



Caroline Rosa Redlich

Segmentação de Imagens Baseada em Grafos de Superpixel

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio.

Orientador: Prof. Marcelo Gattass

Rio de Janeiro
Abril de 2018



Caroline Rosa Redlich

Segmentação de Imagens Baseada em Grafos de Superpixel

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marcelo Gattass

Orientador

Departamento de Informática – PUC-Rio

Prof. Hélio Côrtes Vieira Lopes

Departamento de Informática – PUC-Rio

Prof. Alberto Barbosa Raposo

Departamento de Informática – PUC-Rio

Prof Márcio da Silveira Carvalho

Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 19 de Abril de 2018

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Caroline Rosa Redlich

Graduada em Engenharia da Computação pela Pontifícia Universidade Católica do Rio de Janeiro (Rio de Janeiro, Brasil) em 2014.

Ficha Catalográfica

Redlich, Caroline Rosa

Segmentação de Imagens Baseada em Grafos de Superpixel / Caroline Rosa Redlich; orientador: Marcelo Gattass. – Rio de Janeiro: PUC-Rio, Departamento de Informática, 2018.

v., 74 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia

1. Informática – Teses. 2. Segmentação de imagens;. 3. Superpixels;. 4. Grafos;. 5. K-Means;. I. Gattass, Marcelo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

À vó Tereza e ao vô Dorval

Agradecimentos

Ao meu professor e orientador Marcelo Gattass pelos ensinamentos e orientação desde a graduação.

À minha família. Em especial meus pais Denise Rosa e Richard Redlich pelo apoio e amor incondicional durante toda a vida e à minha irmã Luciana Redlich por sempre acreditar em mim.

Ao meu namorado Matheus Amann, pela paciência e compreensão durante todos esses anos estressada e angustiada com o mestrado.

Aos meus amigos da PUC, especialmente Rustam Mesquita, Suellen Motta, Erica Riello e Pedro Rocha, que nunca mediram esforços para me ajudar e foram de suma importância

Ao Jéferson Coêlho, que exerceu informalmente o papel de meu coordenador desde a graduação.

Às minhas amigas, especialmente Lais Reis, Danielle Leite, Camila Villar e Raiane Oliveira, por tornarem meus dias mais alegres.

Aos meus amigos do 6^o andar e do v3o2, especialmente à minha chefe Letícia Nogueira que permitiu que eu me ausentasse para a conclusão do trabalho.

Ao CNPq, PUC-Rio e Tecgraf, pelo auxílio concedido, sem o qual este trabalho não poderia ter sido realizado.

Muito obrigada!

Resumo

Redlich, Caroline Rosa; Gattass, Marcelo. **Segmentação de Imagens Baseada em Grafos de Superpixel**. Rio de Janeiro, 2018. 74p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A segmentação de imagens com objetivo de determinar a forma de objetos é ainda um problema difícil. A separação de regiões que correspondem a objetos contidos na imagem geralmente leva em consideração propriedades de similaridade, proximidade e descontinuidade. A imagem a ser segmentada pode ser de diversas naturezas, como fotografias, imagens médicas e sísmicas. Podemos encontrar na literatura muitos métodos de segmentação propostos como possíveis soluções para diferentes problemas. Recentemente a técnica de superpixel tem sido utilizada como um passo inicial que reduz o tamanho da entrada do problema. Este trabalho propõe uma metodologia de segmentação de imagens fotográficas e de ultrassom que se baseia em variantes de superpixels. A metodologia proposta se adapta a natureza da imagem e a complexidade do problema utilizando diferentes medidas de similaridade e distância. O trabalho apresenta também resultados que buscam esclarecer o procedimento proposto e a escolha de seus parâmetros.

Palavras-chave

Segmentação de imagens; Superpixels; Grafos; K-Means;

Abstract

Redlich, Caroline Rosa; Gattass, Marcelo (Advisor). **Image Segmentation Based on Superpixel Graphs**. Rio de Janeiro, 2018. 74p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Image segmentation for object modeling is a complex task that is still not well solved. The separation of the regions corresponding to each object in an image is based on proximity, similarity, and discontinuity of its boundaries. The image to be segmented can be of various natures, including photographs, medical and seismic images. We can find in literature many proposed segmentation methods used as solutions to different problems. Recently the superpixel technique has been used as an initial step that reduces the size of the problem input. This work proposes a methodology of segmentation of photographs and ultrasound images based on variants of superpixels. The proposed methodology adapts to the image's nature and to the problem's complexity using different measures of similarity and distance. This work also presents results that seek to clarify the proposed procedure and the choice of its parameters.

