$  e o valores que elas podem

assumir são os valores associados a cada classe  $1, 2, \dots, L$ . Para facilitar o entendimento deste modelo é necessário, primeiramente, observar o esquema de vizinhanças para o campo de *Markov* (Figura 2.1). Considerando o pixel  $x$ , a vizinhança 4-conectada é obtida tomando-se apenas os valores 1 da tabela, denominado campo de primeira ordem. Para um campo de ordem dois, os pixels enumerados 1 e 2 são escolhidos, e assim por diante. Para o método EM/MPM, em particular, geralmente utiliza-se um campo de ordem um.

5	4	3	4	5
4	2	1	2	4
3	1	x	1	3
4	2	1	2	4
5	4	3	4	5

Figura 2.1: Esquema de vizinhanças para os campos de *Markov*. A vizinhança 4-conectada é dada pelos pixels numerados com 1 e a vizinhança 8-conectada é representada pelos pixels numerados com 1 e 2.

Após entender o conceito de vizinhança, um *MRF* pode ser definido como um campo aleatório no qual a probabilidade de uma variável aleatória do campo assumir um valor qualquer, dados os valores de todas as outras variáveis deste campo, é igual à probabilidade desta mesma variável assumir um valor qualquer dado apenas seus vizinhos [Roshholm 1997]. Matematicamente, tem-se  $P(X_s = x_s | x_j, j \in S) = P(X_s = x_s | x_j, j \in G_s)$ , onde  $G_s$  é o conjunto de pixels vizinhos de  $s$ , dado um sistema de vizinhança pré-estabelecido.

Outro conceito importante, ainda relacionado com vizinhança, diz respeito aos **cliques**. Um clique é um conjunto de um ou mais pixels, onde cada pixel é vizinho de todos os demais do conjunto (dado um sistema de vizinhança). Por exemplo, na Figura 2.2 são ilustrados os cliques de um pixel em uma vizinhança 4 e 8-conectada, respectivamente. O conjunto de todos os possíveis cliques de uma imagem é representado por  $C$ .

O conceito fundamental a respeito dos campos de *Markov* está no teorema de Hammersley-Clifford [Besag 1974; Geman and Geman 1984], que diz que a probabilidade conjunta de um *MRF* é dada por  $P(X = x) = p_X(x)$ , onde  $p_X$  é uma distribuição de Gibbs. Uma distribuição de Gibbs relativa a  $S$  e  $G$  é a medida de probabilidade  $p_X$  definida em  $\Omega_X$  (espaço das possíveis realizações de  $X$ ) pela expressão:

$$p_X(x) = \frac{1}{Z} \exp \left\{ \frac{-U(x)}{T} \right\} \quad (2.2)$$

onde  $T$  representa a temperatura e  $Z$  é o fator de normalização dado por  $Z = \sum_{x \in \Omega_X} \exp \left\{ \frac{-U(x)}{T} \right\}$ .

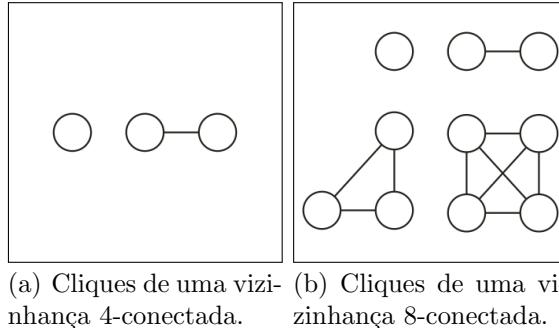


Figura 2.2: Exemplos de Cliques.

A função de energia  $U(x)$  é representada por  $U(x) = \sum_{c \in C} V_c(x)$ , onde  $V_c(x)$  é uma função que depende dos cliques, sendo a família  $\{V_c : c \in C\}$  chamada de potencial. No campo de rótulos, o conjunto  $C \subseteq S$  de pixels é um clique se, para quaisquer pixels  $r, s \in C$ ,  $s \in G_r$ , ou seja, todos são vizinhos entre si. Uma função de energia comumente empregada em estudos de segmentação de imagens [Lakshmanan and Derin 1989; Pappas 1992; Comer and Delp 2000] pode ser definida conforme a Equação 2.3:

$$U(x) = \sum_{\{r,s\} \in C} \beta t(x_r, x_s) + \sum_{\{r\} \in C} \gamma_{x_r} \quad (2.3)$$

onde

$$t(x_r, x_s) = \begin{cases} 0, & \text{se } x_r = x_s, \\ 1, & \text{se } x_r \neq x_s \end{cases} \quad (2.4)$$

Como este modelo usa a vizinhança 4-conectada, apenas os cliques duplos (dois pixels) e únicos (um pixel) são levados em consideração. O parâmetro  $\beta$ , chamado de **parâmetro de interação espacial**, define o peso da função de penalidade  $t(x_r, x_s)$  de um clique duplo. Esta função tem por objetivo desestimular pixels vizinhos a terem rótulos diferentes. Os parâmetros  $\{\gamma_k\}_{k=1,\dots,L}$ , que influenciam nos cliques únicos, podem ser considerados como os custos de ocorrência de cada rótulo  $k$ . Considerando as Equações (2.2) e (2.3) e temperatura constante  $T = 1$ , a probabilidade *a priori* do campo de rótulos pode ser escrita como:

$$p_X(x) = \frac{1}{Z} \exp \left( - \sum_{\{r,s\} \in C} \beta t(x_r, x_s) - \sum_{\{r\} \in C} \gamma_{x_r} \right) \quad (2.5)$$

### 2.2.3 Modelo para a Imagem Observada

Após obter o modelo para o campo de rótulos  $p_X(x)$ , é necessário obter o modelo  $f_{Y|X}$  da imagem observada  $Y$  condicionado no campo de rótulos. Para isso, assume-se que:

1. Inicialmente, as variáveis aleatórias  $Y_1, \dots, Y_N$  são condicionalmente independentes, dado o campo de rótulos  $X$ .
2. A função densidade de probabilidade condicional de  $Y_s$ , dado  $X$ , depende apenas do valor de  $X_s$ , ou seja, a intensidade do nível de cinza de um pixel depende apenas dos parâmetros do rótulo ao qual o mesmo pertence.

Desta forma, a função  $f_{Y|X}$  pode ser escrita da seguinte maneira:

$$f_{Y|X}(y|x, \theta) = \prod_{s=1}^N f_{Y_s|X, \theta}(y_s|x, \theta) = \prod_{s=1}^N f_{Y_s|X_s, \theta}(y_s|x_s, \theta) \quad (2.6)$$

A função  $f_{Y_s|X_r}$  assume a forma de uma distribuição normal (gaussiana), como em outros trabalhos sobre segmentação propostos anteriormente [Pappas 1992; Zhang et al. 2008]. Desta forma, para cada rótulo  $k = 1, \dots, L$ , existem dois parâmetros: média ( $\mu_k$ ) e variância ( $\sigma_k^2$ ). O vetor de parâmetros  $\theta$  possui, portanto, a média e a variância de cada rótulo, ou seja,  $\theta = [\mu_1, \sigma_1^2, \dots, \mu_L, \sigma_L^2]$ . Utilizando a Equação 2.7, a função densidade de probabilidade condicional de  $Y$ , dado  $X$ , também chamada de **função de verossimilhança**, pode ser escrita como:

$$f_{Y|X}(y|x, \theta) = \prod_{s=1}^N \frac{1}{\sqrt{2\pi\sigma_{x_s}^2}} \exp\left(-\frac{(y_s - \mu_{x_s})^2}{2\sigma_{x_s}^2}\right) \quad (2.7)$$

Dessa forma, a **probabilidade à posteriori**  $p_{X|Y}(x|y, \theta)$  será proporcional à probabilidade prévia  $p_X(x)$  vezes a função de verossimilhança  $f_{Y|X}(y|x, \theta)$ . Esta probabilidade é importante, pois servirá de base no processo de otimização estatística. Maximizar o valor desta função significa encontrar o mapa de classes  $X$  (ou imagem segmentada) mais provável, dada a imagem a ser segmentada  $Y$ . Em outras palavras, minimizar o número de pixels mal classificados em  $X$ .

### 2.2.4 Algoritmo MPM

Para descrever o funcionamento do algoritmo MPM (*Maximization of the Posterior Marginals*) assume-se um conjunto inicial de parâmetros  $\theta$ . Este algoritmo trata a segmentação como um problema de otimização, onde busca-se minimizar o valor esperado da quantidade de

pixels classificados incorretamente. Sabe-se que minimizar o valor esperado é igual a maximizar a distribuição marginal  $P(X_s = k|Y = y)$  para todo  $k \in 1, 2, \dots, L$  e para todo  $s \in S$ . Logo, para encontrar uma estimativa MPM de  $X$  é necessário encontrar para cada  $s \in S$ , o valor de  $k$  vizinhos que maximize a Equação 2.8, onde  $\Omega_{k,s} = \{x : x_s = k\}$ :

$$P(X_s = k|Y = y) = \sum_{x \in \Omega_{k,s}} p_{X|Y}(x|y, \theta) \quad (2.8)$$

Percebe-se que o cálculo da Equação 2.8 é inviável computacionalmente devido às inúmeras combinações possíveis para o mapa  $X$ , correspondentes ao espaço  $\Omega_{k,s}$ . Logo, é necessário obter uma aproximação adequada da estimativa do campo de *Markov*. Para isso, Marroquin et al. [Marroquin et al. 1987] propuseram um algoritmo para estimar a probabilidade  $P(X_s = k|Y = y)$  da seguinte forma:

- Usar um amostrador de Gibbs [Geman and Geman 1984] para gerar uma cadeia de *Markov*  $X(t)$ , discreta no tempo, que converja em uma distribuição para um campo aleatório com probabilidade  $p_{X|Y}(x|y, \theta)$  (Equação 2.1).
- Aproximar a densidade marginal  $p_{X_s|Y}(k|y, \theta)$ , que deve ser maximizada, como sendo a fração de tempo em que a cadeia de *Markov* passa no estado  $k$  no pixel  $s$ , para cada  $s$  e  $k$ .

Para gerar uma cadeia de *Markov*  $X(t)$ , usando o amostrador de Gibbs, tem-se um total de  $L^N$  estados possíveis, os quais correspondem aos elementos de  $\Omega_x$ . O algoritmo funciona da seguinte forma: a cada passo, um pixel é visitado, desta forma as cadeias  $X(t-1)$  e  $X(t)$  diferem apenas pela localização de um pixel. Em uma posição de tempo qualquer  $t$ , o estado de  $X(t)$  em um pixel  $s$  corresponde à variável aleatória  $X_s(t)$ . Considere  $q_t \in S$  como sendo o pixel visitado no tempo  $t$ . O estado de  $X_{q_t}(t)$  é determinado gerando-se um valor aleatório com distribuição de probabilidade condicional  $p_{X_{q_t}|Y, X_r, r \in G_{q_t}}(k|y, x_r(t-1), r \in G_{q_t}, \theta)$ . Se a sequência  $\{q_1, q_2, q_3, \dots\}$  contém cada pixel  $s \in S$  com frequência tendendo ao infinito, então para uma configuração  $x(0) \in \Omega_x$ , tem-se:

$$\lim_{t \rightarrow \infty} P(X(t) = x|Y = y, X(0) = x(0)) = p_{X|Y}(x|y, \theta) \quad (2.9)$$

para cada  $x \in \Omega$ . Assim, a cadeia de *Markov* converge em distribuição a um campo aleatório com probabilidade  $p_{X|Y}(x|y, \theta)$ , a qual é a distribuição limite para a cadeia de *Markov*. Para descrever a probabilidade condicional marginal  $p(X_s = k|Y = y, \theta)$  de cada pixel deve-se definir primeiramente a seguinte função:

$$u_{k,s}(t) = \begin{cases} 1, & \text{se } X_s(t) = k \\ 0, & \text{se } X_s(t) \neq k \end{cases} \quad (2.10)$$

Assim, se o amostrador de Gibbs faz  $T_s$  visitas ao pixel  $s$ , então a aproximação dada pela Equação 2.11 fornece as estimativas dos valores necessários para obter uma MPM de  $X$ .

$$p_{X_s,Y}(k|y, \theta) \approx \frac{1}{T_s} \sum_{t=1}^{T_s} u_{k,s}(t) \forall k, s \quad (2.11)$$

### 2.2.5 Algoritmo EM

O algoritmo EM (*Expectation-Maximization*) é um método de estimação de parâmetros que utiliza a função de máxima-verossimilhança e, geralmente, é aplicado em problemas de dados incompletos [Dempster et al. 1977; Jeff Wu 1983; Redner and Walker 1984]. Quando se trata de segmentação de imagens, o algoritmo EM é usado para estimar os parâmetros  $\theta$  do modelo da imagem observada  $Y$ , os quais são usados no processo de otimização do algoritmo MPM. Na prática, o algoritmo EM é um método iterativo de otimização que converge para um máximo local da função de verossimilhança. Em cada iteração, dois passos são realizados: determinação do valor esperado e maximização.

Considerando  $\theta(p)$  o conjunto de parâmetros estimados na iteração  $p$ , então o primeiro passo do algoritmo é calcular a seguinte função:

$$Q(\theta(p), \theta(p-1)) = E[\log f_{X|Y}(y|x, \theta)|Y = y, \theta(p-1) + \log p_X(x|\theta)|Y = y, \theta(p-1)] \quad (2.12)$$

Como a formulação da função de probabilidade de  $X$  não depende de  $\theta$ , exclui-se o segundo termo da Equação 2.12 e usa-se apenas o primeiro. Em seguida, a estimação de  $\theta(p)$  é obtido como sendo o valor  $\theta$  que maximiza  $Q(\theta, \theta(p-1))$ , satisfazendo a Equação 2.13:

$$Q(\theta(p), \theta(p-1)) \geq Q(\theta, \theta(p-1)) \forall \theta \in \Omega_\theta \quad (2.13)$$

Substituindo a Equação 2.7 em 2.12, diferenciando, igualando a zero e resolvendo para  $\theta(p) = [\mu_1, \sigma_1^2(p), ..., \mu_L(p), \sigma_L^2(p)]$  obtém-se as Equações 2.14 e 2.15:

$$\mu_k(p) = \frac{1}{N_k(p)} \sum_{s=1}^N y_{sp}(X_s = k|Y = y, \theta(p-1)) \quad (2.14)$$

$$\sigma_k^2(p) = \frac{1}{N_k(p)} \sum_{s=1}^N (y_s - \mu_k(p))^2 p(X_s = k | Y = y, \theta(p-1)) \quad (2.15)$$

onde  $N_k(p) = \sum_{s=1}^N p(X_s = k | Y = y, \theta(p-1))$  para  $k = 1, \dots, L$ . As equações 2.14 e 2.15 são utilizadas para obter iterativamente as atualizações dos parâmetros contidos no vetor  $\theta$ , ou seja, a média e a variância de cada classe.

## 2.2.6 Algoritmo EM/MPM

Para melhor compreensão do funcionamento do método *EM/MPM*, apresenta-se o seguinte algoritmo [Gerhardinger 2006]:

---

### Algorithm 1: EM/MPM

---

- 1: escolher número de classes  $L$
  - 2: gerar o mapa de classes  $X$  aleatório
  - 3: escolher  $\Theta(0)$
  - 4: **for all** iteração  $p = 1, \dots, P$  **do**
  - 5:    $v_{k,s} \leftarrow 0$  para todo pixel  $s$  e rótulo  $k$
  - 6:   executar o algoritmo MPM com  $\Theta(p-1)$ , obtendo  $v_{k,s}$  e  $X(p,t)$
  - 7:   executar o algoritmo EM com as estimativas  $v_{k,s}$ , obtendo  $\Theta(p)$
  - 8: **end for**
  - 9: a segmentação final é dada pelo último elemento de  $X(p,t)$
- 

Inicialmente, define-se o número de classes (Linha 1) e gera-se um mapa de classes com os valores das variáveis definidas aleatoriamente (Linha 2). Em seguida, deve ser feita a escolha do  $\Theta(0)$  (Linha 3), que consiste na estimativa inicial das variâncias  $\sigma_k^2$  e médias  $\mu_k$  para toda classe  $k = 1, \dots, L$ . O processo *EM/MPM* começa com o laço de iterações (Linha 4), cujo critério de parada é simplesmente o número  $P$  de iterações. Para cada iteração, inicializa-se o vetor  $v_{k,s} \leftarrow 0$  (Linha 5), que corresponde à probabilidade de cada pixel pertencer a determinada classe. Em seguida, executa-se o algoritmo *MPM* (Linha 6) considerando os parâmetros da iteração anterior  $\Theta(p-1)$ , obtendo-se novas estimativas  $v_{k,s}$  e a cadeia *markoviana*  $X(p,t)$ , a qual contém o mapa de classes. Executa-se também o algoritmo *EM* (Linha 7) com as novas estimativas  $v_{k,s}$ , obtendo o novo conjunto de parâmetros (médias e variâncias). Por fim, a segmentação final é dada pelo último elemento da cadeia *markoviana*  $X(p,t)$ .

## 2.3 Segmentação baseada em *Local Binary Pattern (LBP)*

O operador *Local Binary Pattern (LBP)* é uma medida de textura local invariante a iluminação e à rotação da imagem [Ojala et al. 2002; Maenpaa 2003]. O objetivo da técnica é obter um código binário que descreve a textura local em uma vizinhança circular de raio  $R$ , mediante o valor do nível de cinza do pixel central e dos  $P$  pixels vizinhos circularmente simétricos, conforme pode ser visto na Figura 2.3.

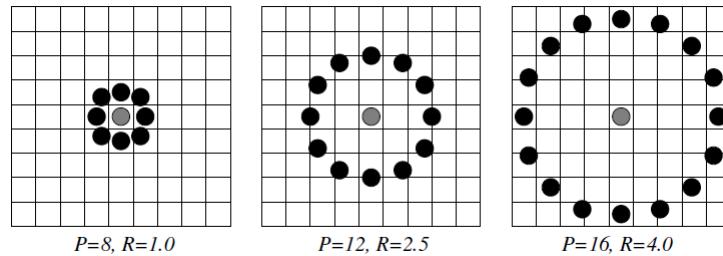


Figura 2.3: Conjunto de vizinhos circularmente simétricos para diferentes valores de  $(P, R)$ .

Para calcular o operador *LBP* para um pixel utiliza-se a Equação 2.16, onde  $g_c$  corresponde ao nível de cinza do pixel central e  $g_p$  ( $p = 0, \dots, P - 1$ ) corresponde aos valores de cinza de  $P$  pixels igualmente espaçados em um círculo de raio  $R$  ( $R > 0$ ) que formam um conjunto de vizinhos circularmente simétricos. Considerando o pixel central  $g_c$  com coordenadas  $(x_c, y_c)$ , as coordenadas dos vizinhos  $g_p$  são dadas por  $(x_c - R\sin(2\pi p/P), y_c + R\cos(2\pi p/P))$ . Ressalta-se que os valores dos vizinhos que não correspondem exatamente ao centro do pixel são estimados por interpolação.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \quad (2.16)$$

onde

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.17)$$

Percebe-se que ao atribuir o fator binomial  $2^p$  para cada sinal  $s(g_p - g_c)$  na Equação 2.16 tem-se um único número  $LBP_{P,R}$  que representa um padrão binário local, que por definição é invariante a qualquer transformação monotônica na escala de cinza, ou seja, contanto que a ordem dos valores de cinza na imagem permaneça a mesma, a saída do operador permanece constante.

Antes de aplicar o fator binomial  $2^p$ , o código  $LBP_{P,R}$  produz  $2^P$  padrões binários (0s e

1s) compostos por um número de  $P$  pixels. Ao rotacionar a imagem, os valores de cinza  $g_p$  correspondentes movem-se ao longo do perímetro do círculo ao redor de  $g_c$ , resultando em um valor diferente de  $LBP_{P,R}$ . Para remover o efeito da rotação, cada código  $LBP_{P,R}$  deve ser rotacionado para uma posição de referência, tornando iguais todas as versões rotacionadas de um código binário. Essa transformação é dada pela Equação 2.18.

$$LBP_{P,R}^{ri} = \min ROR(LBP_{P,R}, i) | i = 0, 1, \dots, P - 1 \quad (2.18)$$

onde o sobrescrito  $ri$  significa *rotation invariant* e  $ROR(x, i)$  executa um deslocamento bit-a-bit para direita sobre o número  $x$  de  $P$  bits  $i$  vezes.

A Figura 2.4 ilustra 36 padrões binários locais únicos, invariantes à rotação e a iluminação, que podem ocorrer no caso de  $P = 8$ , ou seja,  $LBP_{8,R}^{ri}$ . A partir destes padrões, Ojala [Ojala et al. 2002] observou que é possível derivar os **padrões uniformes**. Os padrões uniformes são responsáveis por cerca de 90% de todos os padrões presentes na textura de uma imagem e possuem como característica em comum uma estrutura circular uniforme que contém no máximo 2 transições espaciais (de 0 para 1 e de 1 para 0) no código binário circular, ou seja, a uniformidade  $U \leq 2$ .

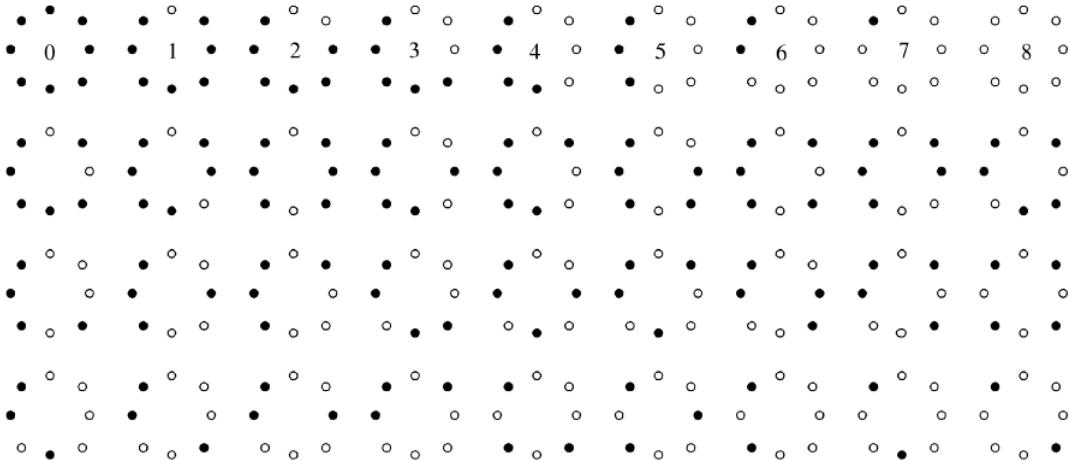


Figura 2.4: 36 padrões binários locais únicos invariantes à rotação referentes à  $LBP_{8,R}^{ri}$ . Os círculos preto e branco correspondem, respectivamente, ao valores de bit 0 e 1 em um padrão de 8 bits do operador. A primeira linha contém os nove **padrões uniformes** e os números dentro deles correspondem aos seus códigos *LBP* únicos. Figura adaptada de [Ojala et al. 2002].

Observando a primeira linha da Figura 2.4 percebe-se que existem  $P + 1$  padrões uniformes em um conjunto de vizinhos circularmente simétricos de  $P$  pixels. Tais padrões recebem um rótulo único correspondente ao número de bits 1 existentes no padrão. Já os padrões não uniformes são agrupados sob o rótulo heterogêneo  $P + 1$ . No final, a característica empregada na

análise da textura é um histograma com  $P + 2$  posições ( $P + 1$  posições contém os padrões uniformes e 1 posição contém todos os padrões não uniformes) dos padrões rotulados acumulados sobre uma amostra de textura. Ressalta-se que o histograma dos padrões uniformes discrimina melhor a textura que o histograma de todos os padrões individuais, pois a proporção relativa de padrões não uniformes de todos os padrões acumulados em um histograma é tão pequena que suas probabilidades não podem ser estimadas de forma confiável [Maenpaa 2003].

## 2.4 Considerações Finais

Este capítulo apresentou conceitos referentes à segmentação de imagens, focando na segmentação baseada em textura devido à sua importância para a realização deste trabalho. Foram apresentadas duas técnicas de segmentação por textura: segmentação *EM/MPM* usando *Markov Random Fields* e segmentação baseada em *Local Binary Pattern*. Estas técnicas foram escolhidas por utilizarem características locais (vizinhança) para analisar textura, sendo úteis na identificação de microtexturas contidas nas imagens. Ressalta-se que a técnica *EM/MPM* baseada em *Markov Random Fields* será utilizada na segmentação final das imagens de ninhais com o intuito de identificar automaticamente as aves para, posteriormente, contá-las. Já a técnica *Local Binary Pattern* será empregada no desenvolvimento de uma função de peso, responsável pela geração do grafo (que representa uma imagem) de *superpixels*, sobre o qual será aplicado os algoritmos de detecção de comunidades. Normalmente, as funções de peso são baseadas em características de cores ou intensidade de cinza e o uso da técnica *LBP* vai permitir analisar o processo de montagem do grafo por meio da extração de características de textura dos *superpixels*.

Capítulo **3**

# Redes Complexas e Segmentação de Imagens

## 3.1 Considerações Iniciais

Os métodos de segmentação de imagens baseados em grafos geralmente levam em conta algoritmos de particionamento espectral, os quais, geralmente não geram as melhores divisões em grupos de vértices e apresentam custo computacional elevado [Theodoridis and Koutroumbas 2003]. No entanto, com o desenvolvimento da teoria das redes complexas, técnicas de segmentação baseadas em grafos foram aprimoradas, ganhando destaque na literatura e alcançando resultados significativos [Angelini et al. 2007; Oliveira et al. 2008; Granell et al. 2011; Rodrigues et al. 2011]. Com a introdução da teoria das redes complexas, a identificação de grupos de vértices em grafos passou a ser feita por algoritmos de detecção de comunidades [Fortunato 2010]. Comunidades são grupos nos quais os vértices são mais intensamente conectados entre si do que com o restante da rede, conforme mostra a Figura 3.1. Estas técnicas de detecção de comunidades vem produzindo particionamentos acurados e com custo computacional satisfatório [Newman and Girvan 2004; Raghavan et al. 2007].

Como o agrupamento de dados é relacionado com segmentação de imagens, é possível usar as técnicas de detecção de comunidades para identificar os objetos nas imagens. Para isso, a imagem é mapeada em um grafo, permitindo que os algoritmos de detecção encontrem comunidades, as quais correspondem aos objetos das imagens. Existem inúmeros algoritmos para identificar comunidades. Alguns são relacionados com conceitos de *clustering* hierárquico e geralmente se baseiam na similaridade entre os vértices. Outros algoritmos não consideram medidas de similaridade como, por exemplo, os algoritmos espetrais, baseados na propagação de rótulos e nas medidas de *betweenness* e modularidade.