Keywords

Image segmentation; Superpixel; Graphs; K-Means;

Sumário

1	Introdução	14
1.1	Estrutura da dissertação	15
2	Conceitos Básicos	16
2.1	Imagem Digital	16
2.1.1	Imagens Fotográficas	17
2.1.2	Imagens de Ultrassom de Feto	19
2.2	Espaço de Cor CIELAB	19
2.3	Superpixel	20
3	Trabalhos Relacionados	22
4	Metodologia Proposta	26
4.1	Geração dos Superpixels/Supervoxels	27
4.2	Construção do Grafo	28
4.3	Clusterização dos Nós do Grafo	30
4.3.1	Métricas de distância	32
4.4	Pós-processamento	34
5	Resultados	36
5.1	Dataset MSRC	36
5.1.1	Imagem do cachorro na grama	36
5.1.2	Imagem de duas vacas no pasto	39
5.1.3	Imagem das flores sobrepostas	41
5.1.4	Imagem do banco de jardim	43
5.2	Dataset Berkeley	45
5.2.1	Imagem da mergulhadora com peixes	46
5.2.2	Imagem do campanário de igreja	49
5.2.3	Imagem da estrela do mar	52
5.2.4	Imagem do coral marinho	55
5.2.5	Imagem do menino no campo	58
5.2.6	Imagem do peixe Nemo	61
5.2.7	Imagem da mulher	64
5.3	Imagem de ultrassom	68
6	Conclusão e Trabalhos Futuros	71
	Referências Bibliográficas	73

Lista de figuras

2.1	Exemplo do processo de aquisição de uma imagem digital. Fonte: <i>Gonzalez e Woods</i> (6).	18
2.2	Imagem contínua, projetada em uma matriz e resultado da amostragem e quantização da imagem. Fonte: <i>Gonzalez e Woods</i> (6).	18
2.3	Exemplo de uma ultrassom de feto.	19
2.4	Exemplo de superpixels em uma imagem.	20
3.1	Fluxograma do SLIC. Fonte: Adaptado de <i>da Rosa</i> (4).	23
3.2	Diferença do espaço de busca do k-means e do SLIC. Fonte: Adaptado de <i>Achanta et al, 2012</i> (3).	23
3.3	Diferença do resultado da aplicação do SLIC(imagens de cima) e SLICO(imagens de baixo).	25
4.1	Fluxograma da metodologia proposta.	26
4.2	Exemplo da geração diferentes números de superpixels.	28
4.3	Vizinhos de um pixel e de um voxel, respectivamente.	30
4.4	Exemplo de grafo para uma imagem com cinco superpixels.	30
4.5	$\Delta E^* \leq 1$	32
4.6	$\Delta E^{*'} \leq 1$ com $\alpha^2 = 10$	33
4.7	Duas possíveis segmentações para a Figura 4.	34
4.8	Exemplo de imagem em que o pós-processamento pode ser interessante.	35
5.1	Imagem do cachorro na grama.	37
5.2	<i>Ground truth</i> fornecido pelo dataset para imagem do cachorro.	37
5.3	Geração de 100, 500 e 1000 superpixels, respectivamente, para a imagem do cachorro.	38
5.4	Imagem das duas vacas no pasto.	39
5.5	<i>Ground truth</i> fornecido pelo dataset para imagem das vacas.	39
5.6	Geração de 100, 500 e 1000 superpixels, respectivamente, para imagem das vacas.	40
5.7	Imagem das flores sobrepostas.	41
5.8	<i>Ground truth</i> fornecido pelo dataset para a imagem das flores.	41
5.9	Geração de 100, 500 e 1000 superpixels, respectivamente, para imagem das flores.	42
5.10	Imagem do banco de jardim.	43
5.11	<i>Ground truth</i> fornecido pelo dataset para a imagem do banco.	43
5.12	Geração de 100, 500 e 1000 superpixels, respectivamente, para imagem do banco.	44
5.13	Imagem da mergulhadora.	46
5.14	<i>Ground truth</i> fornecido pelo dataset para imagem da mergulhadora.	46
5.15	Imagem do campanário de igreja.	49
5.16	<i>Ground truth</i> fornecido pelo dataset para imagem do campanário de igreja.	49