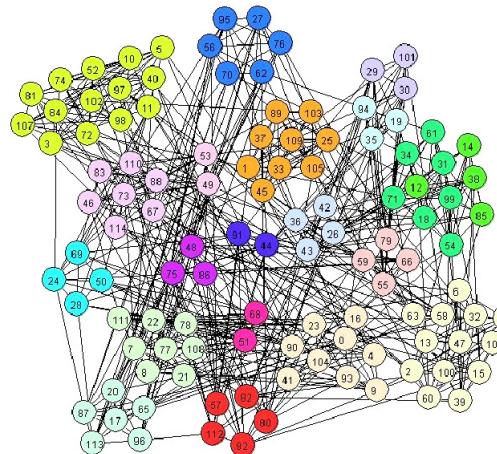


Figura 3.1: Exemplo de comunidades presentes em uma rede complexa. Figura adaptada de Newman [Newman and Girvan 2004].

Uma limitação fundamental da segmentação de imagens baseada em redes complexas é o tamanho da imagem usada. Por exemplo, uma imagem de tamanho  $N \times N$  é mapeada em um grafo com  $N^2$  vértices, onde cada vértice representa um pixel. Logo, apenas pequenas imagens podem ser consideradas na prática, visto que a maioria dos algoritmos de detecção de comunidades apresentam alto custo computacional. Para suprir esta limitação é necessário desenvolver formas de reduzir a cardinalidade do grafo que representa a imagem. Isso pode ser feito por meio de técnicas de agrupamento de pixels, chamadas de *Superpixels* [Ren and Malik 2003; Levinshtein et al. 2009; Achanta et al. 2010; Cigla and Alatan 2010].

Nas seções seguintes serão apresentados os detalhes envolvidos na utilização de redes complexas na segmentação de imagens digitais, começando pela representação da imagem como um grafo, a qual especifica as formas de estabelecer as arestas entre os vértices (seção 3.2). Em seguida, destacam-se alguns algoritmos de detecção de comunidades na seção 3.3. Por fim, na seção 3.4 é abordada a técnica de agrupamento de pixels *Superpixels*, capaz de reduzir a cardinalidade do grafo.

## 3.2 Representando a imagem como um grafo

Para permitir a segmentação de imagens por meio de algoritmos de detecção de comunidades é necessário transformar a imagem em um grafo, onde cada pixel da imagem pode ser representado como um vértice e as arestas são estabelecidas por meio de funções que traduzem alguma relação entre os pixels, chamadas de funções de peso ou similaridade. As seções 3.2.1 e 3.2.2 apresentam, respectivamente, os conceitos básicos sobre grafos e os mecanismos mais

comumente utilizados para representar imagens como grafos, detalhando algumas funções de peso.

### 3.2.1 Conceitos básicos sobre grafos

Um grafo  $G = (V, E)$  é definido por um conjunto finito não-vazio  $V$ , composto por elementos denominados **vértices**, e um conjunto  $E$  de elementos de  $V$ , denominadas **arestas**. Cada aresta de  $E$  é definida por um par  $e = \{u, v\}$ , sendo  $u$  e  $v$  pertencentes a  $V$ . A cardinalidade ou número de vértices e arestas de  $G$  é dado por  $|V|$  e  $|E|$ , respectivamente.

Grafos podem ser dirigidos ou não dirigidos, indicando, respectivamente, se existe ou não direção entre dois vértices. Podem-se também existir grafos ponderados e não ponderados. Em grafos ponderados, arestas entre os vértices  $u$  e  $v$  possuem um peso, que é determinado por uma função  $w(u, v)$ , que respeita as seguintes propriedades:  $w(u, v) > 0$  e  $w(u, v) = w(v, u)$ . Já grafos não ponderados possuem todos os pesos iguais a 1.

A partir dos pesos das arestas define-se a matriz de adjacência  $W_{uv}$ , que é uma matriz simétrica, quadrada, de ordem  $n$  ( $n = |V|$ ) e cujas entradas correspondem ao peso da aresta que liga os vértices  $u$  e  $v$ . Com base nesta matriz, pode-se obter o grau do vértice  $u$ , que é dado por  $\deg(u) = \sum_{v=1}^n W_{uv}$ . Ressalta-se que neste trabalho serão considerados grafos ponderados, não-dirigidos, sem laços (arestas ligando um vértice a ele mesmo) e sem arestas múltiplas (mais de uma aresta incidindo no mesmo par).

### 3.2.2 Formas de representar imagens como grafos

Imagens podem ser representadas como grafos, onde cada vértice tradicionalmente representa o valor do pixel da imagem e as arestas são calculadas de acordo com uma função de peso ou de similaridade  $W(i, j)$ , tal que  $i, j \in V$ . Tais funções devem traduzir algum relacionamento entre o par de pixels  $i$  e  $j$  e, geralmente, são baseadas na distância euclidiana, manhattan, gaussiana e outras. O cálculo destas funções considera dois parâmetros importantes: *threshold* ( $t$ ) e raio ( $R$ ). O valor  $t$  limita o número de conexões criadas verificando se o peso da aresta é menor ou igual a um *threshold*. Já o raio define uma região circular de raio  $R$ , onde as conexões são estabelecidas, evitando que pixels distantes sejam adicionados. O *threshold* pode variar de acordo com a similaridade da intensidade do pixels e o raio varia de acordo com o tamanho da imagem e a proximidade de regiões.

Um exemplo clássico de função de peso é mostrado na Equação 3.1 [Shi and Malik 2000]. Tal função é baseada na distância gaussiana, onde  $I$  corresponde a imagem,  $X_{(i)}$  e  $X_{(j)}$  representam a localização espacial dos pixels  $i$  e  $j$ ,  $\|X(i) - X(j)\|_2$  é a distância euclidiana entre cada par

de pixels,  $r$  define a quantidade de vizinhos para cada pixel,  $F_{(i)}$  é o vetor de características correspondente a cada pixel e  $\sigma$  é um valor ajustado entre 10% e 20% do intervalo total das distâncias entre os vetores de características.

$$W_{i,j} = e^{\frac{-||F_{(i)} - F_{(j)}||_2^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-||X_{(i)} - X_{(j)}||_2^2}{\sigma_X^2}}, & \text{if } ||X(i) - X(j)||_2 < r; \\ 0, & \text{caso contrário.} \end{cases} \quad (3.1)$$

Algumas funções de peso consideram exclusivamente se a imagem é colorida ou em tons de cinza. Por exemplo, a função de peso dada pela Equação 3.2 é aplicada em imagens em tons de cinza, onde  $I_i$  representa o nível de cinza do vértice  $i$ , os pesos são definidos no intervalo  $[0, 1]$  com 0 representando ausência de conexão e 1 representando conexão máxima entre dois vértices e  $t$  corresponde à um limiar responsável por limitar o número de conexões.

$$W_{i,j} = 1 - |I_i - I_j| \quad (3.2)$$

Já a função mostrada na Equação 3.3 é aplicada em imagens coloridas no modelo de cor RGB. Esta função corresponde à distância euclidiana normalizada dos três canais de cores (vermelho, verde e azul) dos pixels  $i$  e  $j$ .

$$W_{i,j} = 1 - \frac{\sqrt{(R_i - R_j)^2 + (G_i - G_j)^2 + (B_i - B_j)^2}}{\sqrt{3}} \quad (3.3)$$

As funções 3.2 e 3.3 foram propostas neste trabalho para lidarem com grafos de *superpixels* (grupos de pixels), e não de pixels isolados. Além destas, novas funções de peso, baseadas no modelo de cor CIELAB e em características de textura, foram propostas. Tais funções serão apresentadas com detalhes na seção 4.2.

### 3.3 Abordagens para a detecção de comunidades

O conceito de estrutura de comunidades é muito importante para permitir o desenvolvimento de algoritmos de detecção, no entanto, sua definição não é um consenso. Intuitivamente, acredita-se que deve haver mais arestas dentro da comunidade que arestas ligando vértices da comunidade com o resto do grafo. Radicchi [Radicchi et al. 2004] definiu comunidades no sentido forte e fraco, com base na comparação da densidade das arestas entre os vértices. No sentido forte, um subgrafo é uma comunidade se todos os seus vértices tem mais conexões entre eles que com o restante dos vértices da rede. Já no sentido fraco, um subgrafo é uma comunidade se a soma de todos os graus dos vértices pertencentes ao subgrafo é maior que

daqueles que estão fora do subgrafo.

Outro problema fundamental relacionado à detecção de comunidades é como definir a melhor divisão da rede em suas comunidades constituintes, visto que em redes reais geralmente nenhuma informação está disponível sobre o número e tamanho das comunidades existentes. Com o objetivo de resolver este problema, Newman [Newman 2004] propôs uma medida, chamada de **Modularidade (Q)**, que mede a qualidade de uma divisão particular da rede. Com base nesta medida, pode-se estimar a qualidade do particionamento obtido por um determinado algoritmo de detecção de comunidades.

Nas subseções seguintes serão descritos alguns algoritmos de detecção de comunidades, desde aqueles relacionados aos conceitos de *clustering* hierárquico (métodos divisivos e aglomerativos), passando pelos algoritmos que utilizam a medida de modularidade e, finalizando com um algoritmo de propagação de rótulos, o qual apresenta tempo de execução quase linear.

### 3.3.1 Algoritmos baseados em *Clustering* Hierárquico

Geralmente pouco se sabe sobre a estrutura de comunidades das redes complexas, porém determinadas redes possuem uma estrutura hierárquica, ou seja, apresentam vários níveis de agrupamento de vértices. Nestes casos, algoritmos de *clustering* hierárquico são bastante úteis para identificar grupos de vértices com alta similaridade, descobrindo divisões naturais da rede [Fortunato 2010]. Os métodos de *clustering* hierárquico podem ser divididos em divisivos e aglomerativos, os quais são apresentados a seguir.

#### Algoritmos Divisivos

Os algoritmos divisivos buscam encontrar as arestas que conectam as diferentes comunidades para, em seguida, removê-las iterativamente, dividindo a rede em grupos desconexos de vértices, ou seja, inicia-se com um único grafo contendo todos os vértices e procede-se dividindo este grafo em subgrafos cada vez menores. Um algoritmo divisivo bem conhecido foi proposto em 2002 por Girvan e Newman [Girvan and Newman 2002] e utiliza o conceito de *Betweenness*.

*Betweenness* é uma medida usada para identificar arestas que conectam comunidades, apresentando valores altos para estas arestas e penalizando arestas que conectam vértices pertencentes a mesma comunidade. Para calcular o valor de *Betweenness* é necessário obter o caminho mínimo entre dois vértices, conforme apresentado na Equação 3.4:

$$B_u = \sum_{ij} \frac{\sigma(i, u, j)}{\sigma(i, j)} \quad (3.4)$$

onde  $\sigma(i, u, j)$  é o número de caminhos mínimos entre os vértices  $i$  e  $j$  que passam pelo vértice ou aresta  $u$ ,  $\sigma(i, j)$  é o número total de caminhos mínimos entre  $i$  e  $j$  e o somatório se aplica a todos os pares  $i$  e  $j$  de vértices distintos.

Segundo o algoritmo de Girvan e Newman [Girvan and Newman 2002], as arestas com alto valor de *Betweenness* serão removidas iterativamente. Depois de remover cada aresta, o *Betweenness* das arestas remanescentes deve ser recalculado. Com isso, a principal desvantagem do algoritmo é seu custo computacional. Em um grafo com  $m$  arestas e  $n$  vértices, a complexidade total para o cálculo do *Betweenness* é  $O(m^2n)$ .

### Algoritmos Aglomerativos

Vértices pertencentes a mesma comunidade tendem a apresentar características semelhantes. Baseada nesta premissa é possível obter comunidades considerando a similaridade entre os vértices. Para isso, os métodos aglomerativos iniciam com todos os vértices desconectados e aplicam algum critério de similaridade para, progressivamente, uni-los e obter as comunidades.

Para avaliar a similaridade associada com a aresta  $(i, j)$  é possível usar várias medidas como, por exemplo, a distância euclidiana e o coeficiente de *Pearson*, representadas, respectivamente, pelas Equações 3.5 e 3.6:

$$DE_{i,j} = \sqrt{\sum_{k \neq i,j} (a_{ik} - a_{jk})^2} \quad (3.5)$$

$$CP_{i,j} = \frac{\frac{1}{n} \sum_k (a_{ik} - \mu_i)(a_{jk} - \mu_j)}{\sigma_i \sigma_j} \quad (3.6)$$

onde  $\mu_i = \frac{1}{n} \sum_j a_{ij}$  e  $\sigma_i = \frac{1}{n-1} \sum_j (a_{ij} - \mu_i)^2$ . Embora o cálculo destas medidas seja computacionalmente rápido, geralmente elas não produzem uma divisão correta da rede em comunidades, nem mesmo quando a estrutura das comunidades já é previamente conhecida. Além disso, elas tendem a encontrar somente os vértices centrais das comunidades (possuem similaridade alta) e deixar de fora os vértices periféricos (possuem menor similaridade) [Newman and Girvan 2004]. Diante disso, uma medida mais usada para identificar comunidades é a modularidade, pois apresenta melhores resultados, porém com custo computacional maior. Esta medida será apresentada na seção 3.3.2.

#### 3.3.2 Algoritmos baseados na medida de Modularidade

A medida de modularidade ( $Q$ ) foi proposta por Newman [Newman 2004] com o intuito de estimar a qualidade de uma divisão particular da rede em comunidades. Tal medida é dada pela

Equação 3.7, onde  $e_{ii}$  é o número de arestas da rede que estão inseridas dentro da comunidade  $i$ , e  $a_i^2$  é este mesmo número, porém considerando que as arestas são tomadas aleatoriamente.

$$Q = \sum_i (e_{ii} - a_i^2) \quad (3.7)$$

De acordo com a medida de modularidade, valores muito próximos de 0 indicam baixa probabilidade da rede estar dividida em comunidades reais, visto que a chance de tais agrupamentos serem propósitos não difere da casualidade de sua formação. Neste sentido, observa-se que quanto maior o valor de  $Q$ , maior a chance de que tais agrupamentos não existam apenas ao acaso (sua presença está, de alguma forma, intrínseca à estrutura e semântica da rede).

Da forma como foi originalmente definida,  $Q$  envolve processos de buscas e divisões iterativas de alto custo computacional, uma vez que deveria calcular  $Q$  para todos os possíveis particionamentos da rede em comunidades. Além disso, provou-se que a maximização da modularidade é um problema NP-Completo [Brandes et al. 2006], logo é provavelmente impossível encontrar uma solução em um tempo razoável, visto que este cresce polinomialmente com o tamanho do grafo. Diante disso, surgiram vários algoritmos capazes de encontrar boas aproximações do valor de modularidade máximo em um tempo razoável. Dentre eles cita-se:

- Otimização Gulosa: Newman [Newman 2004] adotou um algoritmo aglomerativo de otimização **gulosa** para encontrar o melhor valor da modularidade. Partindo-se de um estado no qual cada vértice da rede representa uma comunidade, comunidades são conectadas duas a duas, repetidamente, até que a conexão que resultar no maior valor de  $Q$  seja selecionada. O algoritmo segue até que toda a rede seja considerada uma única comunidade. A variação em  $Q$  após a conexão entre duas comunidades  $i$  e  $j$  pode ser medida da seguinte forma:

$$\Delta Q_{ij} = 2(e_{ij} - a_i a_j) \quad (3.8)$$

onde  $e_{ij}$  é o número de arestas que conectam a comunidade  $i$  à comunidade  $j$ ,  $a_i$  é número total de arestas que conectam a comunidade  $i$  às demais comunidades da rede e pode ser calculada por  $a_i = \sum_k e_{ik}$ , assim como  $a_j$  é o número total de arestas que conectam a comunidade  $j$  às demais comunidades da rede e pode ser calculada da mesma forma que  $a_i$ . Todo processo geralmente é representado na forma de um **dendograma**, ou seja, uma árvore que exibe a ordem das conexões, conforme pode ser observado na Figura 3.2. A linha tracejada representa a partição com maior valor de modularidade encontrado.

A complexidade deste algoritmo é  $O((m+n)n)$  ou  $O(n^2)$  para grafos esparsos, onde  $n$  é a

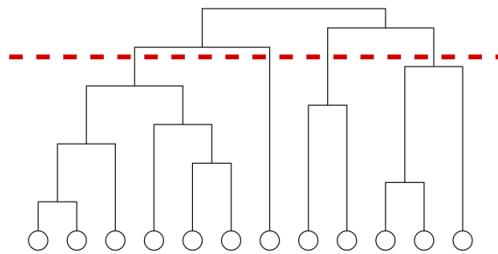


Figura 3.2: Dendrograma que descreve a rede. Linha tracejada representa a partição com maior valor de modularidade.

quantidade de vértices e  $m$  é a quantidade de arestas da rede. Clauset [Clauset et al. 2004] propôs melhorias nas estruturas de dados deste algoritmo, obtendo custo computacional de  $O(md(\log n))$ , onde  $d$  corresponde à profundidade do dendrograma que descreve a rede. Para redes esparsas e hierárquicas, onde  $m \sim n$  e  $d \sim \log n$ , o algoritmo de Clauset executa em  $O(n \log^2 n)$ , custo computacional inferior à implementação tradicional. Ressalta-se que o algoritmo de Clauset, o qual é chamado de *Fast Greedy*, é usado nos experimentos realizados neste trabalho.

- *Simulated Annealing*: é um procedimento probabilístico que procura pelo ótimo global de uma função  $F$  através da exploração do espaço de possíveis estados. Transições de um estado para outro ocorrem com probabilidade 1, se  $F$  aumentar depois de uma mudança, caso contrário, ocorrem com probabilidade  $\exp(\beta\Delta F)$ , onde  $\Delta F$  é o decremento da função e  $\beta$  é um índice de ruído estocástico (um tipo de temperatura inversa) que aumenta após cada iteração e reduz o risco do sistema ficar preso em um ótimo local. O sistema converge, em algum momento, para um estado estável, cuja solução dependerá de quantos estados foram explorados e de quanto devagar  $\beta$  variou.

Guimerà [Guimerà et al. 2004] empregou o *Simulated Annealing* para otimizar a busca pelo melhor valor de modularidade. Ele usou dois tipos de movimento: local e global. No movimento local, um único vértice selecionado aleatoriamente é deslocado de uma comunidade para outra. Já os movimentos globais consistem de junções e divisões das comunidades. Para obter a melhor performance, deve-se otimizar o valor da modularidade de uma bipartição da comunidade, considerada como um grafo isolado. Para isso, considera-se apenas movimentos de vértices individuais e a temperatura é reduzida até atingir o valor adequado para a otimização global.

Movimentos globais reduzem o risco de atingir ótimos locais e resultam em soluções melhores. Geralmente, é necessário combinar movimentos locais e globais em cada iteração. O uso do *Simulated Annealing* para otimizar a modularidade pode convergir para soluções

ótimas, porém ele é um método muito lento, sendo empregado apenas em grafos pequenos.

- Otimização Extrema: Duch e Arenas [Duch and Arenas 2005] propuseram um algoritmo de divisão que otimiza a modularidade  $Q$  por meio de uma busca baseada no algoritmo de Otimização Extrema (EO), desenvolvido por Boettcher [Boettcher 2001]. Tal algoritmo funciona por meio da otimização de uma variável global (modularidade) através do melhoramento de variáveis locais. Como a medida de modularidade pode ser escrita como um somatório sobre os vértices, a modularidade local de um vértice é o valor do termo correspondente neste somatório. Dessa forma, a medida de adequação (*fitness*) para cada vértice pode ser obtida dividindo a modularidade local do vértice pelo seu grau, o que a torna normalizada e independente do grau do vértice.

Baseado na medida de *fitness* de cada vértice, a qual corresponde à variável local envolvida na otimização extrema, o processo de busca heurística para encontrar o valor de modularidade ótimo consiste em, inicialmente, dividir aleatoriamente a rede em duas partições com o mesmo número de vértices (os componentes conectados em cada partição serão entendidos como comunidades). A cada passo o sistema se auto-organiza, movimentando o vértice com menor *fitness* (extremo) de uma partição para outra. O processo é repetido até que o valor máximo de  $Q$  seja obtido. Depois disso, eliminam-se todas as arestas entre ambas as partições e procede-se, recursivamente, com todos os componentes conectados resultantes. O processo termina quando a modularidade  $Q$  não puder ser melhorada.

No procedimento descrito, o vértice selecionado é aquele com pior *fitness*. No entanto, este procedimento é dependente da divisão inicial da rede e há forte possibilidade de ficar preso em ótimos locais. Melhores resultados podem ser obtidos usando uma seleção probabilística, chamada  $\tau - EO$  [Boettcher and Percus 2001], na qual os vértices são ordenados de acordo com seus valores de *fitness*, e então o vértice  $i$  do *rank* é selecionado de acordo com a distribuição de probabilidade  $P(i) \propto i^{-\tau}$ . O custo computacional de todo o processo é  $O(n^2 \ln^2(n))$ , onde  $n \ln(n)$  é o custo do processo de ordenação. Não é um algoritmo rápido, mas os resultados obtidos apresentaram altos valores de modularidade e precisão na determinação da estrutura das comunidades.

- Otimização Espectral: existem diferentes métodos espetrais, dentre eles tem-se o método proposto por Newman [Newman 2006] que reformula o conceito de modularidade em termos dos autovetores de uma nova matriz que caracteriza a rede, chamada de matriz de modularidade. Para cada subgrafo  $g$ , sua matriz de modularidade  $B^{(g)}$ , para os vértices  $i$  e  $j$  pertencentes a  $g$ , tem seus elementos dados por:

$$b_{ij}^{(g)} = a_{ij} - \frac{k_i k_j}{2M} - \delta_{ij} \sum_{u \in g} [a_{iu} - \frac{k_i k_u}{2M}] \quad (3.9)$$

onde  $a_{ij}$  corresponde ao número de arestas entre  $i$  e  $j$  (as medidas  $a_{ij}$  são os elementos da matriz de adjacência),  $\frac{k_i k_j}{2M}$  é o número esperado de arestas entre os vértices  $i$  e  $j$  caso tenham sido colocadas aleatoriamente ( $k_i$  e  $k_j$  representam o grau dos vértices  $i$  e  $j$ , respectivamente) e  $\delta_{ij} = v_i^T v_j$ , onde  $v$  representa o autovetor.

Para dividir a rede em comunidades, considerando a matriz de modularidade  $B^{(g)}$ , deve-se encontrar seu autovalor mais positivo com seu correspondente autovetor. De acordo com os sinais dos elementos do autovetor, a rede é dividida em duas partes: vértices com elementos positivos são atribuídos a uma comunidade e vértices com elementos negativos são atribuídos a outra. Em seguida, o processo é repetido recursivamente para cada comunidade até obter uma divisão da rede onde zero ou uma contribuição negativa para a modularidade total seja encontrada [Newman 2006].

### 3.3.3 Algoritmos baseados na Propagação de Rótulos

Raghavan [Raghavan et al. 2007] propôs um método de detecção de comunidades baseado na propagação de rótulos, chamado *Label Propagation*. Inicialmente, cada vértice recebe um rótulo diferente. A cada iteração, todos os vértices são varridos em ordem sequencial e cada vértice recebe o rótulo da maioria de seus vizinhos. Se não há um único rótulo majoritário, um dos rótulos majoritários é selecionado aleatoriamente. Desta maneira, os rótulos vão propagando através do grafo, onde a maioria dos rótulos desaparece e alguns predominam. O processo atinge a convergência quando cada vértice possui a maioria dos rótulos de seus vizinhos. A Figura 3.3 mostra a propagação de rótulos executada pelo algoritmo.

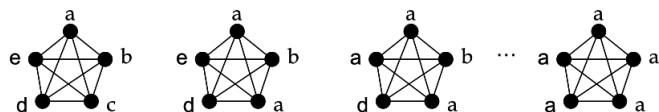


Figura 3.3: Vértices são atualizados um a um, da esquerda para a direita. Devido a alta densidade das arestas (a maior neste caso), todos os vértices adquirem o mesmo rótulo.

Este algoritmo não resulta em uma única solução devido à seleção aleatória de rótulos, quando não existe um único rótulo majoritário entre os vizinhos de um vértice. Logo, é possível derivar diferentes partições a partir da mesma condição inicial. Para permitir a obtenção de informações das diferentes partições, os autores propuseram a agregação de partições, a qual

permite identificar comunidades sobrepostas. A complexidade de tempo de cada iteração é  $O(n)$  ( $n$  é o número de vértices) e o número de iterações para atingir a convergência parece independente do tamanho do grafo ou cresce muito lentamente com ele.

Esta é uma das técnicas mais rápidas de detecção de comunidades, podendo ser usada para analisar grandes redes. No entanto, sabe-se que a maioria das técnicas apresentam alto custo computacional, limitando sua aplicação a redes pequenas. Diante disso, torna-se necessária a redução da cardinalidade do grafo, o qual representa a imagem. Uma possível solução é o uso de técnicas de extração de *Superpixels*, que serão explicadas na seção 3.4, as quais realizam o agrupamento dos pixels da imagem, gerando um grafo com um número menor de vértices.

## 3.4 Agrupamento de Pixels - *Superpixels*

*Superpixel* é uma abordagem de segmentação de imagens baseada em região que consiste em particionar a imagem em vários agrupamentos de pixels. Esta técnica pode ser usada como uma pré-segmentação para reduzir a quantidade de pixels da imagem ou como segmentação final. Existem vários trabalhos na literatura [Ren and Malik 2003; Levinstein et al. 2009; Achanta et al. 2010; Cigla and Alatan 2010] que tratam da extração de *superpixels*. Dentre eles, os algoritmos de Achanta [Achanta et al. 2010] e Cigla [Cigla and Alatan 2010] são considerados mais eficientes computacionalmente. Estes algoritmos são explicados detalhadamente nas subseções 3.4.1 e 3.4.2.

### 3.4.1 Simple Linear Iterative Clustering (SLIC)

Achanta et. al. [Achanta et al. 2010; Achanta et al. 2012] introduziram o algoritmo SLIC para extração de *superpixels* em imagens coloridas, considerando o modelo de cores CIELAB. Esta abordagem é baseada no algoritmo *k-means*, mas emprega um espaço de busca mais eficiente, visto que seu tamanho é limitado, reduzindo o custo computacional. Três passos descrevem a abordagem SLIC:

- Inicialmente, uma imagem de entrada de tamanho  $N \times N$  é dividida em  $k$  regiões retangulares, ou seja, *superpixels* de dimensão  $s \times s$ , onde  $s = \frac{N}{k}$ . Cada *superpixel*  $j$  é representado por um centróide definido como um vetor de cinco dimensões  $C_j = L_j, a_j, b_j, x_j, y_j$ , onde  $L$ ,  $a$  e  $b$  são os valores médios dos três componentes do modelo CIELAB, ou seja, a luminância  $L$  (a qual varia de 0 (preto) a 100 (branco)) e dois intervalos de cores  $a$  e  $b$  que variam, respectivamente, do verde ao vermelho e do azul ao amarelo. Já  $x_j$  e  $y_j$  são as coordenadas do *superpixel*  $j$ .

- Em cada iteração, cada pixel  $i$  é associado com o centróide do *superpixel* mais próximo, dado que a área de busca de cada *superpixel* inclui  $i$ . O tamanho desta área de busca é  $2s \times 2s$ . Em seguida, novos centróides são calculados como uma média do vetor  $C_j$ , considerando todos os pixels do *superpixel*  $j$ . Este processo é executado através da minimização da função  $D$ , definida pela Equação 3.10:

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2}, \quad (3.10)$$

onde  $d_c$  é a distância no espaço de cores e  $d_s$  é a distância espacial, representadas, respectivamente, pelas Equações 3.11 e 3.12:

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}. \quad (3.11)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}. \quad (3.12)$$

Os autores [Achanta et al. 2010] sugerem que *superpixels* acurados podem ser obtidos com no máximo 10 iterações. Já o parâmetro  $m$  ( $1 \leq m \leq 20$ ) controla a compacidade dos *superpixels*, ou seja, quanto maior o valor de  $m$ , maior a ênfase na proximidade espacial e mais compactos os grupos [Achanta et al. 2010].

- Finalmente, grupos de pixels isolados podem surgir após o término das iterações. Estes grupos devem ser associados ao maior *superpixel* vizinho.

### 3.4.2 Speeded-Up TurboPixel

Cigla e Alatan [Cigla and Alatan 2010] sugeriram um método para extração de *superpixels* quase uniformes. Tal método é eficiente em termos de custo computacional, compacidade de segmentos e baixo erro de super-segmentação. Para extrair os *superpixels*, são realizados três passos principais, os quais são apresentados na Figura 3.4 e explicados a seguir:

- Passo inicial: a imagem é dividida em regiões retangulares (*superpixels*), chamadas de segmentos, de acordo com o número de pixels desejado.
- Passo intermediário: os pixels sobre as bordas dos segmentos são testados e atribuídos a novos segmentos por meio da minimização da função de custo apresentada na Equação 3.13:

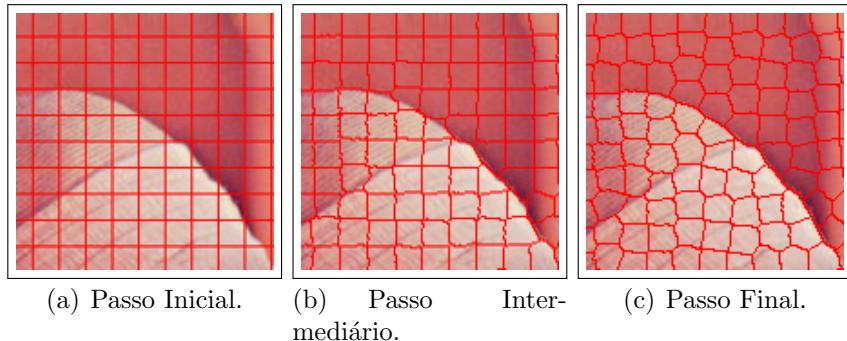


Figura 3.4: Pixels localizados nas bordas dos segmentos são atualizados em cada execução.

$$Custo_{x,y}(i) = \lambda_1 \cdot |I(x,y) - I_i| + \lambda_2 \cdot |(x - C_x^i)^2 + (y - C_y^i)^2| \quad (3.13)$$

onde  $I_i$  indica a intensidade média do  $i$ -ésimo segmento,  $x$  e  $y$  são as posições dos pixels testados entre os diferentes segmentos,  $C_x^i$  e  $C_y^i$  são as posições do centróide do  $i$ -ésimo segmento e  $\lambda_1$  e  $\lambda_2$  correspondem, respectivamente, ao coeficiente de similaridade e de restrição da convexidade. O primeiro termo da Equação 3.13 garante similaridade de intensidade entre os pixels que serão incorporados, enquanto o segundo termo garante que os *superpixels* terão formas convexas, impedindo que pixels distantes sejam incorporados.

- Passo final: é estabelecido um limite máximo de permutações dos pixels, como critério de parada do algoritmo. Ao finalizar as permutações, as bordas dos *superpixels* devem se ajustar às bordas dos objetos da imagem.

## 3.5 Considerações Finais

Neste capítulo foram apresentados alguns conceitos referentes às redes complexas, e a relação das mesmas com segmentação de imagens. Foram descritas algumas formas de representação de imagens como grafos, considerando diferentes funções de peso. Além disso, diversos algoritmos de deteção de comunidades foram apresentados: baseados em *clustering* hierárquico, medida de *betweenness*, modularidade e propagação de rótulos. Dentre eles, considerou-se o *Fast Greedy* (algoritmo baseado na otimização gulosa da medida de modularidade) e o *Label Propagation* (algoritmo baseado na propagação de rótulos) para a aplicação em segmentação de imagens. O primeiro devido ao seu menor custo, em comparação com outros algoritmos também baseados na medida de modularidade, e à sua capacidade de definir a melhor divisão da rede em comunidades. O segundo devido ao seu custo computacional quase linear.

No entanto, após a realização de experimentos, percebeu-se que para a tarefa de segmentação, o algoritmo *Label Propagation* se mostrou menos apropriado que o *Fast Greedy* por dois motivos principais [Linares et al. 2012]: (1) ele nem sempre chega a melhor solução (melhor divisão da rede em comunidades) devido à seleção aleatória dos rótulos, no caso de não haver um único rótulo majoritário entre os vizinhos do vértice analisado, e (2) gera uma grande quantidade de comunidades, resultando em uma imagem super-segmentada. Já o algoritmo *Fast Greedy* sempre encontra a melhor divisão da rede em comunidades, considerando a medida de modularidade como critério. Por isso, o algoritmo *Fast Greedy* foi selecionado para compor a abordagem de segmentação proposta.

Ainda que se trabalhe com algoritmos de detecção de comunidades com custo computacional relativamente menor, estes ainda apresentam dificuldade para lidar com imagens de alta resolução. Para solucionar este problema, foi proposto o uso de técnicas de extração de *superpixel* para reduzir a cardinalidade do grafo (o qual representa a imagem) por meio do agrupamento dos pixels da imagem. Dessa forma, tem-se grafos de *superpixels* e não de pixels. Duas abordagens de extração de *superpixels* foram estudadas e analisadas devido ao seu custo computacional reduzido: a *Speeded-Up Turbopixel* e a SLIC. No entanto, a primeira abordagem foi selecionada para a realização dos experimentos, pois consegue gerar *superpixels* mais uniformes e compactos com menor custo computacional, visto que o algoritmo SLIC analisa a região total dos *superpixels* para reposicionar a borda, enquanto o *Speeded-Up Turbopixel* leva em conta somente os pixels ao longo da borda.

# Uma proposta para segmentação de imagens de alta resolução

## 4.1 Considerações iniciais

Este capítulo apresenta a metodologia de segmentação de imagens de alta resolução proposta. Considerando sua aplicação em imagens aéreas de ninhais de aves do Pantanal, a metodologia pode ser dividida em duas fases: pré-segmentação e segmentação final, conforme apresentado na Figura 4.1. A fase de pré-segmentação é baseada em uma nova abordagem de segmentação que combina algoritmos de detecção de comunidades, pertencentes à teoria das redes complexas, com técnicas de extração de *superpixels* (seção 4.2), e será usada para eliminar informações que não são de interesse para a análise (lagos, estradas etc). Esta fase também pode ser usada como segmentação final em imagens que possuem regiões maiores que os *superpixels*. Lembrando que os *superpixels* tem em torno de 100 pixels, pois considera-se o tamanho inicial de  $10 \times 10$ .

Já a fase de segmentação final é baseada na técnica de segmentação por textura *EM/MPM* usando *Markov Random Fields (MRF)* (seção 4.3). A escolha desta técnica se deve à sua capacidade de representar pequenas regiões da imagem por meio da análise de microtexturas [Kindermann and Snell 1980; Gerhardinger 2006]. Quando se trata da análise de imagens de ninhais, nota-se que apenas a técnica de segmentação baseada em detecção de comunidades e *superpixels* não apresenta resultados satisfatórios, pois esta abordagem é mais adequada para identificar objetos que seja maiores que os *superpixels*, e as aves possuem aproximadamente 40 pixels. Além disso, a análise de características de textura se mostra promissora devido à dificuldade de diferenciar aves e dejetos presentes na copa das árvores, pois ambos possuem a cor branca.

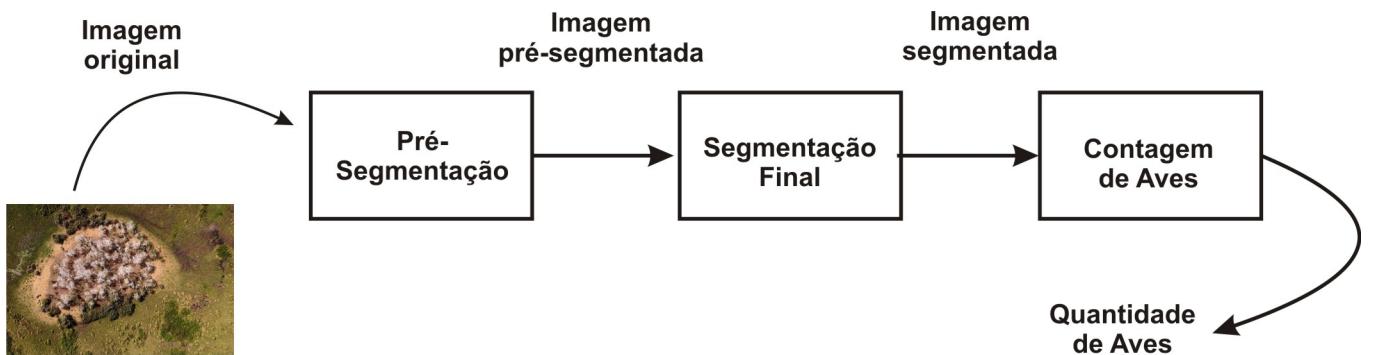


Figura 4.1: Metodologia referente à segmentação de imagens aéreas de ninhais do Pantanal.

Outro ponto importante na área de segmentação de imagens é a possibilidade de avaliar quantitativamente, e não apenas visualmente, a qualidade da segmentação obtida. Logo, este trabalho também propõe uma nova métrica de avaliação baseada em *ground-truth* e, consequentemente, um novo método de seleção da imagem de referência, dentre as diferentes imagens segmentadas manualmente, para comparação com a imagem segmentada automaticamente. A Seção 4.4 apresenta detalhadamente esta nova métrica de avaliação quantitativa.

## 4.2 Segmentação baseada em redes complexas e *superpixels*

A combinação de algoritmos de detecção de comunidades, pertencentes à teoria das redes complexas, com técnicas de extração de *superpixels* representa uma nova abordagem de segmentação, visto que, na prática, combina-se *superpixels* com algoritmos de particionamento em grafos. Cabe ressaltar que a abordagem proposta é apropriada para imagens que possuem objetos/regiões maiores, devido à utilização dos *superpixels*. Também pode ser usada como uma técnica de pré-segmentação, eliminando áreas irrelevantes de imagens de alta resolução, como no caso das imagens de ninhais de aves que serão analisadas. A Figura 4.2 apresenta a metodologia desta nova abordagem de segmentação. Dada uma imagem de tamanho  $N \times N$ , os *superpixels* são extraídos, permitindo a geração da rede complexa (grafo). Em seguida aplicam-se os algoritmos de detecção de comunidades sobre o grafo de *superpixels*, produzindo a imagem segmentada. Detalhes de cada etapa são descritos nas subseções seguintes.

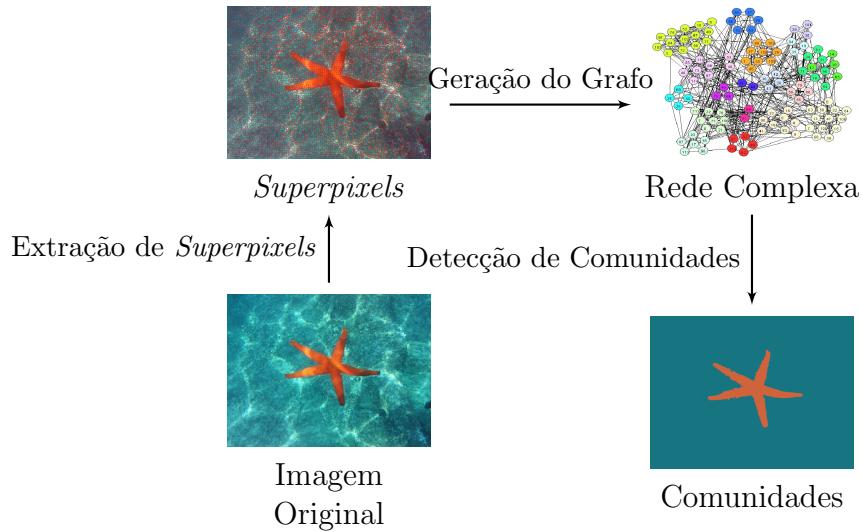


Figura 4.2: Nova abordagem de segmentação de imagens baseada em redes complexas e *superpixels*.

#### 4.2.1 Extração de *Superpixels*

O método de extração de *superpixels* desenvolvido é baseado no algoritmo *Speed-Up Turbopixel* desenvolvido por Cigla e Alatan [Cigla and Alatan 2010]. Este método produz *superpixels* compactos com poucos erros de super-segmentação e também é rápido de computar. Cabe ressaltar que o algoritmo original trabalhava apenas com os níveis de cinza da imagem e foi modificado para incorporar os modelos de cores RGB e CIELAB na função de convergência. Além destes melhoramentos, também foi realizado um amplo estudo para definir os melhores valores para os parâmetros, conforme será apresentado na seção 5.2.3.

Quatro parâmetros distintos influenciam nos resultados do algoritmo: tamanho do *superpixel*  $s$  (inicialmente consiste em um *grid* quadrado cujo tamanho dos lados é  $s$ ); (b) número de iterações  $i$ ; (c) peso da similaridade de intensidade  $\lambda_1$  e (d) peso da restrição de convexidade  $\lambda_2$ . Como mostrado em um trabalho prévio [Linares et al. 2012], pequenas mudanças nos valores destes parâmetros podem afetar a velocidade do algoritmo e a acurácia dos *superpixels* produzidos. Por exemplo, para grandes valores de  $s$  (tamanho do *superpixel*), é necessário um número maior de iterações para a convergência do algoritmo, o que aumenta o tempo de processamento. Para valores muito altos de  $\lambda_2$ , os *superpixels* podem não convergir.

Através dos experimentos, notou-se que a influência do parâmetro  $\lambda_1$  na qualidade da segmentação é reduzida显著mente quando a distância Euclidiana é adotada e o modelo de cores é alterado para CIELAB. Além disso, percebeu-se que os melhores resultados de segmentação foram obtidos com  $\lambda_1 = 1$ , tornando constante o valor deste parâmetro. Com relação

ao parâmetro  $\lambda_2$ , percebeu-se que altos valores geram *superpixels* mais convexos, os quais são difíceis de ajustar adequadamente às bordas dos objetos nas imagens. Experimentos mostraram que valores de  $\lambda_2$  variando de 0.1 à 0.5 produzem resultados acurados.

A quantidade de iterações também foi definida experimentalmente. Notou-se que dez iterações são necessárias para a convergência correta dos *superpixels*, porém, na maioria dos casos, seis iterações são suficientes. Finalmente, o tamanho do *grid* inicial dos *superpixels* foi definido como  $10 \times 10$  de acordo com os experimentos realizados em [Linares et al. 2012], os quais mostraram que valores variando de 10 a 50 levam aos melhores resultados em termos de custo computacional e convergência correta.

#### 4.2.2 Geração do Grafo

Após a extração dos *superpixels*, um grafo (ou rede) é gerado. Geralmente, o que define a existência de uma aresta conectando dois vértices em um grafo é a definição de um peso calculado a partir das características extraídas dos vértices. É importante ressaltar que neste trabalho os vértices não representam um único pixel da imagem, e sim *superpixels* (grupos de pixels). Logo, para gerar o grafo foram consideradas diferentes propriedades dos *superpixels* como, por exemplo, a intensidade média dos níveis de cinza, a média dos canais de cores dos modelos RGB e CIELAB e características de textura extraídas pela técnica *Local Binary Pattern*.

As funções de peso, usadas para estabelecer uma conexão entre os vértices do grafo, empregam dois parâmetros: *threshold* ( $t$ ) e raio ( $R$ ). Conexões são estabelecidas somente se o peso for menor ou igual ao valor de  $t$ . Ressalta-se que o valor definido para  $t$  pode mudar de acordo com a similaridade dos pixels. Experimentos mostraram que um valor de *threshold* muito alto pode gerar comunidades com regiões da imagem misturadas, prejudicando a segmentação e, um valor de  $t$  muito baixo pode super-segmentar a imagem. Destes problemas, o mais grave é a geração de comunidades com regiões misturadas, o que nos leva a evitar valores altos de  $t$ . Já em relação à super-segmentação, percebeu-se que valores baixos de  $t$  geram um grafo com muitos vértices isolados (vértices sem conexões), os quais, após aplicar os algoritmos de detecção de comunidades, são considerados como uma comunidade muito pequena.

Diante destes problemas, este trabalho também propõe a montagem do grafo por meio de um ***threshold adaptativo***, ou seja, inicialmente  $t$  recebe um valor baixo, o qual é incrementado quando o *superpixel* testado não possui nenhuma conexão. Este processo de incremento do valor de  $t$  é realizado iterativamente até que o vértice (*superpixel*) possua pelo menos uma conexão. Além do *threshold* é preciso avaliar o parâmetro raio ( $R$ ), o qual é usado para evitar conexões entre *superpixels* muito distantes. Na prática, as conexões são definidas apenas dentro

de uma região circular de raio  $R$ . A escolha do valor adequado para o raio é difícil, pois este é afetado pelo tamanho da imagem, tamanho dos *superpixels*, a cor e a proximidade das regiões. A Figura 4.3 mostra uma região amarela que comprehende todos os *superpixels* para  $R = 5$ , a partir de um dado *superpixel*.

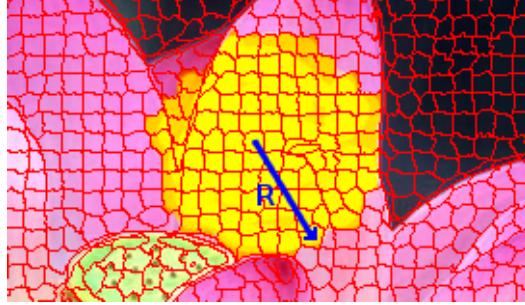


Figura 4.3: Região circular (área amarela) de raio  $R=5$ .

Os parâmetros *threshold* e raio são usados no processo de geração do grafo, o qual é realizado por meio das funções de peso (ou similaridade) responsáveis pelo estabelecimento das conexões entre os vértices, considerando as características extraídas dos *superpixels*. Neste trabalho foram utilizadas e definidas algumas funções de peso. A primeira função testada é baseada nos níveis de cinza da imagem, conforme definido na Equação 4.1.

$$W_{i,j} = 1 - |I_i - I_j| \leq t. \quad (4.1)$$

onde  $W_{i,j}$  é o peso da aresta entre o  $i$ -ésimo e  $j$ -ésimo *superpixel* e está definido no intervalo  $[0, 1]$  e  $I_i$  representa a intensidade média do nível de cinza do *superpixel*  $i$ . As conexões entre os vértices são estabelecidas se o peso é menor ou igual a um *threshold*  $t$ .

Considerando o modelo de cor RGB, foi proposta uma nova função de peso que consiste na distância Euclidiana normalizada, referente à média dos valores dos três canais de cores dos pixels pertencentes aos *superpixels*, conforme a Equação 4.2.

$$W_{i,j} = 1 - \frac{\sqrt{(|R_i - R_j|^2 + |G_i - G_j|^2 + |B_i - B_j|^2)}}{\sqrt{3}} \leq t. \quad (4.2)$$

Outra função de peso proposta emprega o modelo de cor CIELAB e é representada pela Equação 4.3, onde  $L$  corresponde à luminância (a qual varia de 0 (preto) a 100 (branco)) e  $a$  e  $b$  são dois intervalos de cores, que vão do verde ao vermelho e do azul ao amarelo, respectivamente.

$$W_{i,j} = \sqrt{(L_i - L_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \leq t. \quad (4.3)$$

Por fim, a última função de peso proposta é baseada em características de texturas extraídas

dos *superpixels*. Geralmente, o tamanho dos *superpixels* é relativamente pequeno (cerca de 100 pixels), o que dificulta a aplicação dos métodos tradicionais de extração de textura como, por exemplo, os momentos estatísticos ou as matrizes de coocorrência. Logo, optou-se pela implementação da técnica *Local Binary Pattern* (ver seção 2.3), que trabalha com extração de características locais, sendo adequada para extrair textura de pequenas regiões.

Esta técnica gera um histograma  $H$  de padrões binários para cada *superpixel*. Logo, a função de peso é dada pela distância Euclidiana referente a todas as posições do histograma  $H$ , conforme apresentado na Equação 4.4, onde  $P + 2$  corresponde ao número de posições do histograma. Como a quantidade de pixels contida nos *superpixels* não é a mesma para todos, os histogramas são inicialmente normalizados através da divisão de seus valores pelo número total de pixels de cada *superpixel*.

$$W_{i,j} = P + 2 - \left( \sqrt{\sum_{p=1}^{P+2} (H_{i_p} - H_{j_p})^2} \right) \leq t. \quad (4.4)$$

### 4.2.3 Aplicação dos Algoritmos de Detecção de Comunidades

Após a construção do grafo, dois algoritmos de detecção de comunidades foram usados para segmentar as imagens: *Fast Greedy* (*FG*) e *Label Propagation* (*LP*). Estes algoritmos estão implementados na biblioteca *Igraph* [Csardi and Nepusz 2006] e foram incorporados na implementação. Como resultado destes algoritmos tem-se a divisão dos vértices do grafo (*superpixels*) em comunidades, as quais representam as diferentes regiões da imagem. Ressalta-se que, na literatura, geralmente combina-se *superpixels* e algoritmos de particionamento em grafos para realizar a segmentação de imagens [Levinshtein et al. 2009; Cigla and Alatan 2010; Achanta et al. 2010; Achanta et al. 2012; Zhenguo et al. 2012]. Já, a nossa abordagem trabalha com algoritmos de detecção de comunidades aplicados em grafos de *superpixels* e até onde se sabe, esta é uma nova abordagem de segmentação.

Para a tarefa de segmentação, o algoritmo *Fast Greedy* se mostrou mais apropriado que o *Label Propagation*, pois resulta em uma melhor divisão da rede em comunidades, ou seja, aquela com maior valor de modularidade. Já o algoritmo *Label Propagation* nem sempre gera a melhor divisão da rede devido à seleção aleatória dos rótulos, no caso de não haver um único rótulo majoritário entre os vizinhos do vértice analisado. Além disso, experimentos mostraram que ele resulta em uma grande quantidade de comunidades, gerando uma imagem super-segmentada.

## 4.3 Abordagem de segmentação baseada em *Markov Random Fields (MRF)*

A abordagem de segmentação *EM/MPM* baseada em *MRF* complementará a abordagem baseada em redes complexas e *superpixels* na análise das imagens aéreas de ninhais de aves do Pantanal. A técnica *MRF* é capaz de modelar microtexturas apropriadamente, pois usa características puramente locais (vizinhança) para representar as classes (rótulos) dos pixels das imagens. Na prática, as classes são modeladas como campos aleatórios de *Markov* e o problema de segmentação é proposto como um problema de otimização estatística de dado critério (número de pixels classificados corretamente). Detalhes desta abordagem podem ser encontrados na seção 2.2.

Resumidamente, o algoritmo *EM/MPM* funciona da seguinte forma. Inicialmente, define-se o número de classes (rótulos) e gera-se um mapa de classes com os valores das variáveis definidas de forma aleatória. Em seguida, escolhe-se os valores do  $\Theta(0)$ , que consiste na estimativa inicial das variâncias  $\sigma_k^2$  e médias  $\mu_k$  para todas as classes  $k = 1, \dots, L$ . O processo *EM/MPM* começa com um laço de iterações, cujo critério de parada é simplesmente o número  $P$  de iterações. Para cada iteração, inicializa-se o vetor  $v_{k,s} \leftarrow 0$ , que corresponde à probabilidade de cada pixel pertencer a determinada classe. Em seguida, executa-se o algoritmo *MPM* considerando os parâmetros da iteração anterior  $\Theta(p-1)$ , obtendo-se novas estimativas  $v_{k,s}$  e a cadeia *markoviana*  $X(p,t)$ , a qual contém o mapa de classes. Executa-se também o algoritmo *EM* com as novas estimativas  $v_{k,s}$ , obtendo o novo conjunto de parâmetros (médias e variâncias). Por fim, a segmentação final é dada pelo último elemento da cadeia *markoviana*  $X(p,t)$ .

Para não utilizar parâmetros iniciais totalmente aleatórios, visto que influenciam diretamente nos resultados da segmentação, pode-se tentar estimá-los usando alguma abordagem. Para isso, optou-se por uma técnica baseada em limiarização de Otsu, proposta por Gerhardinger [Gerhardinger 2006], cujos detalhes são apresentados na subseção 4.3.1. Além disso, optou-se por incluir a abordagem *Simulated Annealing* [Kirkpatrick et al. 1983; Cerny 1985] no modelo *EM/MPM*, o que melhorou consideravelmente os resultados.

### 4.3.1 Estimação inicial dos parâmetros

No algoritmo *EM/MPM*, é necessário obter a estimativa inicial das médias  $\mu_k$  para cada rótulo  $k$ . Esses valores são importantes pois interferem nos resultados da segmentação. Comer e Delp [Comer and Delp 1995; Comer and Delp 2000] propuseram uma solução que busca manter as médias diferentes entre si e bem distribuídas dentro do domínio dos valores de níveis

de cinza. Porém, nesta abordagem os valores iniciais das médias são calculados independentes da imagem, o que aumenta a chance de serem muito diferentes das médias reais. Já Gerhardinger [Gerhardinger 2006] propôs um algoritmo que usa limiarização (*thresholding*) para pré-segmentar a imagem e, em seguida, obter as estimativas iniciais das médias. O algoritmo se baseia nos seguintes passos:

1. Obter os valores de limiarização  $T_j$ , para cada  $j = 1, \dots, L - 1$ , que dividem o histograma da imagem em  $L$  diferentes regiões.
2. Definir  $T_0 = 0$  e  $T_L = 255$  e calcular as médias  $\mu_k$  para cada  $k = 1, \dots, L$ , usando a Equação 4.5, que consiste na média dos valores do histograma na faixa  $T_{k-1}$  e  $T_k$ :

$$\mu_k = \frac{\sum_{s|T_{k-1} \leq y_s < T_k} y_s}{\sum_{s|T_{k-1} \leq y_s < T_k} 1} \quad (4.5)$$

Para determinar os valores de limiarização, usou-se a técnica de Otsu [Otsu 1979], que se baseia na escolha dos limiares que maximizam a variância inter-classes. Considerando uma limiarização bi-modal, ou seja, com apenas um limiar separando duas classes, calcula-se:

$$T = \arg \min_{\min < t < \max} f(t) \quad (4.6)$$

onde

$$f(t) = w_0(t)w_1(t)[\mu_0(t) - \mu_1(t)]^2 \quad (4.7)$$

$w_0(t) = \sum_{i=\min}^t p_i$ ,  $w_1(t) = 1 - w_0(t)$ ,  $\mu_0(t) = \frac{\sum_{i=\min}^t ip_i}{w_0(t)}$  e  $\mu_1(t) = \frac{\sum_{i=\min}^{\max} ip_i - \sum_{i=\min}^t ip_i}{w_1(t)}$ , onde  $p_i$  é a frequência de pixels de intensidade  $i$  e min e max são os valores mínimo e máximo do nível de cinza, respectivamente. Com base na Equação 4.6, implementou-se a versão recursiva da técnica de Otsu, ou seja, aplica-se a Equação 4.6 recursivamente nas duas regiões no histograma recentemente separadas, até que o número de regiões seja atingido. Este algoritmo é descrito a seguir, onde  $L$  é o número de regiões pré-determinado,  $n$  é o nível de recursão e  $i$  é o índice de cada limiar.

Cada vez que o passo 2 for executado, o número de regiões dobrará. Quando o número de classes procuradas não for potência de 2, é necessário executar o passo 3, onde são ignorados alguns dos limiares recém-criados, utilizando como critério o valor de  $f(t)$ . Dessa forma, são preservados os pontos  $T_i$  correspondentes aos menores valores da  $f(T_i)$ . Reordenando os valores

---

**Algorithm 2:** Otsu recursivo

---

- 1: Fazer  $\min = 0$ ,  $\max = 255$ ,  $n = 1$  e  $i = 1$
  - 2: Encontrar o limiar  $T_i$  usando a Equação 4.6
    1. Finalizar caso  $L \leq 2^n$ .
    2. Executar passo 2 com  $\min = \min$ ,  $\max = T_i$  e  $i = i + 1$
    3. Executar passo 2 com  $\min = T_i$ ,  $\max = \max$  e  $i = i + 1$
    4. Fazer  $n = n + 1$
  - 3: Caso  $L > 2^n$ , ignorar  $2^n - L$  limiares dentre aqueles cujos índices são  $2^{n-1} < i < 2^n$ , para os quais os valores de  $f(T_i)$  são maiores
- 

$T_i$ , para  $i = 1, \dots, L - 1$ , o pré-processamento é concluído por meio das médias  $\mu_k$  de acordo com a Equação 4.5.

### 4.3.2 Simulated Annealing

A técnica *Simulated Annealing (SA)*, proposta de forma independente por Cenry [Cerny 1985] e Kirkpatrick [Kirkpatrick et al. 1983], consiste na generalização de um algoritmo de otimização que visa encontrar a orientação de menor energia (mais estável) de um sistema, usando para isso o método de Metropolis Monte Carlo [Metropolis et al. 1973]. A ideia é substituir a solução atual por uma solução próxima (ou seja, na sua vizinhança no espaço de soluções), escolhida de acordo com uma função objetivo e com uma variável  $T$  (denominada Temperatura). Quanto maior for  $T$ , maior a componente aleatória que será inclusa na próxima solução escolhida. À medida que o algoritmo progride, o valor de  $T$  é decrementado, começando o algoritmo a convergir para uma solução ótima, necessariamente local.

No método *EM/MPM*, este conceito pode ser aplicado na função de probabilidade do campo de rótulos, que é uma distribuição de Gibbs definida com temperatura constante  $T = 1$ . Para melhorar os resultados da segmentação, faz-se um aumento da temperatura no início do processo, que vai diminuindo gradualmente com as iterações do algoritmo. Sabe-se que no início das iterações, o campo de rótulos se encontra bem diferente de uma boa segmentação, visto que, o primeiro campo de rótulos é praticamente aleatório, não existe relação entre o rótulo de um pixel e seu vizinho. Entretanto, à medida que o processo é executado o campo vai se aproximando do resultado esperado. Logo, o uso da técnica *Simulated Annealing* faz sentido e foi considerada no desenvolvimento da abordagem de segmentação baseada em *MRF*.

## 4.4 Nova métrica de avaliação quantitativa da qualidade das segmentações

Sabe-se da importância de avaliar quantitativamente a qualidade das imagens segmentadas. No entanto, esta não é uma tarefa fácil, pois envolve a subjetividade humana, sendo dependente do ponto de vista do usuário que está avaliando. Algumas técnicas de avaliação desenvolvidas são baseada em *ground-truth*, ou seja, um conjunto de imagens previamente segmentadas manualmente por diferentes usuários é usado para comparar com a imagem segmentada automaticamente. Arberlález [Arbeláez et al. 2009] propôs uma métrica baseada em *ground-truth* cujo objetivo é medir a cobertura ou similaridade entre duas imagens segmentadas. Esta métrica é bastante conhecida e utilizada e seu uso é sugerido pelo grupo de Visão Computacional<sup>1</sup> da Universidade de Berkeley da Califórnia, que disponibiliza um conjunto de imagens para pesquisa e educação.

Segundo Arberlález [Arbeláez et al. 2009] para obter o valor da cobertura, primeiro é calculada a sobreposição entre as regiões  $R$  e  $R'$  ( $R \in S$ ), conforme apresentado na Equação 4.8. Em seguida, calcula-se a cobertura das segmentações  $S$  e  $S'$  através da Equação 4.9:

$$O(R, R') = \frac{|R \cap R'|}{|R \cup R'|} \quad (4.8)$$

$$C(S, S') = \frac{1}{N} \sum |R| \max_{R' \in S'} \{O(R, R')\} \quad (4.9)$$

onde  $N$  é o número total de pixels da imagem e a função max corresponde ao maior valor de sobreposição  $O(R, R')$  entre as regiões  $R$  e  $R'$ . Esta métrica gera bons resultados, mas apresenta dois problemas significativos. O primeiro problema desta métrica é que ela não satisfaz a propriedade de simetria, ou seja,  $C(S, S') \neq C(S', S)$ , o que representa um grande inconveniente. O segundo problema refere-se à comparação de duas imagens idênticas, cujo resultado é 0.5, ou seja,  $d(S, S') = 0.5$  onde  $S = S'$ . Normalmente, o resultado esperado é 1 ou 0. Ambos os problemas encorajaram o desenvolvimento de uma nova métrica de avaliação.

A ideia básica da nova métrica é medir a similaridade entre duas imagens segmentadas calculando a maior intersecção das regiões que as compõem. Dadas duas imagens segmentadas  $S$  e  $S'$  compostas de  $R$  e  $R'$  regiões respectivamente, é criada uma matriz  $M$  de tamanho  $R \times R'$ , onde:

$$M_{ij} = |R_i \cap R'_j| \quad (4.10)$$

---

<sup>1</sup><http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

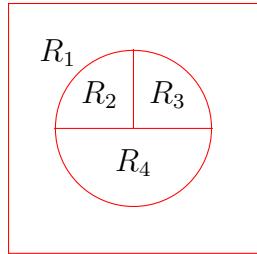
Assim, a intersecção (similaridade)  $I$  entre duas imagens segmentadas  $S$  e  $S'$  é definida como:

$$I(S, S') = \frac{1}{N} \sum_{k=1}^{\min(R, R')} \max_k(M_{lc}) \quad (4.11)$$

onde  $N$  é o número de pixels da imagem e  $\max_k(M_{lc})$  corresponde ao maior elemento da matriz  $M$ , tanto para linha ( $l$ ) quanto para coluna ( $c$ ). Ressalta-se que esta nova métrica satisfaz as seguintes propriedades:

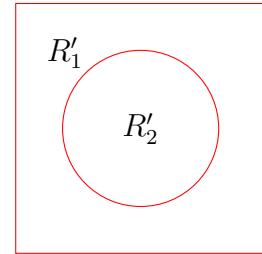
1.  $0 \leq I(S, S') \leq 1$ .
2.  $I(S, S') = 1$ , if  $S = S'$ .
3.  $I(S, S') = I(S', S)$ .
4.  $I(S, S') \geq I(S, S'') \ \& \ I(S, S'') \geq I(S', S'') \Rightarrow I(S, S') \geq I(S', S'')$ .

O exemplo a seguir ilustra mais apropriadamente a métrica proposta. Sejam duas imagens segmentadas  $S$  e  $S'$  (Figura 4.4), a similaridade é obtida aplicando os seguintes passos:



$$|R_1| = 80, |R_2| = 5, |R_3| = 5, |R_4| = 10$$

(a)  $S = 100$



$$|R'_1| = 80, |R'_2| = 20$$

(b)  $S' = 100$

Figura 4.4: Duas imagens segmentadas  $S$  e  $S'$  divididas em  $R = 4$  e  $R' = 2$  regiões, respectivamente.

1. Cria-se a matriz  $M$  com as intersecções entre todas as regiões de cada imagem segmentada usando a Equação 4.10:

$$M = \begin{vmatrix} & R_1 & R_2 & R_3 & R_4 \\ R'_1 & 80 & 0 & 0 & 0 \\ R'_2 & 0 & 5 & 5 & 10 \end{vmatrix} \quad (4.12)$$

2. Dado que o resultado da função  $\min(R, R') = 2$ , é necessário encontrar os dois maiores valores da matriz, tanto para linha ( $l$ ) quanto para coluna ( $c$ ):

$$\max(M_{rc}) = [80, 10] \quad (4.13)$$

3. Finalmente, os maiores valores obtidos devem ser somados e divididos por  $N$ :

$$\begin{aligned} \sum_{k=1}^2 \max_k(M_{rc}) &= 90 \\ N &= 100 \\ I(S, S') &= \frac{90}{100} = 0.9 \end{aligned} \quad (4.14)$$

O resultado da métrica é 0.9, indicando que as duas imagens segmentadas são muito similares, ou seja, tem similaridade igual a 90%. Caso fosse aplicada a métrica proposta por Arbeláez [Arbeláez et al. 2009], o resultado seria 0.43, indicando que a semelhança entre as segmentações testadas é de apenas 43%, o que não condiz com a realidade, visto que o resultado não representa a semelhança real das imagens do exemplo dado.

Cabe ressaltar que este método de avaliação de segmentação proposto é uma contribuição inédita, do ponto de vista de processamento de imagens, e de grande importância para a área devido às dificuldades existentes para avaliar quantitativamente as imagens segmentadas, pois sabe-se que, na maioria das vezes, as imagens são avaliadas apenas subjetivamente, segundo a opinião de um observador.

#### 4.4.1 Novo método de seleção da imagem de referência

O método de avaliação da qualidade da segmentação proposto na seção 4.4 realiza a comparação da imagem segmentada automaticamente com imagens segmentadas manualmente por diferentes usuários. Devido à subjetividade inerente ao processo de segmentação manual, a mesma imagem pode ser segmentada de diferentes maneiras, conforme pode ser visto na Figura 4.5. Para facilitar o processo de comparação, propôs-se um método de seleção de uma **imagem de referência** dentre as diferentes segmentações manuais.

A imagem de referência corresponde àquela que, na média, mais se assemelha às demais segmentações manuais. Para realizar o processo de seleção da imagem de referência, é empregada a métrica descrita na Seção 4.4. Para isso, calculam-se as interseções entre todas as

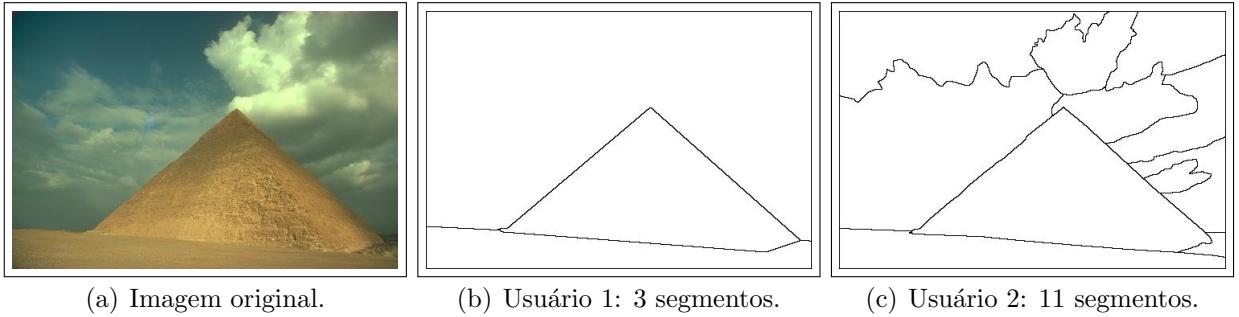


Figura 4.5: Diferentes segmentações manuais para a mesma imagem devido à subjetividade do usuário.

segmentações manuais, as quais são armazenadas em uma matriz  $M$ . Em seguida, são somadas todas as interseções de cada linha e é escolhida a segmentação com a maior soma. O processo pode ser descrito pela Equação 4.15:

$$S_k|k = \arg \max_i \sum_{j=1}^n M_{ij} \quad (4.15)$$

onde  $M$  é a matriz das interseções entre todas as segmentações manuais,  $n$  é quantidade de imagens segmentadas manualmente e  $S_k$  é a segmentação selecionada para ser comparada com a segmentação automática, ou seja, a imagem de referência. Este processo foi aplicado sobre todas as segmentações manuais das 300 imagens do conjunto Berkeley [Martin et al. 2001] e os resultados foram satisfatórios, conforme será apresentado no capítulo seguinte. A Figura 4.6 mostra um exemplo do processo de seleção de uma imagem de referência, considerando todas as segmentações manuais.

## 4.5 Considerações finais

Neste capítulo foi apresentada uma abordagem de segmentação de imagens de alta resolução, que consiste na combinação de algoritmos de detecção de comunidades, pertencentes à teoria das redes complexas, com técnicas de extração de *superpixels*. Para complementar esta abordagem no processo de análise de imagens de ninhais de aves do Pantanal, propõe-se a técnica de segmentação *EM/MPM*, baseada em *Markov Random Fields*, com algumas melhorias (*Simulated Annealing* e estimação inicial do parâmetros via limiarização).

Devido à importância de avaliar quantitativamente (e não apenas visualmente) o resultado da segmentação obtida, propôs-se uma nova métrica de avaliação quantitativa da qualidade das imagens segmentadas. Por ser baseada em *ground-truth*, também foi necessária a criação de

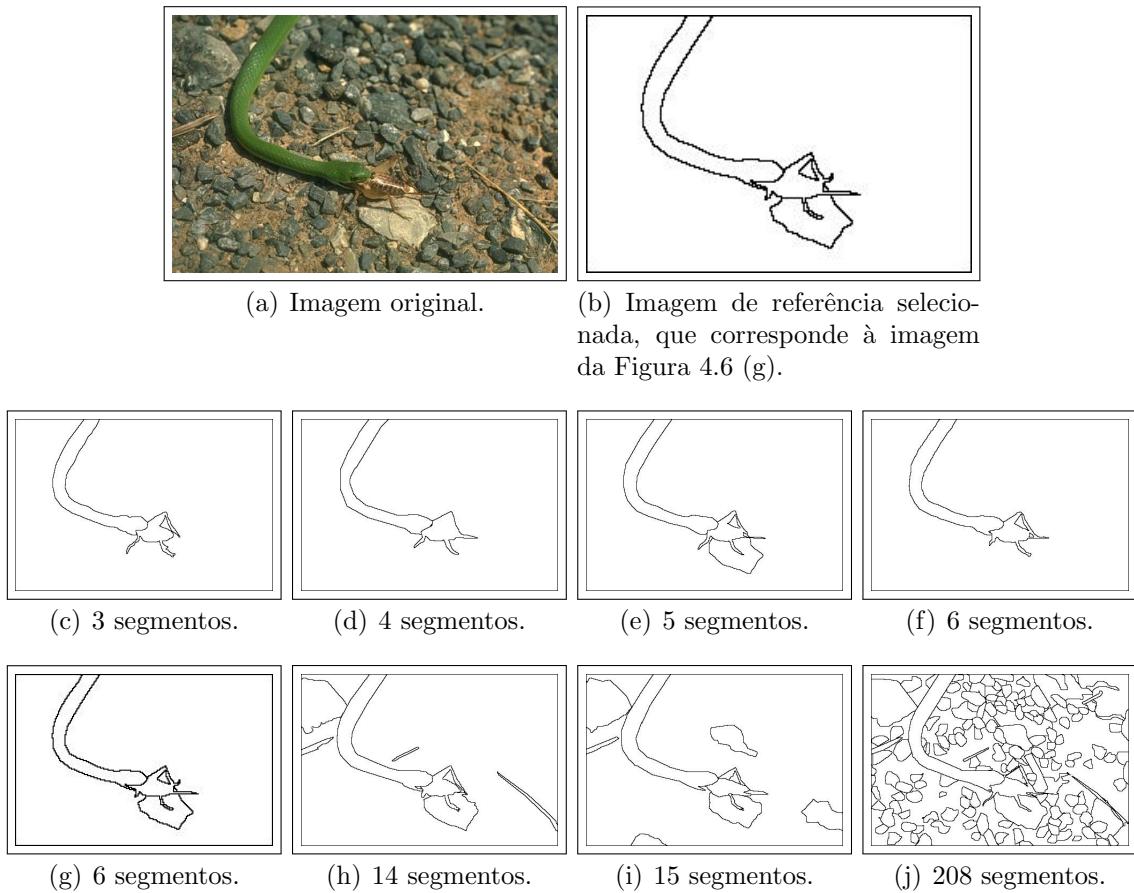


Figura 4.6: Seleção da imagem de referência.

um novo método para selecionar uma imagem de referência, dentre as diferentes segmentações manuais geradas pelos usuários, para ser comparada com a segmentação automática. Os diferentes aspectos da metodologia de segmentação proposta serão ilustrados nos capítulos 5 e 6 por meio de vários experimentos.

# Capítulo 5

## Resultados

### 5.1 Considerações Iniciais

Este capítulo descreve alguns experimentos que ilustram aspectos distintos da abordagem de segmentação proposta, a qual consiste na combinação de algoritmos de detecção de comunidades em grafos com técnicas de extração de *superpixels*. Os experimentos demonstraram a acurácia da segmentação obtida com a nova abordagem e o baixo custo computacional. Além disso, a métrica quantitativa de avaliação da qualidade da segmentação, também proposta neste trabalho, é usada para avaliar as segmentações obtidas. Por fim, faz-se uma comparação dos resultados da abordagem proposta com uma técnica semelhante, definida por Felzenwalb e Huttenlocher [Felzenswalb and Huttenlocher 2004].

### 5.2 Experimentos

Nesta seção são apresentados os resultados da abordagem de segmentação proposta, baseada em *superpixels* e detecção de comunidades em grafos. Na seção 5.2.1 apresenta-se um estudo sobre influência do tamanho inicial dos *superpixels* no tempo de processamento. Em seguida, na seção 5.2.2 mostra-se que é possível usar características de textura no processo de geração do grafo, mesmo com *superpixels* de tamanho pequeno (cerca de 100 pixels). Já na seção 5.2.3 realiza-se um estudo aprofundado sobre o efeito dos parâmetros raio e *threshold* na qualidade da segmentação, considerando o banco de imagens Berkeley [Martin et al. 2001] e a métrica quantitativa de avaliação proposta. Por fim, na seção 5.2.4 compara-se a técnica de segmentação proposta com uma abordagem bem conhecida na literatura [Felzenswalb and Huttenlocher 2004].

### 5.2.1 Influência da variação do tamanho dos *Superpixels*

Este primeiro experimento aborda a influência do tamanho dos *superpixels* no tempo de processamento das três etapas do processo de segmentação: convergência dos *superpixels*, construção do grafo e detecção das comunidades. Para isso foi utilizada uma imagem sintética de tamanho  $700 \times 700$ , contendo três objetos com níveis de cinza distintos e fundo branco. A Figura 5.1 mostra a imagem original (a) e os *superpixels* convergidos considerando os diferentes tamanhos iniciais: 10, 20, 50, 100 e 150 (b,...,f).

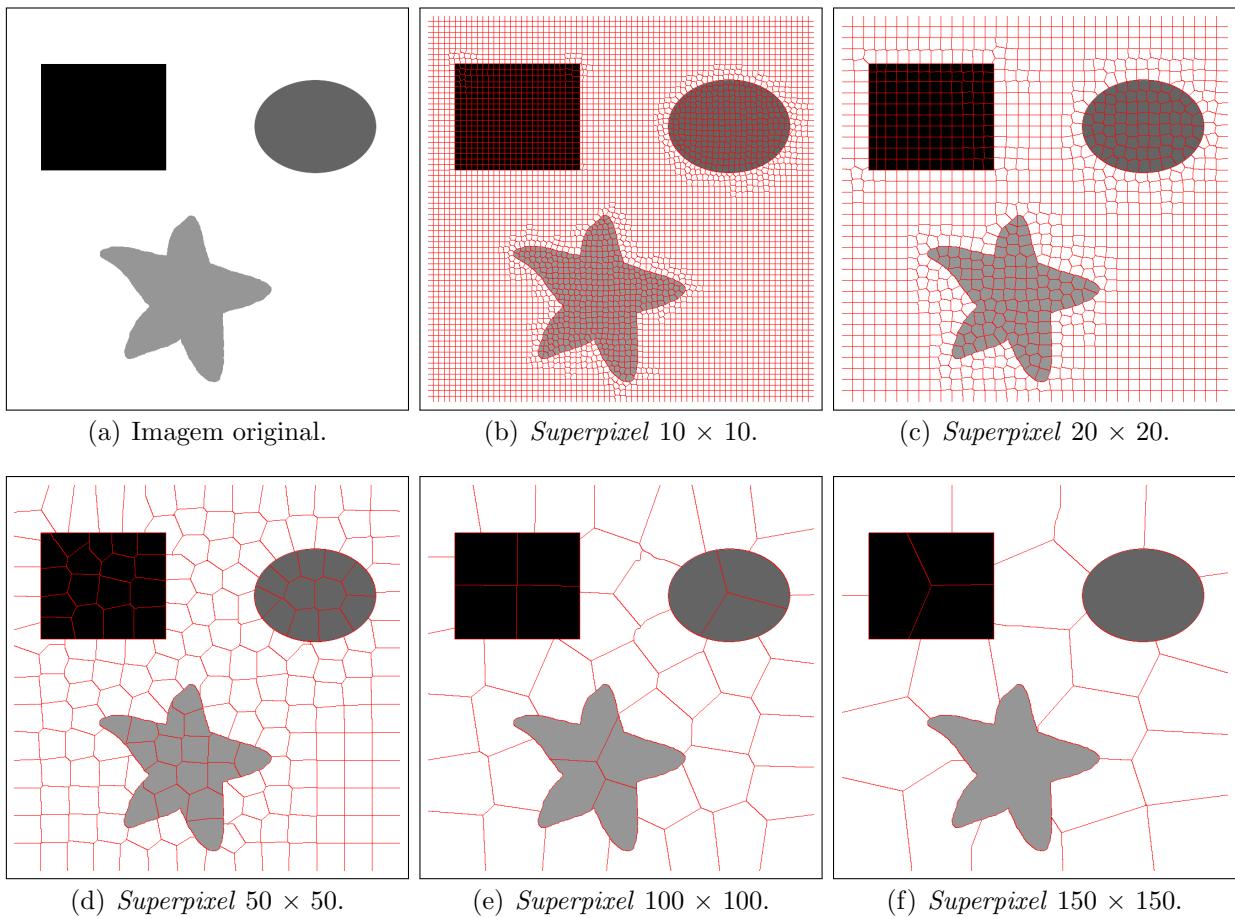


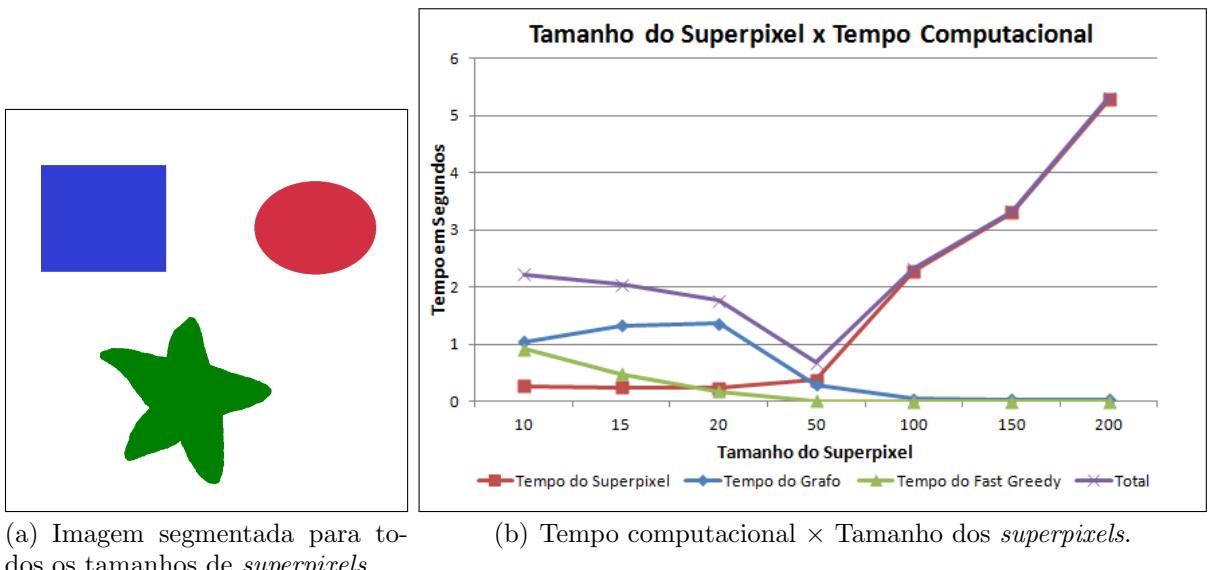
Figura 5.1: Variando o tamanho inicial dos *superpixels*.

A Figura 5.2(a) apresenta o resultado da aplicação do algoritmo de detecção de comunidades, neste caso, o *Fast Greedy*. Percebe-se que independente do tamanho do *superpixel*, obteve-se 100% de acurácia de segmentação, ou seja, quatro classes distintas (cada uma representada por uma cor). Este fato deve-se à conversão correta dos *superpixels* às bordas dos objetos e à detecção das quatro comunidades (três objetos mais o fundo) pelo *Fast Greedy*. Ressalta-se que a função de peso usada para montar o grafo baseou-se nos níveis de cinza da imagem (Equação

4.1).

Para melhor compreender a influência do tamanho dos *superpixels* no custo computacional do processo de segmentação apresentam-se o gráfico da Figura 5.2(b) e a Tabela 5.1. Em geral, com o aumento do tamanho dos *superpixels*, aumenta-se o custo total do processo. Percebe-se que o tempo gasto para a convergência correta dos *superpixels* às bordas da imagem aumenta, porém o tempo do algoritmo de detecção de comunidades diminui consideravelmente, pois o número de vértices do grafo é menor. Já o tempo de construção do grafo, além de ser influenciado pelo tamanho do *superpixel*, também pode ser influenciado pelo raio e *threshold* escolhidos.

Neste experimento também deve-se observar a influência dos diferentes tamanhos dos *superpixels* nos valores dos parâmetros usados na extração dos *superpixels* (iterações,  $\lambda_1$  e  $\lambda_2$ ) e na geração do grafo ( $t$  e  $R$ ), conforme pode ser observado na Tabela 5.1. Tais parâmetros podem variar e afetar diretamente o resultado final da segmentação da imagem. No mais, este experimento mostra que existe um compromisso entre *superpixels* de tamanho  $50 \times 50$  e o tempo de segmentação. Nota-se que o menor tempo de processamento é obtido com este tamanho, superando *superpixels* de tamanhos menores (10, 15 e 20).



(a) Imagem segmentada para todos os tamanhos de *superpixels*.

(b) Tempo computacional × Tamanho dos *superpixels*.

Figura 5.2: Resultado de segmentação (a) e tempo computacional (b) considerando a segmentação de uma imagem sintética.

Tabela 5.1: Parâmetros usados no processo de segmentação da imagem sintética e tempo computacional obtido, para os vários tamanhos dos *superpixels* (Tempo em segundos).

Parâmetros do <i>Superpixel</i>				Parâmetros da Rede		Tempo Computacional			
Tamanho(total)	Iterações	$\lambda_1$	$\lambda_2$	<i>Threshold</i>	Raio	<i>Superpixel</i>	Grafo	<i>Fast Greedy</i>	Total
10 (4900)	7	1	0.005	0.89	6	0.2631	1.0427	0.9139	2.22
15 (2209)	7	1	0.001	0.7	7	0.2407	1.3229	0.4815	2.04
20 (1225)	7	1	0.001	0.7	7	0.2288	1.3606	0.1774	1.77
50 (196)	12	1	0.001	0.7	4	0.3785	0.2962	0.0069	0.69
100 (49)	70	3	0.00009	0.7	2	2.2807	0.0419	0.0001	.32
150 (25)	100	5.5	0.00009	0.7	2	3.2996	0.0283	$6.51 \times 10^{-5}$	3.33
200 (16)	150	9	0.00009	0.72	2	10	0.0368	$6.91 \times 10^{-5}$	10.04

### 5.2.2 Uso de características de textura extraídas via LBP

Geralmente, o tamanho dos *superpixels* obtidos é pequeno (cerca de 100 pixels), o que dificulta a aplicação de métodos tradicionais de extração de textura como, por exemplo, os momentos estatísticos ou as matrizes de coocorrência, visto que eles precisam de uma região maior para conseguir extrair informações de textura representativas. Logo, optou-se pela implementação da técnica *Local Binary Pattern (LBP)* (ver seção 2.3) que trabalha com extração de características locais (vizinhança), sendo adequada para extraer textura de pequenas regiões. A técnica *LBP* foi implementada em linguagem *C++* e, para a realização dos experimentos, considerou-se  $P = 8$ ,  $R = 1$  e a interpolação 8-conectada para obter os níveis de cinza dos vizinhos que não se encontram exatamente no centro do pixel. A Figura 5.3 apresenta a configuração dos vizinhos usada nos experimentos.

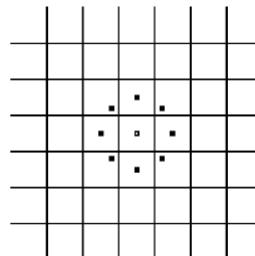


Figura 5.3: Configuração dos vizinhos usados no experimento:  $P = 8$ ,  $R = 1$  e interpolação 8-conectada.

Para verificar a eficácia da técnica de extração de textura *LBP*, experimentos preliminares foram realizados em imagens compostas de mosaicos de texturas. Inicialmente, os *superpixels* foram obtidos considerando-se os seguintes valores de parâmetros: iterações = 10,  $\lambda_1 = 1$  e  $\lambda_2 = 0.09$ . Em seguida, extraiu-se um histograma *LBP* de cada *superpixel*, o qual foi usado para montar o grafo. Durante a montagem do grafo, escolheu-se o raio igual a 5 e o *threshold* igual a 0.1. Por fim, a segmentação foi feita usando-se o algoritmo *Fast Greedy*. Todos os

valores dos parâmetros foram encontrados através da realização de vários experimentos. A Figura 5.4 apresenta os resultados obtidos.

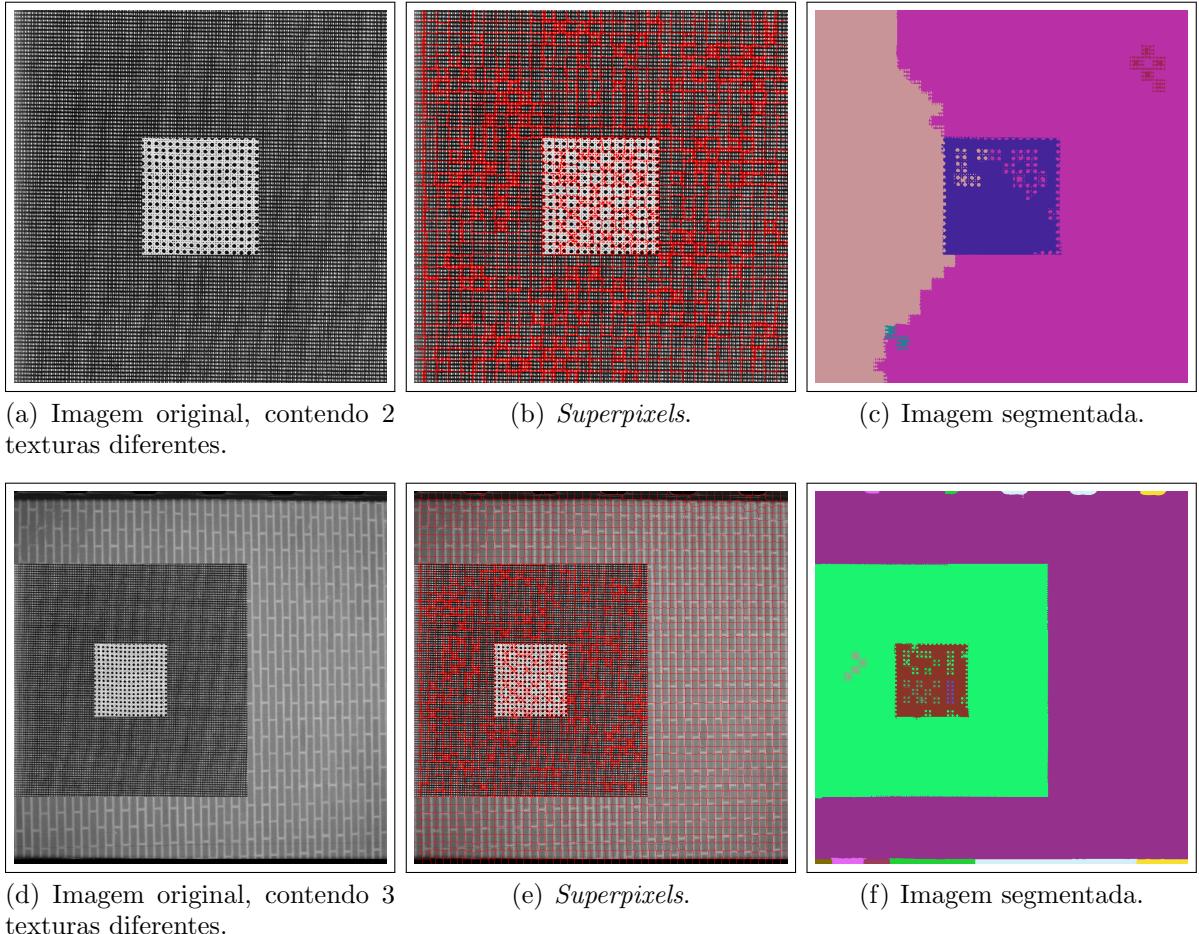


Figura 5.4: Resultados da aplicação do método de segmentação proposto em imagens texturizadas. Função de peso, usada na geração do grafo, baseada na técnica *LBP*.

Este experimento mostrou que independente do tamanho dos *superpixels* e de sua geração ser feita com base em características de cor, foi possível extrair texturas dos *superpixels* durante o processo de geração do grafo, permitindo uma segmentação satisfatória da imagem. No entanto, como a textura presente nestas imagens é pequena e regular, em alguns casos, os *superpixels* se ajustaram muito, obtendo uma maior quantidade de comunidades que o esperado.

Outro experimento foi realizado com uma imagem real de tamanho  $1004 \times 1024$ , apresentada na Figura 5.5(a). Os valores dos parâmetros usados neste experimento foram os mesmos do anterior referente aos mosaicos de texturas. Como resultado, o algoritmo *Fast Greedy* retornou 5 comunidades (Figura 5.5(c)). Aparentemente, pode-se pensar que houve um erro de segmentação, pois o fundo do mar e a estrela pertencem à mesma comunidade, porém isto acon-

teceu devido à semelhança da textura de ambos. Caso fosse considerada a cor, provavelmente estariam em comunidades diferentes.

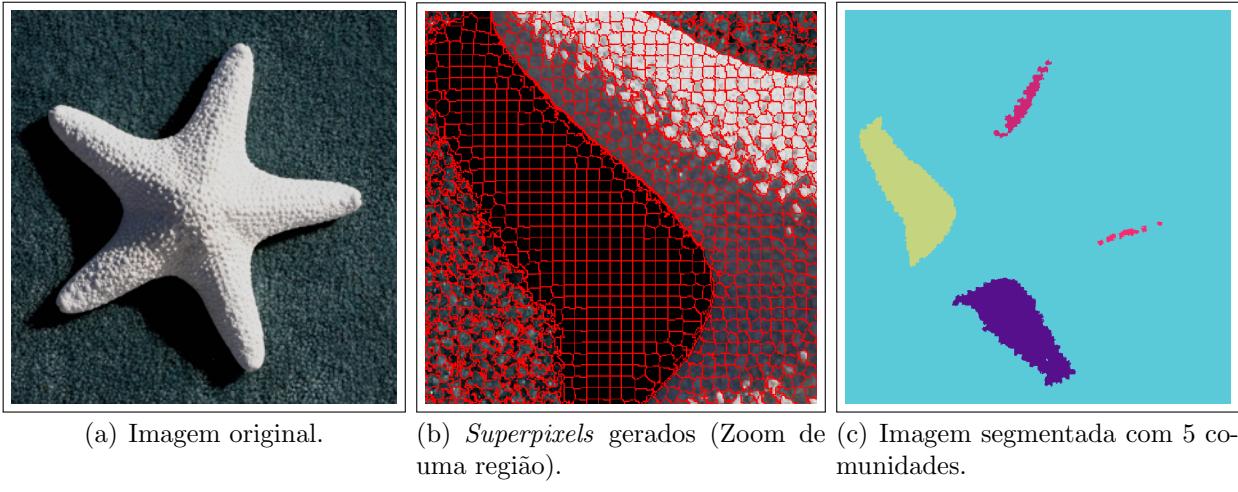


Figura 5.5: Segmentação de uma imagem real, cujo grafo foi montado usando uma função de peso baseada na técnica de extração de textura *LBP*.

A Tabela 5.2 apresenta o tempo computacional referente a todo o processo de segmentação. Embora o tempo para gerar *superpixels* e detectar comunidades seja baixo, o tempo para montar o grafo é relativamente alto devido ao uso da técnica *LBP* para extrair histogramas de textura de cada *superpixel*. Logo, segmentar imagens usando características de textura extraídas via *LBP* é computacionalmente mais custoso que segmentar imagens usando apenas características de cor, no processo de geração do grafo.

Tabela 5.2: Tempo computacional (em segundos) para os experimentos cuja função de peso é baseada em textura.

Imagen	<i>Superpixel</i>	Grafo	Comunidades	Total
Figura 5.4(a)	0.94	31.15	3.25	35.34
Figura 5.4(d)	2.33	179.56	3.94	185.83
Figura 5.5(a)	2.35	162.00	9.40	173.75

### 5.2.3 Avaliação da qualidade da segmentação e definição dos melhores valores para os parâmetros

Este experimento tem por objetivo avaliar quantitativamente a qualidade das segmentações obtidas considerando diferentes valores para os parâmetros raio ( $R$ ) e *threshold* ( $t$ ), usados durante o processo de geração do grafo. Com isso, pôde-se testar a técnica de avaliação proposta na Seção 4.4 e também verificar a influência dos valores dos parâmetros na qualidade da

segmentação obtida. Para realizar este experimento foi usado o conjunto de imagens Berkeley [Martin et al. 2001], o qual possui 300 imagens de tamanho  $481 \times 321$ . Cada uma destas imagens possui um número diferente de segmentações manuais providas por diferentes usuários, permitindo compará-las com as imagens segmentadas automaticamente. Tal comparação é feita usando-se a métrica de avaliação proposta na Seção 4.4.

Cada imagem do conjunto Berkeley foi testada variando-se o raio de 1 a 5 e o *threshold* adaptativo de 0.5 a 40, com incremento de 0.5. Além disso, durante a geração dos *superpixels* considerou-se  $\lambda_1 = 1$ ,  $\lambda_2 = 0.09$ , *superpixels* de tamanho inicial  $10 \times 10$  e modelo de cor CIELAB (Equação 4.3). Como resultado, foram geradas 400 segmentações diferentes para cada imagem e um total de 120.000 segmentações. Para melhor analisar estes resultados, essa enorme quantidade de dados foi armazenada em um banco de dados *Postgresql*<sup>1</sup>. As imagens segmentadas foram comparadas com a imagem de referência obtida levando em conta a técnica proposta na Seção 4.4.1.

A Figura 5.6 apresenta os histogramas da qualidade da segmentação ( $I(S, S')$ ), obtida pela Equação 4.11, considerando o raio de  $R = 1$  até  $R = 5$ . Lembrando que  $0 \leq I(S, S') \leq 1$ , onde  $I(S, S') = 1$  representa a maior acurácia (eixo  $x$ ). Observa-se que o melhor valor de qualidade de segmentação foi obtido para  $R = 4$ , com média e desvio padrão da qualidade igual 0.74 e 0.13, respectivamente. Para  $R = 5$ , a média e o desvio padrão são 0.73 e 0.13, respectivamente. Como os valores obtidos são bastante similares, conclui-se que  $R = 4$  ou  $R = 5$  podem ser usados para obter segmentações acuradas. Também verificou-se que não há perda de qualidade de segmentação para  $R = 5$ , porém o custo computacional aumenta.

A Figura 5.7 mostra a relação entre o *threshold* ( $t$ ) e a qualidade da segmentação para raio variando de  $R = 1$  até  $R = 5$ . Utilizou-se o esquema de *threshold* adaptativo, onde  $t$  varia de 0.5 à 40, com incremento de 0.5 a cada iteração. Ressalta-se que para montar os histogramas utilizou-se apenas as segmentações com os maiores valores de qualidade de segmentação. O coeficiente de correlação de Pearson entre a qualidade da segmentação ( $I(S, S')$ ) e o *threshold* é  $\rho \approx -0.3$  para todos os casos. Logo, diferentemente do raio, o valor do *threshold* não tem uma influência significativa no resultado da segmentação. Os resultados mostraram que valores mais baixos de  $t$ ,  $t \leq 15$ , geram segmentações mais acuradas. Para efeito de comparação, a Figura 5.8 mostra o tempo de processamento da segmentação, considerando o valor do raio, para o *threshold* estático (igual a 0.75) e adaptativo. Percebeu-se que o uso da abordagem do *threshold* adaptativo não implica no aumento significativo do tempo de processamento, mesmo tendo que avaliar alguns *superpixels* mais que outros. Entretanto, verificou-se que o uso do *threshold* adaptativo provê resultados mais acurados.

---

<sup>1</sup><http://www.postgresql.org/>

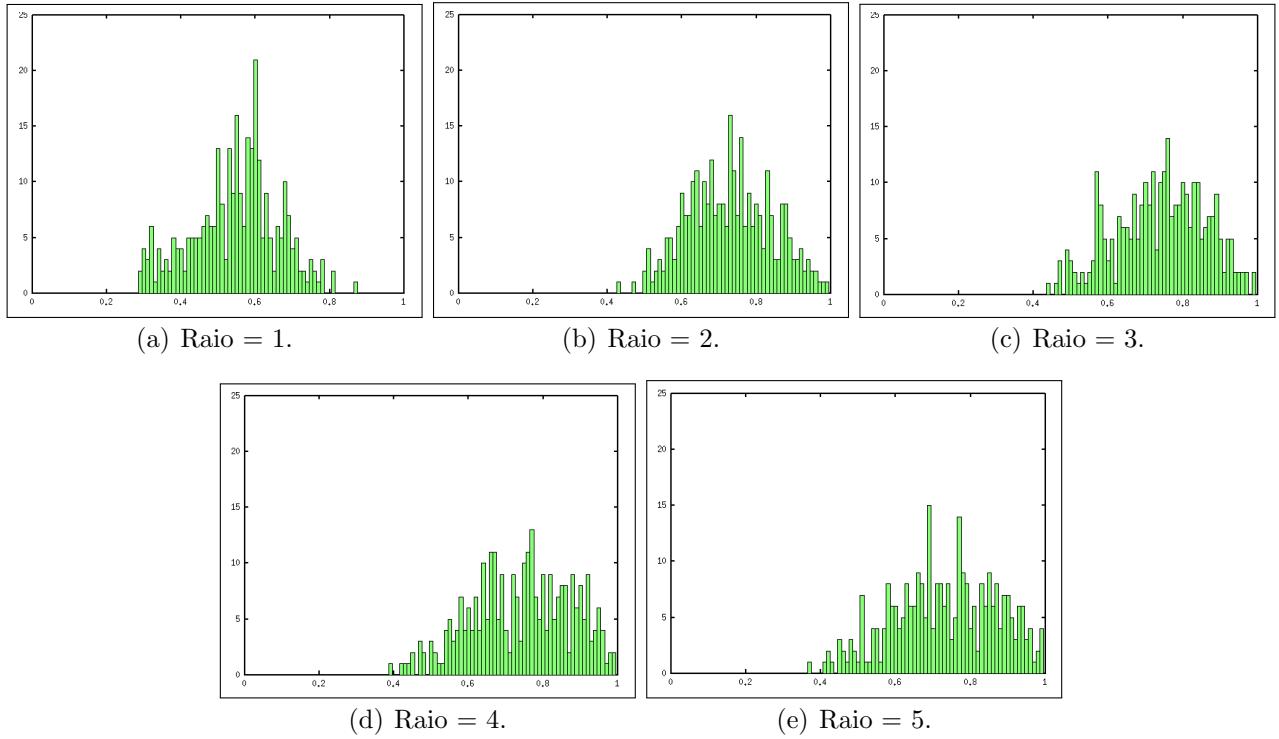


Figura 5.6: Histogramas da qualidade da segmentação ( $I(S, S')$ ) para (a)  $R = 1$ , (b)  $R = 2$ , (c)  $R = 3$ , (d)  $R = 4$  e (e)  $R = 5$ . O eixo  $x$  representa o valor de qualidade e o eixo  $y$  corresponde à quantidade de imagens.

A Figura 5.9 mostra alguns resultados da técnica de segmentação proposta, a qual combina *superpixels* e algoritmos de detecção de comunidades. As segmentações são muito acuradas quando comparadas com as segmentações manuais do banco de imagens Berkeley. A qualidade da segmentação é calculada considerando a imagem de referência obtida conforme descrito na seção 4.4.1. A abordagem proposta demonstra ser muito acurada, visto que  $I(S, S') > 0.95$  para todos os casos.

#### 5.2.4 Comparação com método de segmentação relacionado

O método de segmentação proposto, baseado em *superpixels* e redes complexas, foi comparado com um método de segmentação baseado em grafos bem conhecido na literatura, desenvolvido por Felzenwalb e Huttenlocher [Felzenswalb and Huttenlocher 2004]. Tal método segmenta a imagem de acordo com um critério para avaliar se existe ou não uma borda entre dois componentes em uma região. Este critério calcula a dissimilaridade não negativa entre dois componentes vizinhos, a qual é representada no grafo como o peso da aresta que conecta dois vértices. Assim, o grafo é criado conectando-se apenas os pixels vizinhos na imagem.

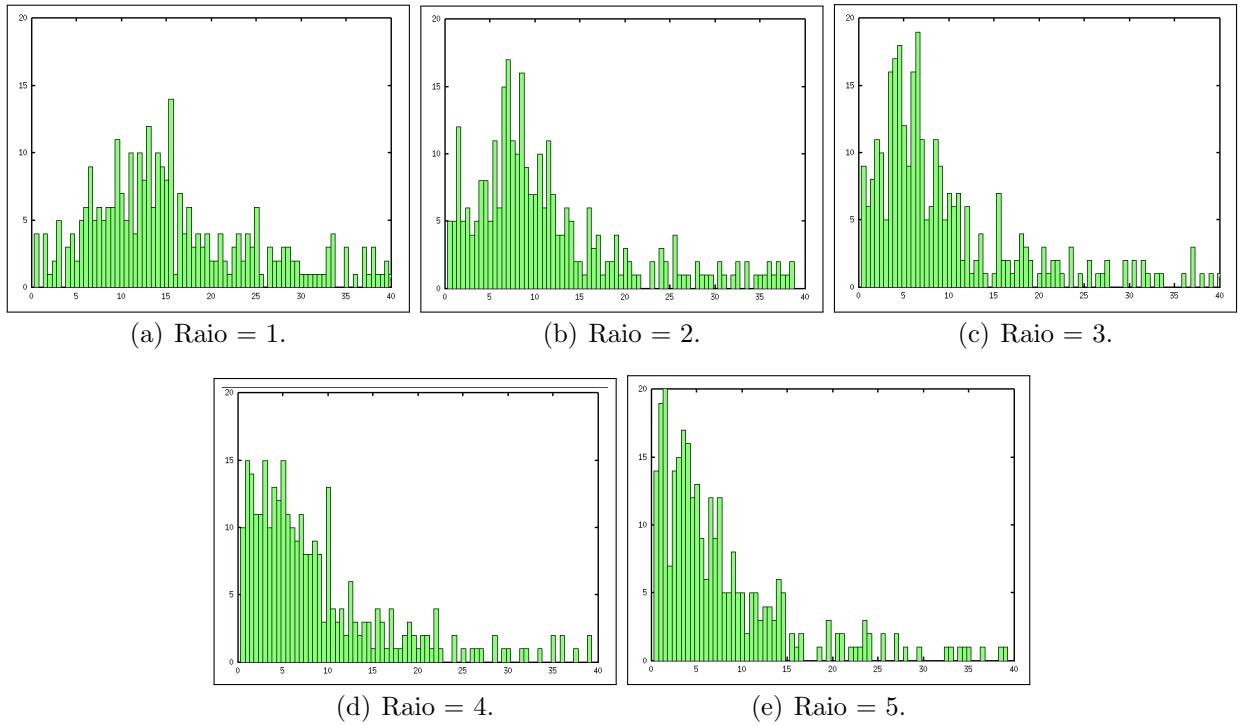


Figura 5.7: Histogramas referente à relação entre a qualidade da segmentação  $I(S, S')$  e o *threshold*, considerando (a)  $R = 1$ , (b)  $R = 2$ , (c)  $R = 3$ , (d)  $R = 4$  e (e)  $R = 5$ . O eixo  $x$  representa os valores do *threshold* adaptativo e o eixo  $y$  representa a quantidade de imagens. Ressalta-se que considerou-se apenas as imagens com os maiores valores de qualidade.

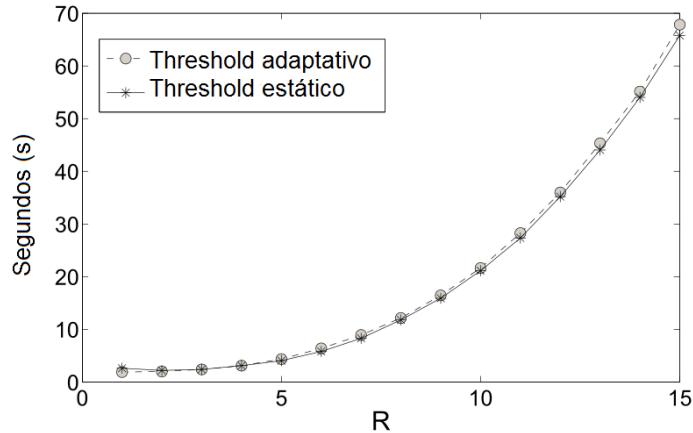


Figura 5.8: Tempo de processamento (em segundos) de acordo com o raio ( $R$ ) para os *thresholds* estático ( $t = 0.75$ ) e adaptativo.

Lembrando que a quantidade de vértices no grafo é igual à quantidade de pixels na imagem.

Em cada passo do algoritmo, compara-se o peso da aresta do vértice atual com seus vizinhos. Se o peso é menor a um limiar  $k$ , estes vértices são unidos em um segmento. Após unir

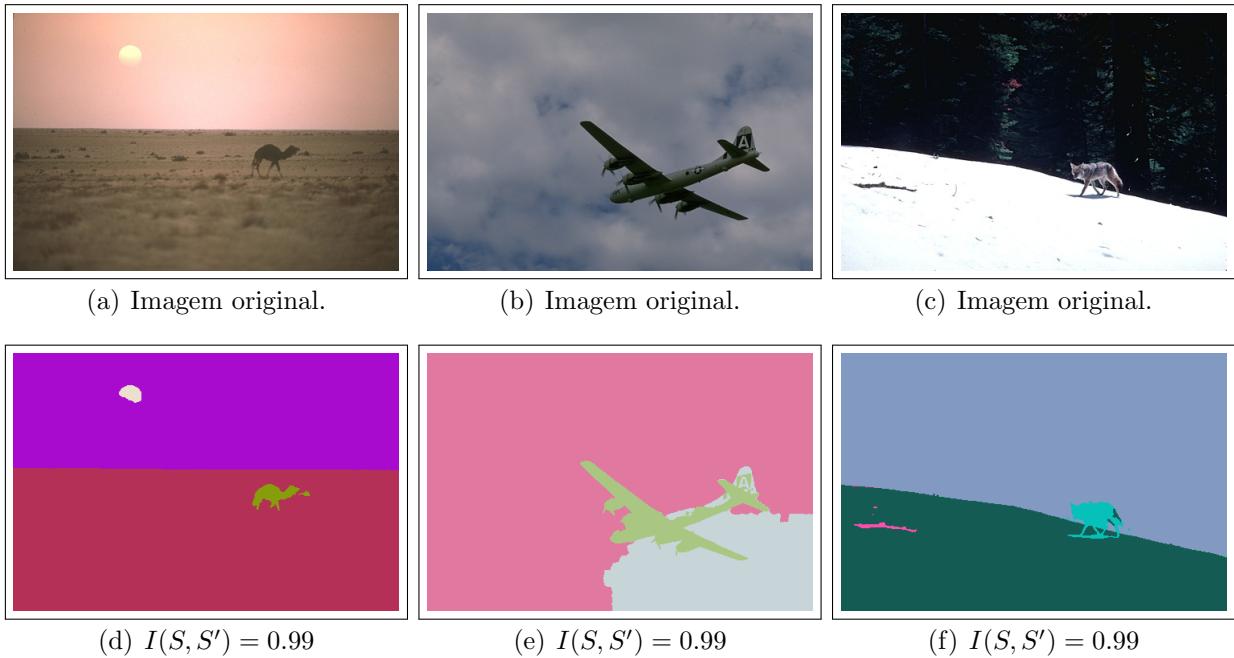


Figura 5.9: Exemplos de segmentações obtidas pela abordagem proposta, baseada em *superpixels* e detecção de comunidades em grafos. (a)-(c) são imagens originais e (d)-(f) são as respectivas segmentações com os valores de qualidade  $I(S, S')$ .

dois vértices, é aplicado um processo que avalia a similaridade (com base na intensidade de cor) do segmento no qual foram inseridos. Este algoritmo também realiza um pré-processamento e um pós-processamento para melhorar o resultado. O pré-processamento consiste em aplicar um filtro Gaussiano sobre a imagem para eliminar o ruído e o pós-processamento é aplicado para evitar a presença de segmentos muito pequenos, permitindo que o usuário defina o menor tamanho do segmento (parâmetro  $min$ ). Neste pós-processamento, quando é detectada a presença de um segmento muito pequeno, este é unido a um segmento vizinho por meio da função de dissimilaridade. No total, o algoritmo precisa de três parâmetros:  $\sigma$  para o filtro Gaussiano, limiar  $k$  para controlar a similaridade dos pixels e  $min$  para limitar o menor tamanho dos segmentos.

O método de Felzenwalb foi testado no conjunto de imagens Berkeley [Martin et al. 2001] considerando dois conjuntos de parâmetros definidos pelos autores como a melhor combinação. Ambos os conjuntos produziram resultados muito similares, obtendo valor médio de qualidade de segmentação igual a 0.52 e 0.48, usando a técnica de avaliação proposta na seção 4.4. Percebe-se que as médias são baixas (e próximas do aleatório) quando comparadas às médias obtidas pelo método de segmentação proposto neste trabalho. A Figura 5.10 apresenta alguns exemplos dos resultados obtidos pelo método de segmentação proposto e pelo método de

Felzenwalb, juntamente com os valores de qualidade.

### 5.3 Considerações Finais

Neste capítulo foram apresentados experimentos relacionados com a abordagem de redes complexas combinada com a extração de *superpixels* para a segmentação de imagens de alta resolução. Resultados mostraram que o método proposto é capaz de realizar uma segmentação acurada em baixo tempo computacional. Ressalta-se que a redução da cardinalidade do grafo pelo método de extração de *superpixels* é fundamental para a redução do custo computacional da montagem do grafo e da aplicação do algoritmo de detecção de comunidades. Sem a abordagem de *superpixels*, utilizar determinados algoritmos de detecção de comunidades para segmentar grandes imagens se torna inviável.

Uma questão importante é a influência dos valores dos parâmetros usados na extração dos *superpixels* (tamanho inicial, iterações,  $\lambda_1$  e  $\lambda_2$ ) e na geração do grafo (*threshold t* e raio *R*) nos resultados da segmentação. Logo, a busca pelos melhores valores de parâmetros, considerando o banco de imagens Berkeley [Martin et al. 2001] e a técnica de avaliação proposta, foi de extrema importância para viabilizar o uso da abordagem de segmentação proposta em conjuntos de imagens diferentes.

Além disso, pôde-se perceber a importância do uso do *threshold* adaptativo, o qual evita a super-segmentação da imagem sem elevar muito o custo computacional. Por fim, notou-se que funções de peso baseadas em características de textura são viáveis e podem levar à bons resultados em imagens texturizadas. No entanto, o uso da técnica *Local Binary Pattern* eleva o custo computacional devido à geração de um histograma de textura para cada *superpixel*.

No próximo capítulo serão apresentados os resultados desta abordagem de segmentação em imagens de ninhais de aves do Pantanal. Além disso, a abordagem de segmentação por textura baseada em *MRF* também será considerada e diferentes experimentos serão realizados para mostrar sua acurácia na identificação de aves.

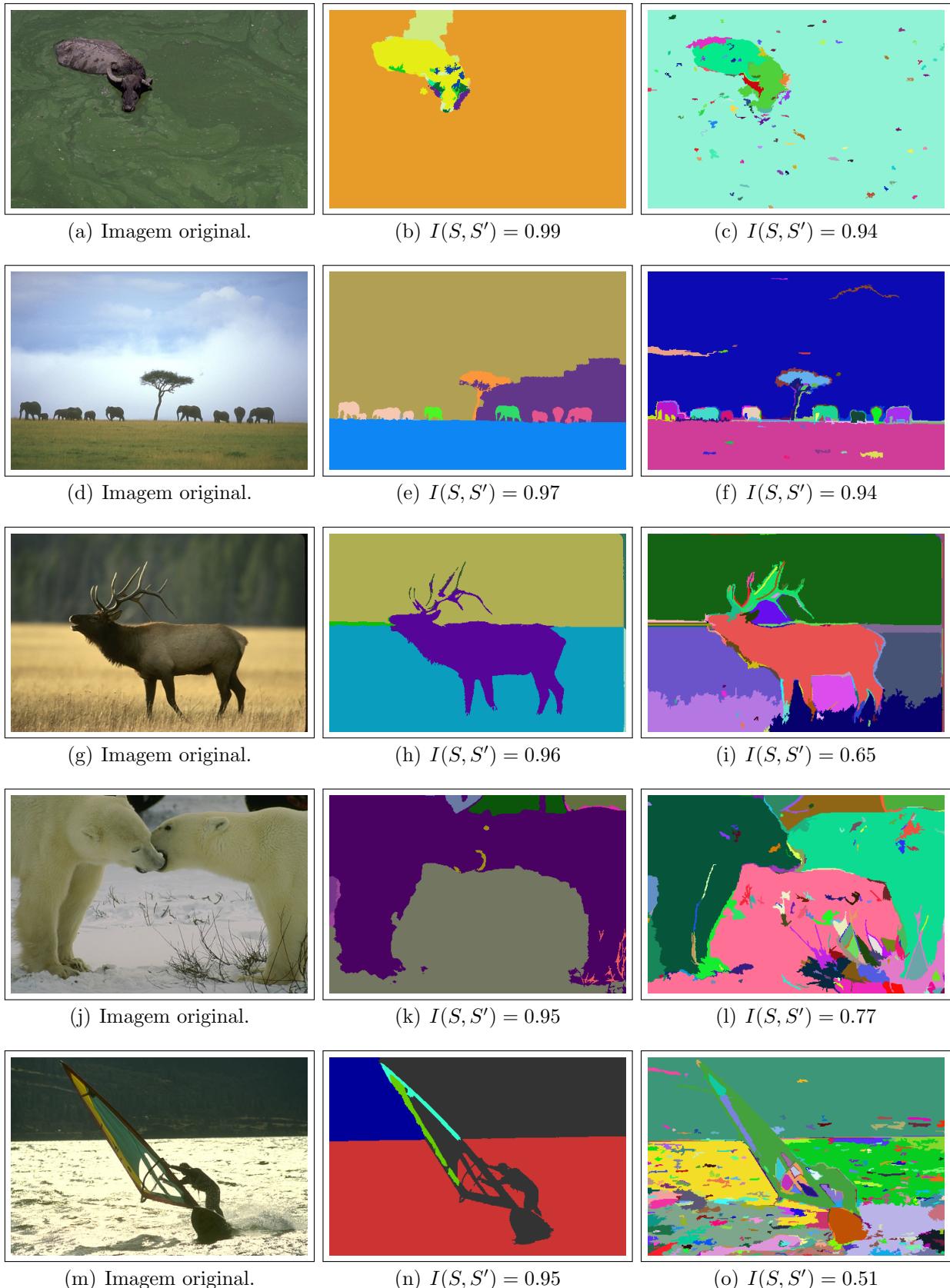


Figura 5.10: A primeira coluna representa a imagem original, enquanto a segunda e a terceira representam as segmentações obtidas pela abordagem proposta e pelo método de Felzenwalb e Hunttenlocher [Felzenswalb and Huttenlocher 2004], respectivamente. O valor de  $I(S, S')$  indica a qualidade da segmentação quantificada pelo método proposto na Seção 4.4.

# Análise das Imagens de Ninhais de Aves do Pantanal

## 6.1 Considerações Iniciais

Um dos objetivos deste trabalho de doutorado é analisar imagens de alta resolução de ninhais de aves do Pantanal mato-grossense, na tentativa de identificar e contar as aves automaticamente. Esta não é uma tarefa fácil devido ao tamanho e características das imagens adquiridas. Primeiramente, as imagens apresentam uma resolução de  $5184 \times 3456$ , o que limita a aplicação de técnicas que apresentam alto custo computacional. Em segundo lugar, algumas imagens possuem áreas irrelevantes para a análise como, por exemplo, lagos e estradas, que podem ser eliminadas. Além disso, a presença de dejetos no topo das árvores dificulta a identificação das aves, pois este material orgânico, ao se ressecar, apresenta coloração esbranquiçada, muito semelhante à das aves, o que dificulta a identificação. Em vista destes fatos, optou-se por uma abordagem de segmentação em dois estágios: pré-segmentação, por meio de redes complexas e *superpixels*, e segmentação final via *Markov Random Fields (MRF)*.

A primeira etapa segmenta a imagem globalmente, separando a região de interesse (ninhais) das demais (lagos, estradas etc). Ressalta-se que a necessidade desta etapa dependerá da imagem analisada, visto que algumas imagens obtidas focam diretamente os ninhais. A segunda etapa, via *MRF*, analisa a microtextura da região dos ninhais e segmenta as aves, diferenciando-as dos dejetos. Pode-se dizer que esta abordagem é uma segmentação em duas escalas distintas, uma vez que a pré-segmentação, por ser baseada em *superpixels*, se atém a objetos maiores, enquanto a segmentação por *MRF* é capaz de identificar pássaros que possuem uma área bastante pequena.

As próximas seções detalharão todo o processo de análise das imagens de ninhais, destacando

os resultados obtidos por cada técnica e as dificuldades encontradas. Inicialmente, a seção 6.2 explica o processo de aquisição das imagens de ninhais no Pantanal. Em seguida, a seção 6.3 aborda a pré-segmentação baseada em redes complexas e *superpixels*. Por fim, a seção 6.4 apresenta os resultados da segmentação final usando *MRF*.

## 6.2 Aquisição de imagens de ninhais no Pantanal

Foi realizada, em outubro de 2013, uma viagem ao Pantanal Mato-Grossense, para obtenção de amostras de imagens aéreas de ninhais da ave *Mycteria Americana*. Como os responsáveis pela aeronave não-tripulada AGX Tiriba (pertencente aos integrantes do projeto Monitoramento Biológico pelo Tamanho Populacional de Colônias de Aves Aquáticas de Topo de Cadeia Alimentar no Pantanal (Proc. CNPQ n°563299/2010 – 0)), que seria usada para a aquisição das imagens, não obtiveram a autorização da ANATEL a tempo de realizar a missão (as aves são migratórias e permanecem no Pantanal apenas em uma determinada época do ano), foi necessário usar um avião tripulado equipado com uma câmera de alta resolução.

A aeronave sobrevoou os ninhais a uma altitude de cerca de 900 metros, permitindo uma resolução de aproximadamente  $10\text{cm/pixel}$ . No total, dois ninhais foram imageados, um na Fazenda Sangradouro e outro na Fazenda Piuval, e um conjunto de 450 imagens foi obtido. No entanto, praticamente metade destas imagens são repetidas ou apresentam sobreposição, reduzindo o número de imagens disponíveis para análise. A Figura 6.1 apresenta algumas amostras destas imagens. Para melhor visualizar as aves, a Figura 6.2 apresenta o zoom de uma região da imagem apresentada na Figura 6.1(a). É possível perceber a presença dos dejetos no topo das árvores, o que dificulta a segmentação das aves, tornando necessário o uso de técnicas que capturem microtexturas.

## 6.3 Pré-segmentação: aplicando a abordagem de segmentação baseada em redes complexas e *superpixels*

Esta etapa consiste na aplicação da abordagem de segmentação baseada em redes complexas e *superpixels* nas imagens de ninhais. Como esta abordagem gera grafos de *superpixels* e não de pixels, a sua aplicação direta na identificação das aves não apresenta resultados satisfatórios, pois as aves ocupam uma parte muito pequena da imagem. Com isso, essa abordagem foi usada para delimitar os ninhais, eliminando informações que não são de interesse para a análise.

Inicialmente, foi realizado um estudo semelhante ao apresentado na seção 5.2.3 para definir os melhores valores para os parâmetros usados no processo de extração dos *superpixels* (tamanho

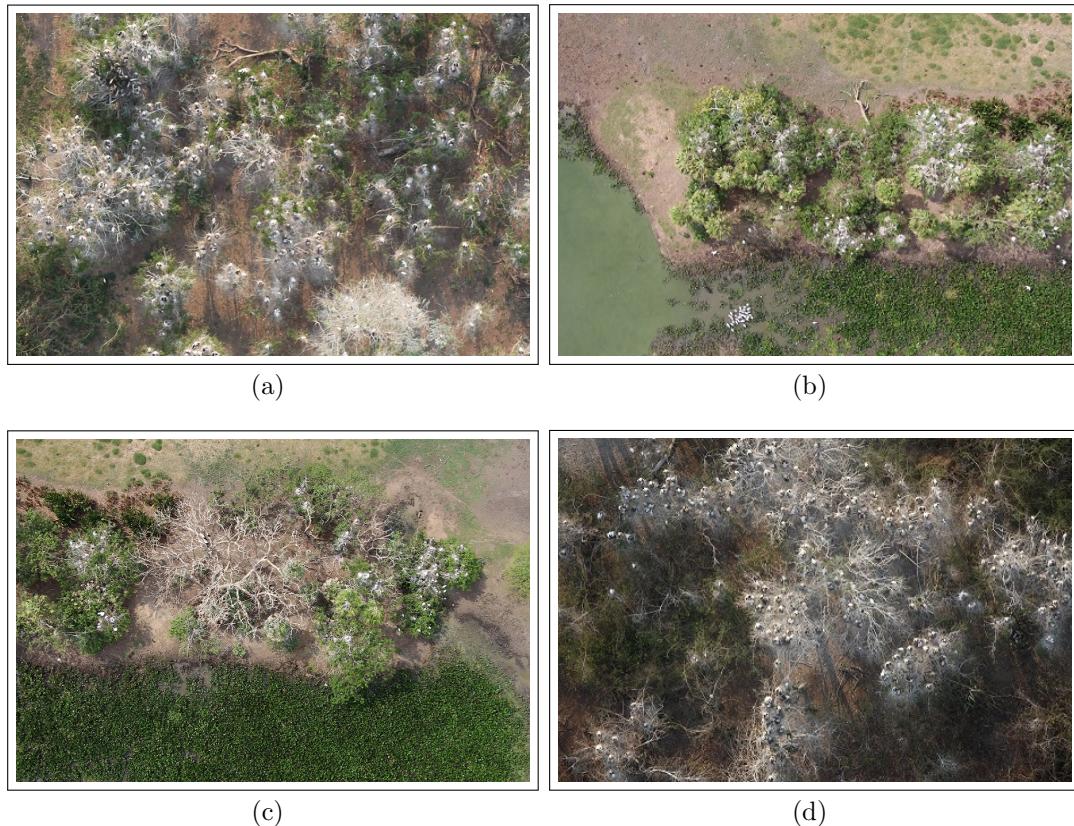


Figura 6.1: Amostras de imagens de ninhais do Pantanal obtidas na expedição realizada em outubro de 2013.



Figura 6.2: Zoom digital de uma região da imagem apresentada na Figura 6.1(a).

inicial, iterações,  $\lambda_1$  e  $\lambda_2$ ) e geração do grafo ( $threshold(t)$  e raio( $R$ )), considerando o conjunto de imagens de ninhais. Após vários experimentos, os melhores resultados foram obtidos com tamanho inicial dos *superpixels* igual a 10, 6 iterações,  $\lambda_1 = 1$ ,  $\lambda_2 = 0.09$ ,  $t = 3$  e  $R = 5$ . Além disso, a função de peso baseada no modelo de cores CIELAB foi utilizada. As Figuras 6.3 e 6.4 apresentam algumas amostras dos resultados.

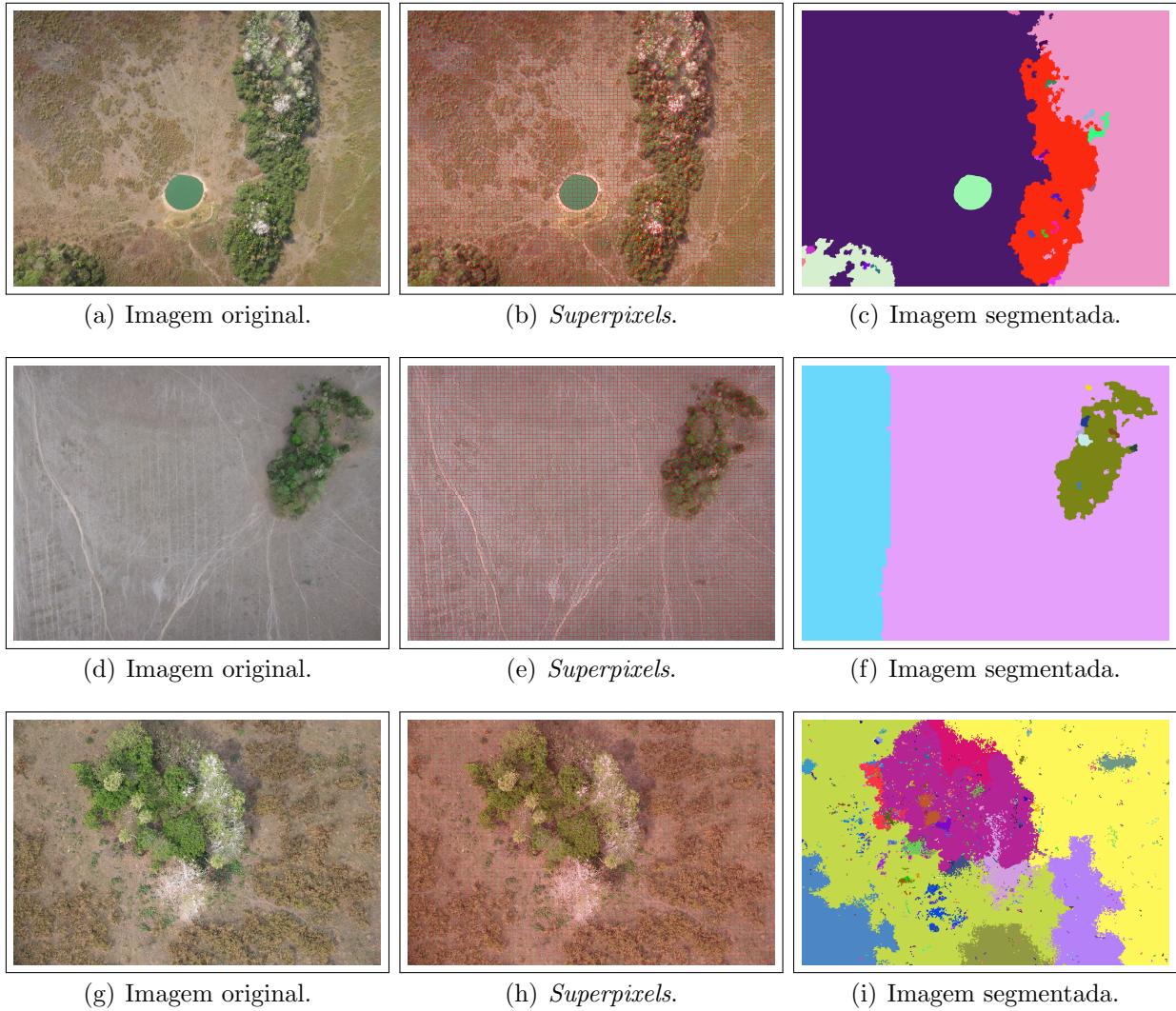


Figura 6.3: Resultados da pré-segmentação de algumas imagens de ninhais do Pantanal utilizando o método proposto baseado em redes complexas e *superpixels*.

Os resultados mostraram que a técnica baseada em redes complexas e *superpixels* consegue segmentar, na maioria das vezes, os ninhais em comunidades diferentes do restante dos objetos (lagos, estradas, fundo etc) da imagem. Com isso, é possível eliminar as regiões que não são necessárias, considerando a cor da comunidade que elas pertencem, por exemplo, na Figura 6.3(f), pode-se eliminar as regiões representadas pelas comunidades de cor azul e lilás e analisar

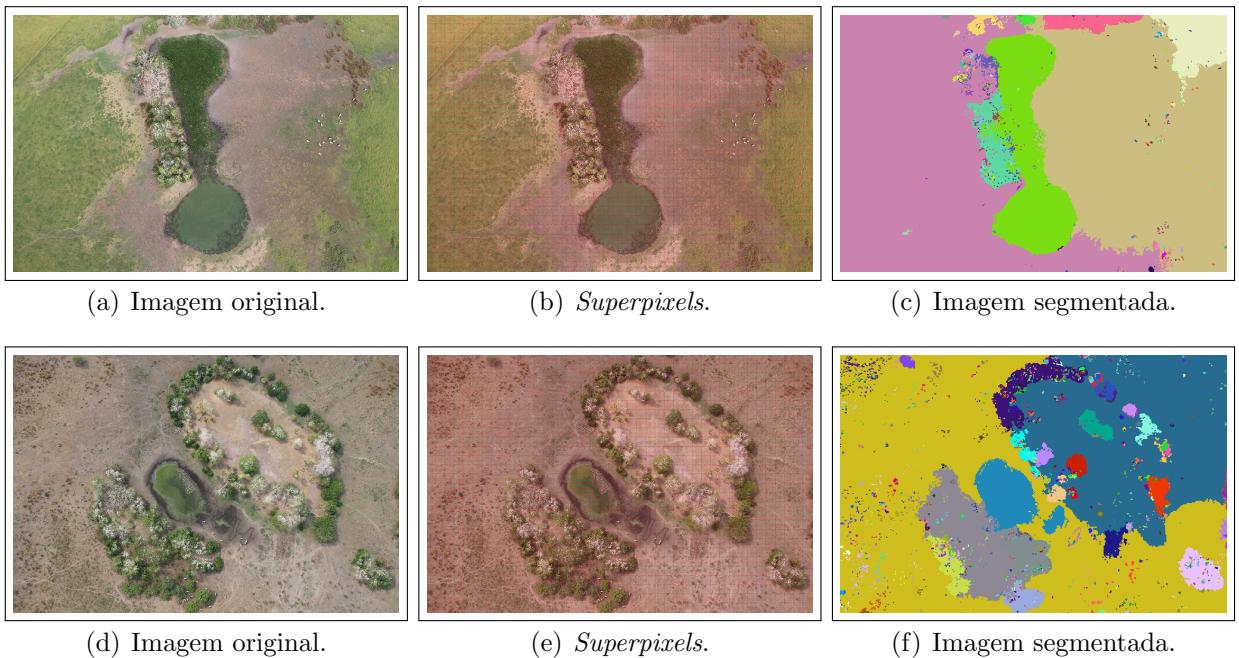


Figura 6.4: Resultados da pré-segmentação de algumas imagens de ninhais do Pantanal utilizando o método proposto baseado em redes complexas e *superpixels*.

apenas o restante via *MRF*. Por meio deste processo, é possível reduzir o custo computacional de se trabalhar com toda a imagem.

## 6.4 Segmentação final: aplicando a técnica de segmentação baseada em *Markov Random Fields*

Na prática, para realizar a segmentação usando *Markov Random Fields*, combinam-se dois algoritmos: *Expectation Maximization (EM)* (para estimção dos parâmetros da distribuição da imagem) e *Maximization of the Posterior Marginal (MPM)* (para realizar a otimização estatística do campo de rótulos), conforme pode ser consultado na seção 2.2. Além disso, com o intuito de melhorar os resultados, propôs-se a inclusão do algoritmo *Simulated Annealing* no *EM/MPM* e a determinação inicial dos parâmetros ( $\Theta(0)$ ) considerando limiarização via Otsu (maiores detalhes no Capítulo 4).

Diferentes experimentos foram realizados para ilustrar os resultados da técnica de segmentação utilizada. O primeiro experimento (seção 6.4.1) aborda a escolha dos valores dos parâmetros usados pela segmentação *EM/MPM* baseada em *MRF* considerando a análise das imagens de ninhais e também apresenta alguns resultados. Já o segundo experimento (seção 6.4.2) apresenta a análise de uma imagem de um ninhal completo, considerando diferentes esca-

las. Por fim, o terceiro experimento (seção 6.4.3) ilustra a comparação da contagem automática com a contagem *in loco* e a contagem manual nas próprias imagens para mostrar a acurácia da técnica.

### 6.4.1 Experimento 1: Definição dos valores dos parâmetros da segmentação baseada em *MRF* para análise de imagens de ninhais

A técnica de segmentação *EM/MPM* baseada em *Markov Random Fields* possui alguns parâmetros que precisam ser ajustados para permitir uma segmentação satisfatória. Para isso, diferentes experimentos foram realizados nas imagens de ninhais, variando os valores dos parâmetros. Na prática, considera-se os seguintes parâmetros: número de classes  $L$ , número de iterações  $P$ , número de visitas por pixel  $T$  e número de vizinhos  $k$  (4 ou 8 conectados). Caso o *Simulated Annealing* seja usado, deve-se definir o valor inicial de  $\beta^*$ ,  $\beta_0$ , o incremento  $I$  e o valor máximo de  $\beta^*$ ,  $\beta_{max}$ .

Os diversos experimentos resultaram nas seguintes conclusões: (1) a estimativa inicial dos parâmetros por meio de limiarização e o uso do *Simulated Annealing* geraram resultados mais precisos, (2) o número de visitas por pixel foi fixado como  $T = 3$ , dado que o uso de valores maiores implica em um grande aumento do custo computacional por iteração sem muito ganho na qualidade da segmentação, (3) o número de vizinhos  $k = 8$  e (4) o número de iterações  $P = 200$  geraram resultados satisfatórios, sem elevar o custo. Já em relação aos parâmetros do *Simulated Annealing*, percebeu-se que  $\beta_0 = 0$ ,  $I = 0.05$  e  $\beta_{max} = 2.4$  são valores que melhoram a qualidade dos resultados, ou seja, a identificação das aves se torna mais precisa. Por fim, como a maioria dos pixels presentes nas imagens de ninhais representam vegetação e são mais escuros (quando a imagem é convertida do modelo RGB para nível de cinza), a técnica de segmentação *MRF* passou a ser aplicada, sem perda da qualidade da segmentação, apenas aos pixels  $s|s \geq 80$ , reduzindo ainda mais o tempo de processamento.

Em relação ao número de classes, os experimentos foram realizados com  $k = 3, 4, 5$ , porém os melhores resultados foram obtidos com  $k = 5$ . A Figura 6.5 apresenta alguns exemplos da aplicação do algoritmo de segmentação *EM/MPM* usando *MRF* em imagens de ninhais, considerando os valores de parâmetros mencionados anteriormente. A Figura 6.6 mostra a aplicação de zoom em uma região da imagem apresentada na Figura 6.5(b), permitindo uma melhor visualização das aves, identificadas na cor cinza clara.

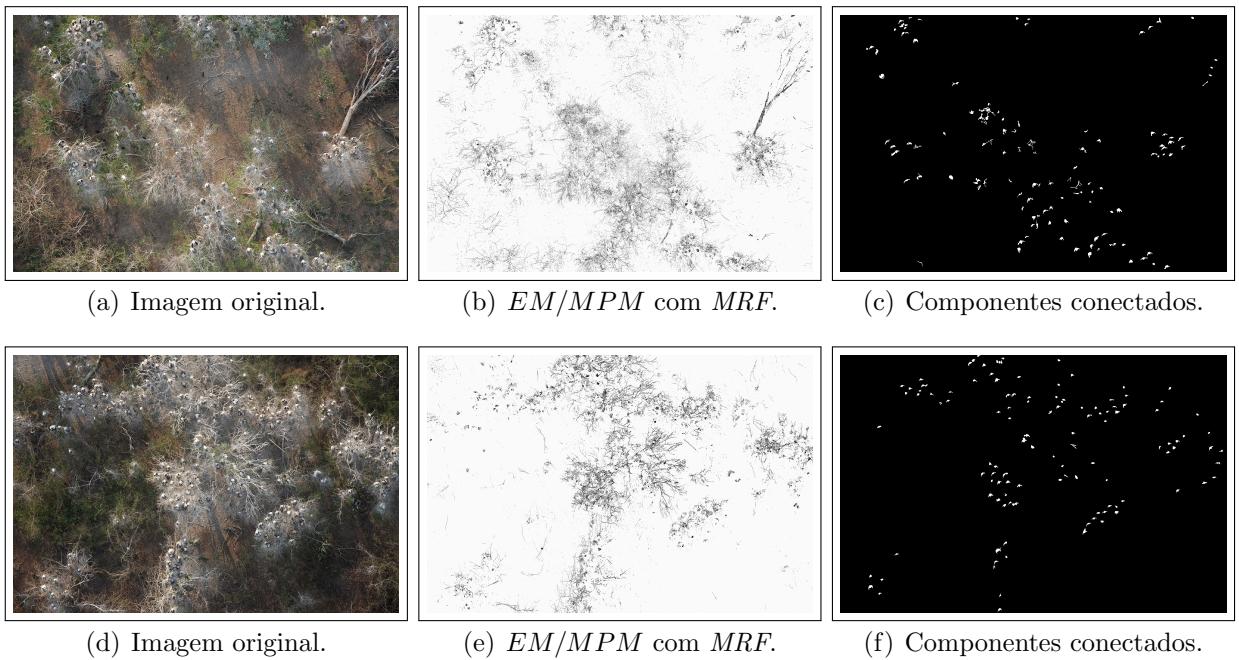


Figura 6.5: Amostra dos resultados da aplicação da segmentação *EM/MPM* usando *MRF* em imagens de ninhais de aves.

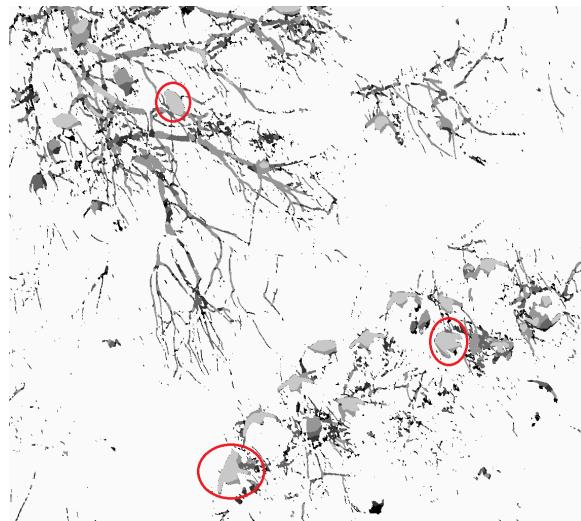


Figura 6.6: Zoom de uma região da imagem apresentada na Figura 6.5(b). Aves identificadas pela cor cinza claro, marcada pelo círculo vermelho.

#### 6.4.2 Experimento 2: Comparação da segmentação de imagens de ninhais considerando diferentes escalas

Algumas imagens do conjunto contém um ninhal completo, porém, para conseguir captá-lo é necessário sobrevoa-lo à uma altitude maior, o que reduz o tamanho das aves presentes na ima-

gem. Logo, este experimento busca verificar a influência de escalas diferentes na segmentação automática das imagens. Para isso, a imagem apresentada na Figura 6.7 foi dividida em nove sub-imagens e a técnica *EM/MPM* com *MRF* foi executada em cada uma delas. Para efeito de comparação, um zoom de 100% foi aplicado em cada uma das nove imagens e o processo de segmentação foi repetido.

A Figura 6.8 apresenta uma amostra dos resultados da segmentação, considerando a imagem original e a imagem com zoom. Ao analisar visualmente as imagens é possível perceber que nas imagens originais (sem zoom), a técnica demonstrou dificuldade para identificar corretamente as aves, ou seja, várias aves foram unidas em um grupo, como pode ser observado nas Figuras 6.8(b), (e) e (h). Já nas imagens com zoom este problema não ocorreu, conforme visto nas Figuras 6.8(c), (f) e (i), tornando a identificação das aves mais precisa.



Figura 6.7: Imagem de um ninhal da Fazenda Sangradouro, no Pantanal.

#### 6.4.3 Experimento 3: Comparação da contagem automática de aves com a contagem manual

Este experimento tem por objetivo comparar a contagem automática, realizada pela segmentação baseada em *MRF*, com a contagem manual. Tradicionalmente, a contagem de aves para fins de censo demográfico é feita *in loco*, por biólogos especialistas. Estes visitam os ninhais e, por observação, contam o número de indivíduos existentes. Neste trabalho, por acreditar ser um espécie de contagem mais apropriada que a contagem *in loco*, considerou-se também a contagem manual das aves visualizadas nas imagens em que se aplicou o método automático. Os resultados apresentados neste experimento levam em consideração ambas as contagens (*in loco* e nas próprias imagens).

Inicialmente, selecionou-se uma imagem de um ninhal que havia sido analisado por uma equipe de biólogos. Este ninhal é apresentado na imagem da Figura 6.7, a qual foi dividida

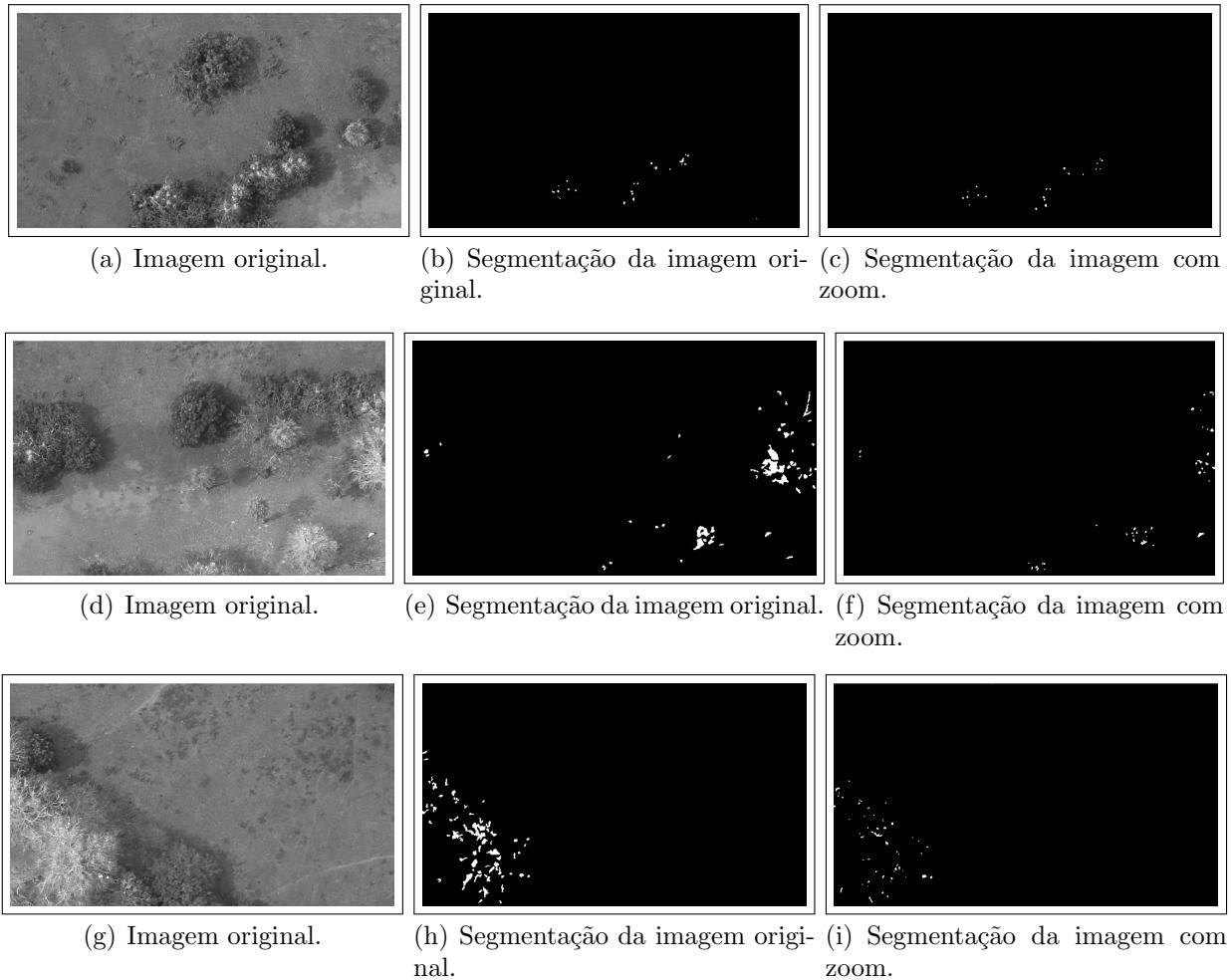


Figura 6.8: Comparação dos resultados obtidos pela técnica *EM/MPM* com *MRF*, considerando a aplicação de zoom nas imagens.

em nove sub-imagens e um zoom de 100% foi aplicado em cada uma delas, pois percebeu-se que os resultados são mais acurados, como mostrado na seção 6.4.2. Em seguida, a técnica *EM/MPM* baseada em *MRF* foi aplicada e as aves foram identificadas. Observou-se que, em algumas situações, objetos “não aves” (como galhos cobertos de detritos e outras estruturas de intensidade de cinza semelhante) foram erroneamente classificadas como aves. Isso requereu um pós-processamento baseado simplesmente na quantidade de pixels destes objetos, ou seja, objetos com um determinado número de pixels foram eliminados. Por fim, a contagem das aves foi realizada por meio de componentes conectados.

A segmentação automática das nove sub-imagens e a contagem das aves são apresentadas nas Figuras 6.9, 6.10 e 6.11, onde cada uma corresponde, respectivamente, à faixa superior, central e inferior da imagem apresentada na Figura 6.7. A soma dos componentes identificados

em todas as nove sub-imagens resultou em um total de 325 componentes (aves). Já a equipe da Profa. Sílvia Nassif identificou, a partir do solo, um total de 392 ninhos, dos quais 341 foram confirmados como sendo da ave *Mycteria Americana*. A Tabela 6.1 apresenta os resultados das contagens.

Tabela 6.1: Comparação da contagem *in loco* com a contagem automática das aves.

Contagem <i>in loco</i>	Contagem Automática
341	325

Cabe mencionar que a comparação da segmentação automática com a contagem *in loco* pode não ser totalmente apropriada por diversas razões como, por exemplo: (1) a contagem *in loco* não foi feita na mesma ocasião que a obtenção da imagem aérea do ninhal, com isso algumas aves poderiam não estar nos ninhos, (2) a segmentação automática também considera aves em vôo e (3) aves muito próximas podem ser identificadas como uma única ave, em alguns casos na segmentação automática. Porém, ao analisar visualmente os componentes (aves) identificados nas imagens via *MRF*, é possível notar uma alta acurácia de segmentação.

Na tentativa de obter uma comparação mais apropriada da contagem automática, via *MRF*, com a contagem manual, foram selecionadas 20 imagens de ninhais de aves e realizado o processo de segmentação e contagem das aves. Em seguida, foi feita a contagem manual das aves nas próprias imagens para comparação dos resultados. O gráfico apresentado na Figura 6.12 mostra a relação entre os valores obtidos com a contagem automática e a contagem manual das 20 imagens selecionadas e, com base no coeficiente de correlação de Pearson obtido  $\rho = 0,98$ , é possível perceber a alta correlação entre os resultados (contagem das aves). A Figura 6.13 apresenta uma amostra dos resultados, destacando a quantidade de aves obtida automaticamente e manualmente.

## 6.5 Considerações Finais

Este capítulo abordou a análise das imagens aéreas de ninhais de aves do Pantanal por meio de diferentes experimentos. Percebeu-se que esta análise não é uma tarefa fácil devido à limitação do conjunto de imagens obtido, às características peculiares das imagens (presença de dejetos na mesma tonalidade das aves) e à sua alta resolução. No entanto, os resultados da contagem automática das aves foram considerados bastante promissores, o que é comprovado pela comparação com a contagem manual.

Cabe ressaltar que foram realizados dois tipos de contagem manual para fins de comparação. A primeira contagem manual, chamada de contagem *in loco*, foi estimada por biólogos especi-

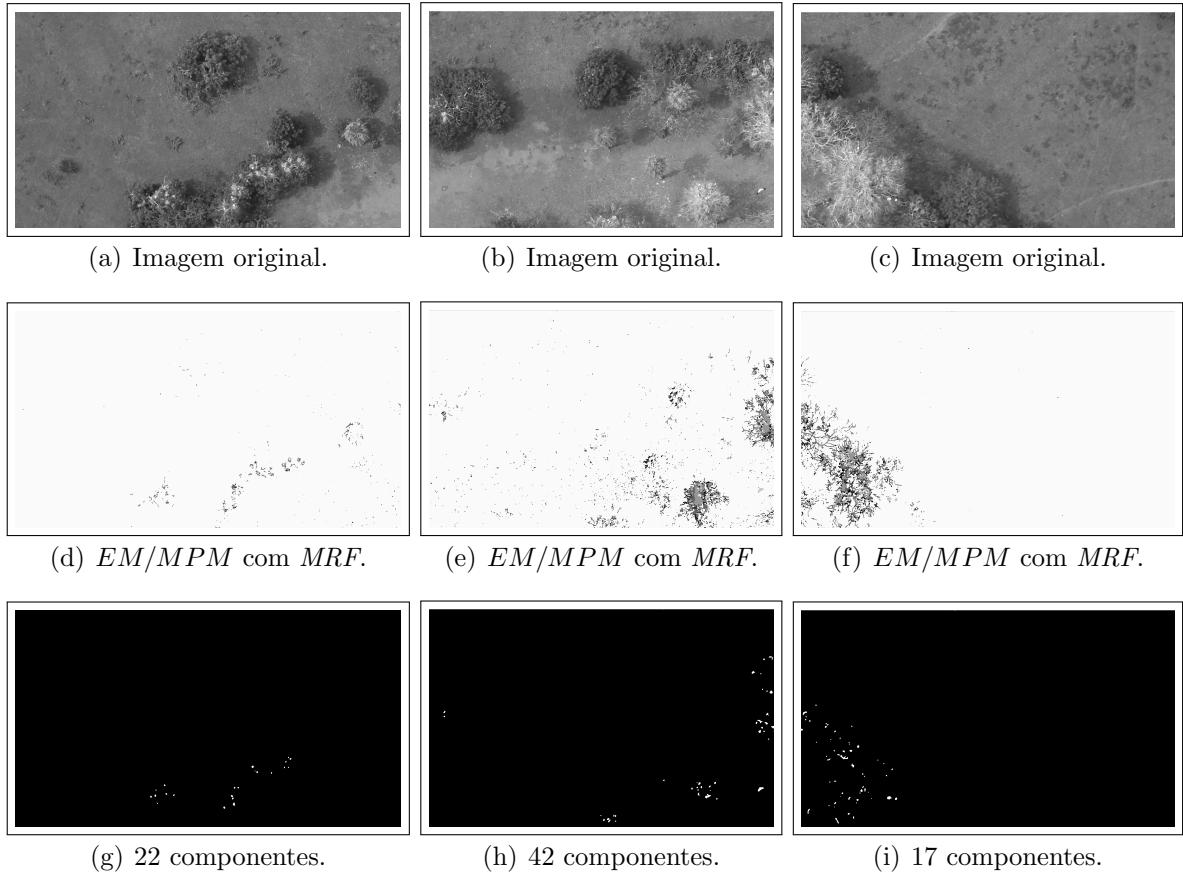


Figura 6.9: Resultados das sub-imagens que representam a faixa superior da imagem apresentada na Figura 6.7.

alistas por meio da observação das aves nos próprios ninhais. Este é o processo de contagem padrão atualmente adotado por profissionais da área para o censo populacional de aves. Esta técnica é um tanto subjetiva dado que é realizada com indivíduos (aves) que se deslocam de um lugar a outro durante a contagem e que pode levar vários minutos (ou horas) para que todo o ninhal seja percorrido. Além disso, muitas vezes impossibilitado de observar 100% do ninhal, o profissional faz projeções e estimativas do número total de indivíduos baseado na densidade das aves até então observadas. Portanto, uma comparação com o método automático baseado em imagens, sob estas condições, tem boas chances de ser pouco significativo ou pouco apropriado. Desta forma, optou-se por realizar uma contagem manual das aves nas próprias imagens dos ninhais em que foram feitas as contagens automáticas via segmentação.

Considerando os resultados de ambas as contagens manuais (*in loco* e nas próprias imagens) para comparação, percebe-se que a técnica de segmentação automática via *MRF* alcança resultados de contagens semelhantes, demonstrando sua capacidade de captar microtexturas (aves). Além disso, ao observar visualmente as segmentações obtidas é possível perceber que a

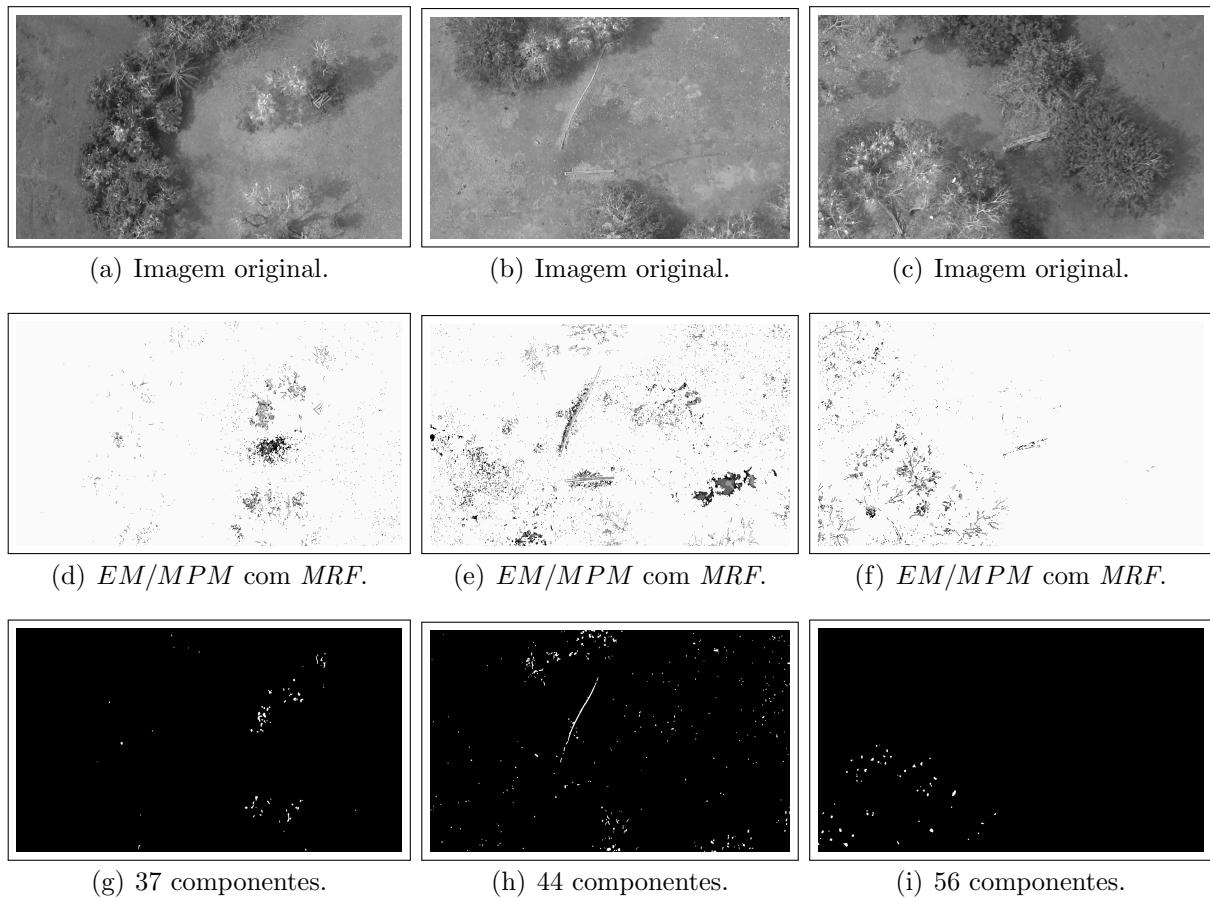


Figura 6.10: Resultados das sub-imagens que representam a faixa central da imagem apresentada na Figura 6.7.

maioria das aves foram identificadas corretamente e atribuídas à mesma classe.

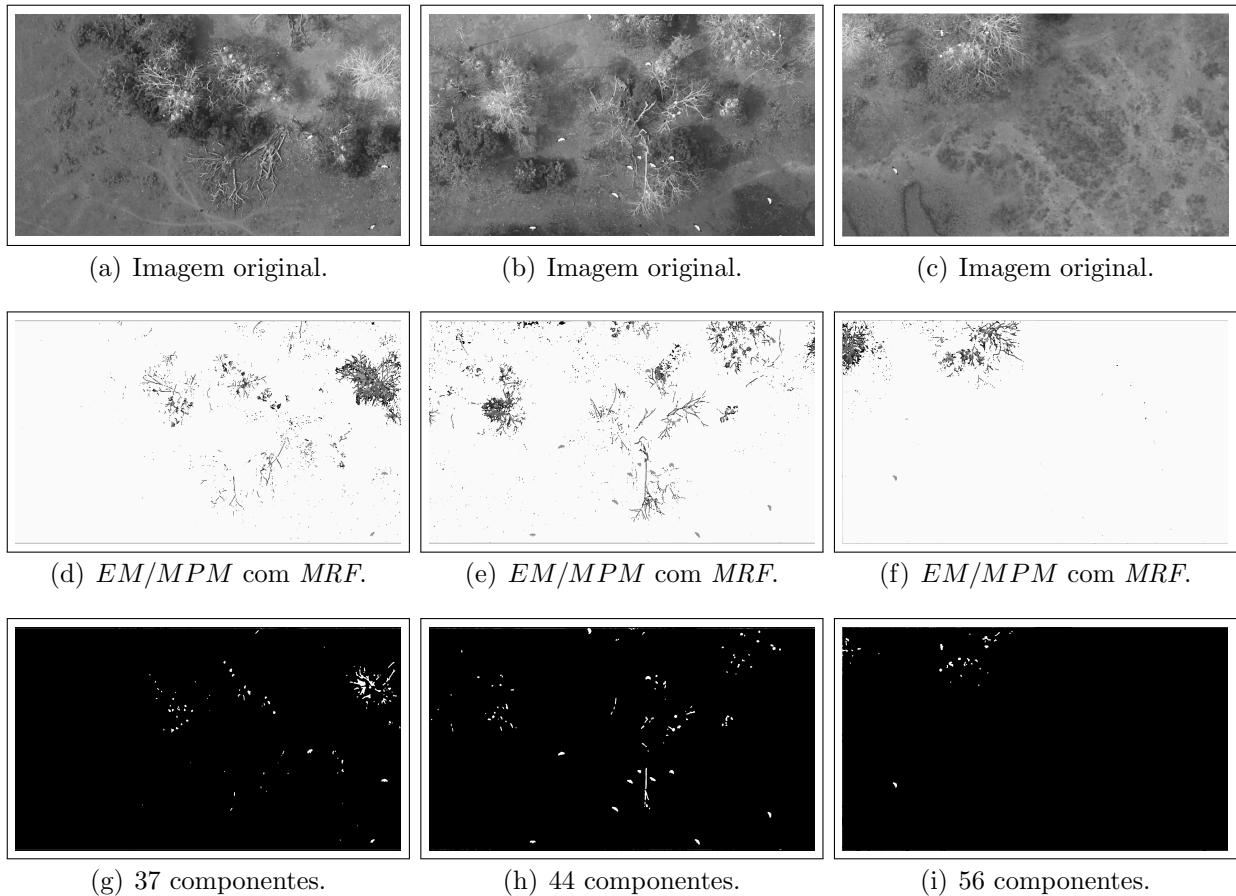


Figura 6.11: Resultados das sub-imagens que representam a faixa inferior da imagem apresentada na Figura 6.7.

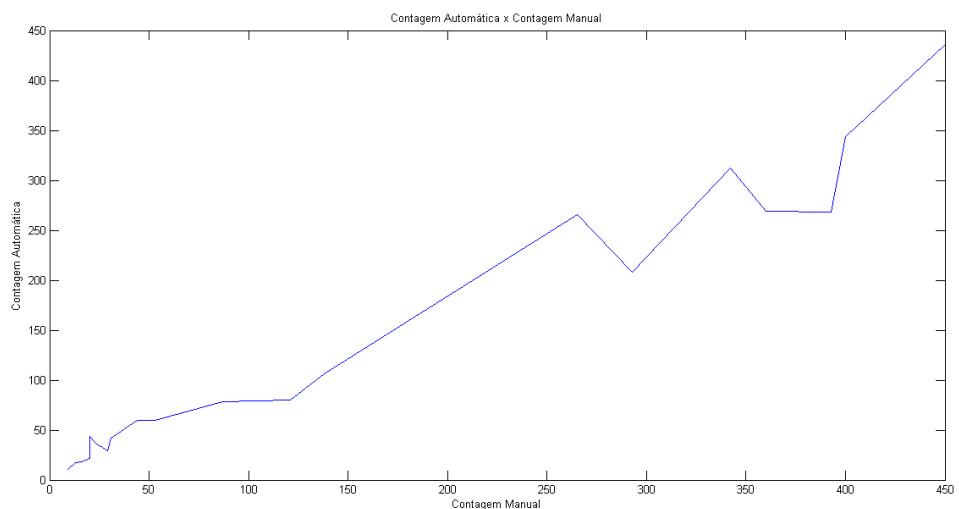


Figura 6.12: Relação entre a contagem automática e a contagem manual.

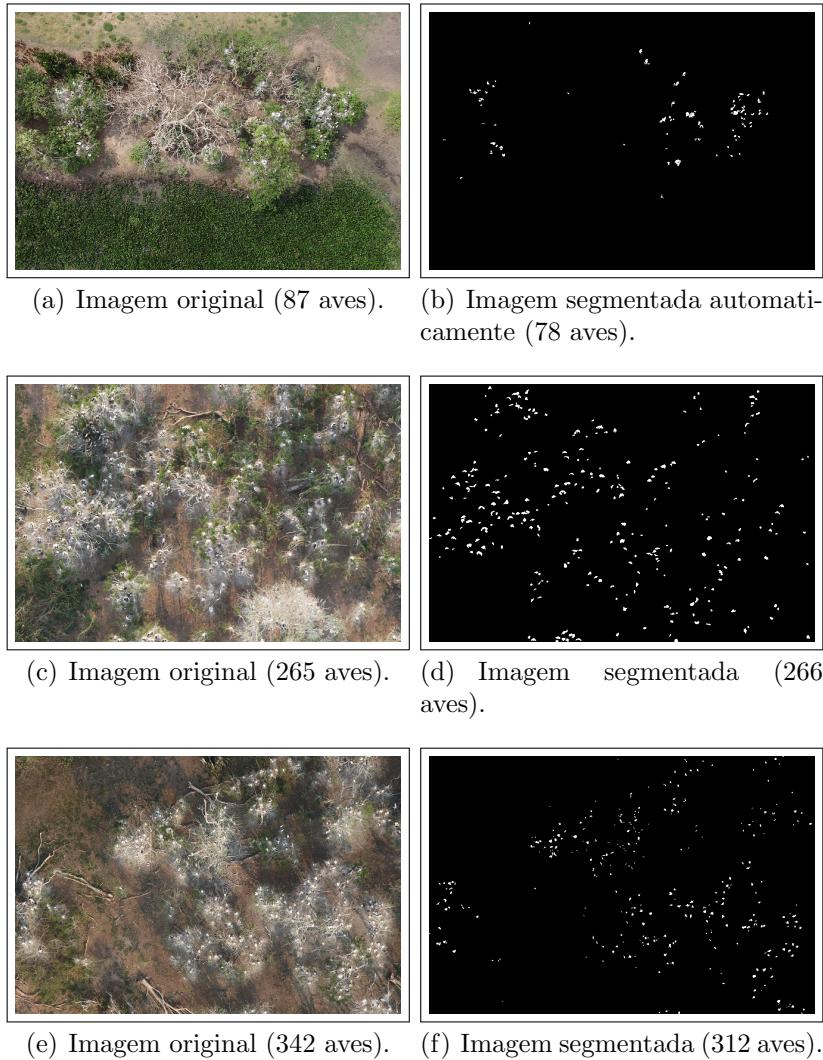


Figura 6.13: Comparaç $\text{\~ao}$  da segmenta $\text{\~ao}$  autom $\text{\'atica}$  usando *MRF* com a segmenta $\text{\~ao}$  manual obtida pela contagem das aves nas pr $\text{op}rias$  imagens. Os valores entre par $\text{enteze}$  representam a quantidade de aves.

# Conclusões

Este trabalho desenvolveu uma nova técnica de segmentação que combina algoritmos de detecção de comunidades, pertencentes à teoria das redes complexas, com técnicas de extração de *superpixels*. Para isso, diversas modificações foram realizadas para se trabalhar com grafos de *superpixels*, os quais representam as imagens. Novas funções de pesos foram desenvolvidas, além de um amplo estudo para definir os melhores valores dos parâmetros usados na extração dos *superpixels* e geração do grafo. Esta combinação mostrou-se bastante promissora no quesito acurácia de segmentação e tempo de processamento.

Como um dos objetivos do trabalho foi o de realizar o censo demográfico de aves por meio da análise de imagens aéreas de ninhais, a técnica de segmentação por textura *EM/MPM* baseada em *Markov Random Fields (MRF)* foi implementada e complementou a abordagem de segmentação citada anteriormente. Esta técnica foi escolhida devido à sua capacidade de capturar microtexturas, permitindo uma melhor diferenciação das aves e dos dejetos, ambos na cor branca. Para aprimorar ainda mais os resultados, a abordagem de *Simulated Annealing* foi incluída na técnica *MRF* e os valores iniciais dos parâmetros passaram a ser estimados por Limiarização usando a técnica de Otsu. Os resultados mostraram que a identificação e contagem automática das aves nas imagens foi bastante satisfatória, com coeficiente de Pearson  $\rho = 0.98$ , quando comparada com a contagem manual (*in loco* e nas próprias imagens).

Por fim, devido à importância de avaliar quantitativamente (e não apenas visualmente como é feito na maioria das vezes) a qualidade das segmentações obtidas, propôs-se uma nova métrica quantitativa de avaliação baseada em *ground-truth*. Para viabilizar este processo, também foi desenvolvida uma técnica de seleção da imagem de referência (entre as várias segmentações manuais providas por diferentes usuários), que será comparada com o resultado da segmentação automática. A métrica de avaliação juntamente com a técnica de seleção da imagem de referência são contribuições deste trabalho para o campo de segmentação de imagens.

## 7.1 Limitações

A segmentação de imagens é uma das tarefas mais complexas da análise de imagens devido à subjetividade implícita do processo, à grande quantidade de imagens diferentes e à diversidade das propriedades que podem ser analisadas como, por exemplo, cor, textura, forma etc. Assim, a abordagem de segmentação proposta, baseada em algoritmos de detecção de comunidades e *superpixels*, também apresenta algumas limitações. Primeiramente, esta abordagem é baseada em informações de cores e, consequentemente, é dependente da cor das regiões. Logo, podem existir regiões diferentes com a mesma cor, as quais serão consideradas como pertencentes à mesma comunidade. Além disso, os resultados desta abordagem são dependentes dos valores dos parâmetros usados e caso altere completamente o banco de imagens como, por exemplo, usar imagens em tons de cinza, pode-se ter que realizar um novo estudo para definir os melhores valores para alguns parâmetros. Em relação à segmentação das imagens aéreas de ninhais de aves, em regiões muito esbranquiçadas, onde aves e dejetos se misturaram, a técnica apresentou dificuldade para delimitar apropriadamente as bordas das aves. Além disso, aves muito próximas ou sobrepostas são identificadas como um único objeto, prejudicando a contagem correta dos indivíduos.

## 7.2 Contribuições

As principais contribuições deste trabalho são:

- Proposição de uma nova abordagem de segmentação que combina algoritmos de detecção de comunidades, pertencentes à teoria das redes complexas, com técnicas de extração de *superpixels*. Como a maioria dos algoritmos de detecção de comunidades apresentam custo computacional elevado, sua aplicação é limitada à imagens de baixa resolução. No entanto, com o uso das técnicas de extração de *superpixels*, a imagem passa a ser representada por um grafo de *superpixels*, e não de pixels, reduzindo a cardinalidade do grafo e, consequentemente, o custo computacional da aplicação de algoritmos de detecção de comunidades.
- Amplo estudo para definição de valores adequados para os parâmetros do processo de extração de *superpixels* e geração do grafo que representa a imagem. Neste contexto, destacam-se (1) a definição de novas funções de peso, responsáveis pelo estabelecimento das conexões entre os vértices do grafo, considerando as características de cor e textura dos *superpixels*, e (2) a proposição do *threshold adaptativo*, parâmetro fundamental no

processo de geração do grafo, o qual representa a imagem, responsável pela eliminação do problema de vértices desconectados.

- Desenvolvimento de uma nova métrica quantitativa, baseada em *ground-truth*, para avaliação da qualidade das imagens segmentadas. Tal métrica também serviu de base para a criação de um novo método de seleção de uma imagem de referência, dentre as diferentes imagens segmentadas manualmente pelos usuários, para ser comparada com a segmentação automática.
- Desenvolvimento de um método de contagem automática de aves por meio da análise de imagens aéreas de ninhais usando técnicas de segmentação que, com base nos resultados visuais e no coeficiente de Pearson obtido, se mostraram apropriadas para a aplicação. Cabe ressaltar que a contagem automática de aves por meio de imagens aéreas ainda é pouco estudada na literatura.

### 7.3 Trabalhos Futuros

Em relação à abordagem de segmentação baseada em redes complexas e *superpixels* pretende-se estudar a modificação da função de convergência dos *superpixels*, para que esta possa utilizar informações diferentes da cor. Além disso, é importante estudar e propor novas funções de peso, responsáveis pela geração do grafo de *superpixels*, considerando outros tipos de características, inclusive métodos de extração de textura com custo computacional menor que o *LBP*.

Considerando a métrica quantitativa de avaliação da qualidade de segmentação percebe-se a necessidade de uma melhor formulação matemática. Por fim, deve-se concentrar esforços na segmentação das imagens aéreas de ninhais, realizando uma análise quantitativa mais aprofundada, com um maior número de imagens, para confirmar os resultados obtidos.

# Referências Bibliográficas

- [Achanta et al. 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). SLIC Superpixels Compared to State-of-the-Art Superpixels Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.
- [Achanta et al. 2010] Achanta, R., Shaji, A. Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2010). SLIC Superpixels. *EPFL Technical Report 149300*.
- [Angelini et al. 2007] Angelini, L., Marinazzo, D., Pellicoro, M., and Stramaglia, S. (2007). Natural Clustering: the Modularity Approach. *Journal of Statistical Mechanics: Theory and Experiment, IOP Publishing*.
- [Ankenbrandt et al. 1990] Ankenbrandt, B. P., Buckles, B. P., and Petry, F. E. (1990). Scene Recognition using Genetic Algorithms with Semantic Net. *Pattern Recognition Letters*, 11:285–283.
- [Arbeláez et al. 2009] Arbeláez, P., Maire, M., Fowlkes, C., and Malik, J. (2009). From Contours to Regions: An Empirical Evaluation. *Computer Vision and Pattern Recognition*, pages 2294–2301.
- [Balan 2003] Balan, A. G. R. (2003). Técnicas de Segmentação de Imagens Aéreas para a Contagem de População de Aves. Master thesis, Universidade de São Paulo (USP), Instituto de Ciências Matemáticas e de Computação.
- [Besag 1974] Besag, J. (1974). Spatial Interaction and the Statistical Analysis of Lattice Systems. *Journal of the Royal Statistical Society*, 36(2):192–236.
- [Bezdeck and Pal 1992] Bezdeck, J. C. and Pal, S. K. (1992). Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data. *IEEE Press*.

- [Binh et al. 2012] Binh, H. T. T., Loi, M. D., and Thuy, N. T. (2012). Improving Image Segmentation using Genetic Algorithm. *11th International Conference on Machine Learning Applications (ICMLA)*, 2:12–15.
- [Boettcher 2001] Boettcher, S. (2001). Extremal Optimization for Graph Partitioning. *Physical Review E*, 64(026114).
- [Boettcher and Percus 2001] Boettcher, S. and Percus, A. G. (2001). Optimization with Extremal Dynamics,. *Physical Review Letter* 86, pages 5211–5214.
- [Brandes et al. 2006] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikołoski, Z., and Wagner, D. (2006). On Modularity - NP-Completeness and Beyond. Technical report 2006-19, University Karlsruhe - Faculty of Informatics. <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/3255>.
- [Cairns 1979] Cairns, D. (1979). Censusing Hole-Nesting Auks by Visual Counts. *Bird-Banding*, 50:358–364.
- [Cerny 1985] Cerny, V. (1985). Thremodynamical Approach to the Traveling Salesman Problem: An Efficient Simmulated Algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51.
- [Cigla and Alatan 2010] Cigla, C. and Alatan, A. A. (2010). Efficient Graph-based Image Segmentation via Speeded-up Turbo Pixels. *17th International Conference on Image Processing*, pages 26–29.
- [Clauset et al. 2004] Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding Community Structure in Very Large Networks. *Physical Review E*, 79(066111).
- [Comer and Delp 1995] Comer, M. L. and Delp, E. J. (1995). Multiresolution Image Segmentation. *IEEE Int. Conf. Acoustic, Speech and Sinal Processing*, pages 2415–2418.
- [Comer and Delp 2000] Comer, M. L. and Delp, E. J. (2000). The EM/MPM Algorithm for Segmentation of Textured Images: Analysis and Further Experimental Results. *IEEE Transactions on Image Processing*, 9(10):1731–1744.
- [Comer et al. 1996] Comer, M. L., Liu, S., and Delp, E. J. (1996). Statistical Segmentation of Mammograms. *Proceedings of the 3rd International Workshop on Digital Mammography*, pages 475–478.

- [Costa and Cesar Jr 2000] Costa, L. D. F. and Cesar Jr, R. M. (2000). Shape Analysis and Classification: Theory and Practice. *CRC Press*.
- [Cross and Jain 1983] Cross, G. R. and Jain, A. (1983). Markov Random Fields Texture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(1):25–39.
- [Csardi and Nepusz 2006] Csardi, G. and Nepusz, T. (2006). The Igraph Software Package for Complex Network Research. *Inter Journal, Complex Systems*. <http://igraph.sourceforge.net/index.html>.
- [Dempster et al. 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Imcomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society B*.
- [Duch and Arenas 2005] Duch, J. and Arenas, A. (2005). Community Detection Complex Networks using Extremal Optimization. *Physical Review E*, 72.
- [Felzenswalb and Huttenlocher 2004] Felzenswalb, P. and Huttenlocher, D. (2004). Efficient Graph based Image Segmentation. *International Journal on Computer Vision*, 59(4):169–181.
- [Fortunato 2010] Fortunato, S. (2010). Community Detection in Graphs. *Physics Reports - Elsevier*, 486:75–174.
- [Fu and Mui 1981] Fu, K. S. and Mui, J. K. (1981). A Survey on Image Segmentation. *Pattern Recognition*, 13:3–16.
- [Geman and Geman 1984] Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.
- [Gerhardinger 2006] Gerhardinger, L. C. (2006). Segmentação de Imagens e Validação de Classes por Abordagem Estocástica. Master thesis, Universidade de São Paulo (USP), Instituto de Ciências Matemáticas e de Computação.
- [Giordana and Pieczynski 1997] Giordana, N. and Pieczynski, W. (1997). Estimation of Generalized Multisensor Hidden Markov Chains and Unsupervised Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):465–475.

- [Girvan and Newman 2002] Girvan, M. and Newman, M. E. J. (2002). Community Structure in Social and Biological Networks. *Proceedings of the National Academy of Science USA*, 99(12):7821–7826.
- [Gonzalez et al. 1990] Gonzalez, M., Benitez, D., and Suarez, C. P. (1990). Segmentation and Recognition in Visual Chromatic Spaces. *Cybernetics Systems*, 21:241–247.
- [Gonzalez and Woods 2010] Gonzalez, R. C. and Woods, R. E. (2010). *Processamento de Imagens Digitais*. Editora Pearson/Prentice Hall, 3 edição edition.
- [Granell et al. 2011] Granell, C., Gómez, S., and Arenas, A. (2011). Mesoscopic Analysis of Networks: Applications to Exploratory Analysis and Data Clustering. *Chaos*, 21(1).
- [Guimerà et al. 2004] Guimerà, R., Sales-Pardo, M., and Amaral, L. A. N. (2004). Modularity from Fluctuations in Random Graphs and Complex Networks. *Physical Review E* 70, 2.
- [Haralick and Shapiro 1985] Haralick, R. M. and Shapiro, L. G. (1985). Image Segmentation Techniques. *Computer Vision, Graphics and Image Processing*.
- [Jeff Wu 1983] Jeff Wu, C. F. (1983). On the Convergence Properties of the EM Algorithm. *The Annals Statistics*, 11(1):95–103.
- [Karasulu and Balli 2010] Karasulu, B. and Balli, S. (2010). Image Segmentation using Fuzzy Logic, Neural Networks and Genetic Algorithms: Survey and Trends. *Machine Graphics and Vision International Journal*, 19(4):367–409.
- [Kindermann and Snell 1980] Kindermann, R. and Snell, J. L. (1980). *Markov Random Fields and their Applications*, volume 1. American Mathematical Society.
- [Kirkpatrick et al. 1983] Kirkpatrick, C. D., Gelatt, D., and Vecchi, M. P. (1983). Optimization by Simmulated Annealing. *Science*, 220:671–680.
- [Lakshmanan and Derin 1989] Lakshmanan, S. and Derin, H. (1989). Simultaneous Parameter Estimation and Segmentation of Gibbs Random Fields Using Simulated Annealing. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 11(8):799–813.
- [Lama et al. 2000] Lama, S. D., Lopes, I., and Lama, M. D. (2000). Genetic Variability and Level of Differentiation in Wood Stork Populations. *IBIS*.
- [Levinshtein et al. 2009] Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddigi, K. (2009). Turbopixels: Fast Superpixels using Geometric Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:2290–2297.

- [Linares et al. 2012] Linares, O. A. C., Botelho, G. M., Rodrigues, F., and Batista Neto, J. E. S. (2012). Segmentation of Large Images with Complex Networks. *Proceedings of the 25th Conference on Graphics, Patterns and Images - SIBGRAPI*.
- [Loarie et al. 2009] Loarie, S. R., Duffy, P. B., Hamilton, H., Asner, G. P., Field, C. B., and Ackerly, D. D. (2009). The Velocity of Climate Change. *Journal Nature*, 462:1052–1055.
- [Lucchese and Mitra 2001] Lucchese, L. and Mitra, K. (2001). Color Image Segmentation: A State-of-Art Survey. *Proc. Indian Nat. Sci. Acad. (INSA-A)*, 67-A:207–221.
- [Maenpaa 2003] Maenpaa, T. (2003). The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications. Dissertation, Faculty of Technology, University of Oulu.
- [Marroquin et al. 1987] Marroquin, J., Mitter, S., and Poggio, T. (1987). Probabilistic Solution of Ill-posed Problems in Computational Vision. *Journal of the American Statistical Association*, 82(397):76–89.
- [Martin et al. 2001] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). 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*, volume 2, pages 416–423.
- [Materka and Strzelecki 1998] Materka, A. and Strzelecki, M. (1998). Texture Analysis Methods: A Review. Dissertation, Technical University of Lodz.
- [Metropolis et al. 1973] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1973). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092.
- [Newman 2004] Newman, M. E. J. (2004). Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E*, 69(066133).
- [Newman 2006] Newman, M. E. J. (2006). Finding Community Structure in Networks using the Eigenvectors of Matrices. *Physics/0605087*.
- [Newman and Girvan 2004] Newman, M. E. J. and Girvan, M. (2004). Finding and Evaluating Community Structure in Networks. *Physical Review E*, 69(026113).
- [Ojala et al. 2002] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.

- [Oliveira et al. 2008] Oliveira, B. S., Zhao, L., Faceli, K., and Carvalho, A. (2008). Data Clustering based on Complex Network Community Detection. *IEEE Congress on Evolutionary Computation (CEC)*, pages 2121–2126.
- [Otsu 1979] Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histogram. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66.
- [Pal and Pal 1993] Pal, N. R. and Pal, S. K. (1993). A Review on Image Segmentation Techniques. *Pattern Recognition*, 26(9):1277–1294.
- [Pappas 1992] Pappas, T. N. (1992). An Adaptive Clustering Algorithm for Image Segmentation. *IEEE Transactions on Signal Processing*, 40:901–914.
- [Parker 2010] Parker, J. R. (2010). Algorithms for Image Processing and Computer Vision. *Wiley.com*.
- [Qing et al. 2005] Qing, X., Jie, Y., and Siyi, D. (2005). Texture Segmentation using LBP embedded Region Competition. *Electronic Letters on Computer Vision and Image Analysis*, 5(1):41–47.
- [Radicchi et al. 2004] Radicchi, C., Castellano, F., Cecconi, F., Loreto, V., and Parisi, D. (2004). Defining and Identifying Communities in Networks. *Proceedings of the National Academy of Science USA*, 101(9):2658–2663.
- [Raghavan et al. 2007] Raghavan, U. S., Albert, R., and Kumara, S. (2007). Near Linear Time Algorithm to Detect Community Structures in Large-scale Networks. *Physical Review E* 76.
- [Redner and Walker 1984] Redner, R. A. and Walker, H. F. (1984). Mixture Densities, Maximum Likelihood and the EM Algorithm. *Society for Industrial and Applied Mathematics Review*, 26(2):195–239.
- [Reed and Hans 1993] Reed, T. R. and Hans, J. M. a. (1993). A Review of Recent Texture Segmentation and Feature Extraction Techniques. *Image Understanding*, 57(3):359–372.
- [Ren and Malik 2003] Ren, X. and Malik, J. (2003). Learning a Classification Model for Segmentation. *Proceedings on the Ninth IEEE International Conference on Computer Vision*, 1:10–17.
- [Rodrigues et al. 2011] Rodrigues, F. A., de Arruda, G. F., and Costa, L. F. (2011). A Complex Networks Approach for Data Clustering,. *Arxiv preprint arXiv:1101.5141*.

- [Rosholm 1997] Rosholm, A. (1997). *Statistical Methods for Segmentation and Classification of Images*. Ph.d. thesis, Technical University of Denmark.
- [Shi and Malik 2000] Shi, J. and Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.
- [Sonka 1999] Sonka, M. (1999). *Image Processing, Analysis and Machine Vision*. PWS.
- [Theodoridis and Koutroumbas 2003] Theodoridis, S. and Koutroumbas, K. (2003). *Pattern Recognition*. Academic Press.
- [Torbati et al. 2014] Torbati, N., Ayatollahi, A., and Kermani, A. (2014). An Efficient Neural Network based Method for Medical Image Segmentation. *Computers in Biology and Medicine*, 44:76–87.
- [Tripathi et al. 2012] Tripathi, S., Kumar, K., Singh, B. K., and Singh, R. P. (2012). Image Segmentation: A Review. *International Journal of Computer Science and Management Research*, 1(4):838–843.
- [Xi 2010] Xi, Q. (2010). An Improved Image Segmentation Algorithm Base on Normalized Cut. *2nd International Conference on Computer Engineering and Technology*, 7:293–296.
- [Zhang et al. 2008] Zhang, H., Fritts, J. E., and Goldman, S. A. (2008). Image Segmentation Evaluation: A Survey of Unsupervised Methods. *Computer Vision and Image Understanding*, Elsevier, 110(2):260–280.
- [Zhenguo et al. 2012] Zhenguo, L., Wu, X., and Chang, S. (2012). Segmentation using Superpixels: a Bipartite Graph Partitioning Approach. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 789–796.