5.17	Imagem da estrela do mar.	52
5.18	<i>Ground truth</i> fornecido pelo dataset para a imagem da estrela do mar.	52
5.19	Imagem G.	55
5.20	<i>Ground truth</i> fornecido pelo dataset para imagem da estrela do mar.	55
5.21	Imagem do menino no campo.	58
5.22	<i>Ground truth</i> fornecido pelo dataset para imagem do menino no campo .	58
5.23	Imagem do nemo.	61
5.24	<i>Ground truth</i> fornecido pelo dataset para a imagem do nemo.	61
5.25	Imagem da mulher.	64
5.26	<i>Ground truth</i> fornecido pelo dataset para a imagem da mulher.	64
5.27	Resultado após o pós-processamento para a imagem da estrela do mar.	68
5.28	Resultado após o pós-processamento para a imagem do coral marinho.	68
5.29	Algumas camadas do volume de ultrassom.	69
5.30	Geração de diferentes números de supervoxels em uma camada do volume.	69

Lista de tabelas

5.1	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do cachorro.	39
5.2	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem das vacas.	41
5.3	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem das flores.	43
5.4	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do banco.	45
5.5	Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem da mergulhadora.	46
5.6	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da mergulhadora utilizando o SLIC como segmentação inicial.	47
5.7	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da mergulhadora utilizando o SLICO como segmentação inicial.	48
5.8	Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem da igreja.	49
5.9	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da igreja utilizando o SLIC como segmentação inicial.	50
5.10	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da igreja utilizando o SLICO como segmentação inicial.	51
5.11	Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem da estrela do mar.	52
5.12	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da estrela do mar utilizando o SLIC como segmentação inicial.	53
5.13	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da estrela do mar utilizando o SLICO como segmentação inicial.	54
5.14	Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem do coral marinho.	55
5.15	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do coral utilizando o SLIC como segmentação inicial.	56
5.16	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do coral utilizando o SLICO como segmentação inicial.	57
5.17	Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem do menino.	58

5.18	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do menino utilizando o SLIC como segmentação inicial.	59
5.19	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do menino utilizando o SLICO como segmentação inicial.	60
5.20	Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem do peixe.	61
5.21	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do peixe utilizando o SLIC como segmentação inicial.	62
5.22	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do peixe utilizando o SLICO como segmentação inicial..	63
5.23	Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem da mulher.	64
5.24	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da mulher utilizando o SLIC como segmentação inicial.	65
5.25	Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da mulher utilizando o SLICO como segmentação inicial.	66
5.26	Resultado da aplicação do método apresentado em diferentes fatias.	70

*Antes de dar certo, dá errado. Dá muito errado.
Você se desespera, acha que vai enlouquecer,
perde a noção. E aí, finalmente, a felicidade
chega.*

Matheus Rocha, .

1 Introdução

A visão é um sentido primário humano e nos permite coletar informações sem a necessidade de interação física. A Visão Computacional é uma área da ciência que estuda métodos, técnicas e teorias que visam extrair e interpretar informações presentes em uma imagem. Um dos maiores objetivos da visão computacional é buscar um modelo que se aproxime da visão humana.

Uma das tarefas mais difíceis e importantes da visão computacional é a segmentação. Seu objetivo é a separação de regiões que correspondem a objetos contidos na imagem, geralmente levando em consideração propriedades de similaridade, proximidade e descontinuidade (6). Existem diversos métodos de segmentação propostos na literatura, e uma grande quantidade de possíveis aplicações para os mesmos. Em muitas aplicações, a segmentação é feita como um pré-processamento. A segmentação de objetos naturais, porém, ainda é um problema atual e difícil, ainda mais se tratando de imagens 3D e objetos naturais.

Em muitas aplicações de imagens, principalmente as que envolvem segmentação, a utilização de superpixels tem sido fundamental. Um superpixel é definido como uma região compacta de pixels similares e conexos. A medida de similaridade pode ser definida de diversas maneiras, usando cor e posição. A utilização de superpixels na entrada do problema, em substituição aos pixels pode ser vantajosa em vários sentidos, como na redução da redundância, uma vez que pixels semelhantes ficam agrupados em um mesmo superpixel, e a aderência as bordas que os superpixels tem que facilitam a segmentação. Uma vez que na segmentação a geração de superpixels é um passo normalmente de pré-processamento, eles devem ser gerados de maneira eficiente, para não impactar negativamente o desempenho do algoritmo de segmentação (4).

Este trabalho apresenta um método de segmentação simples de entender e fácil de implementar, que utiliza superpixels como uma segmentação inicial, aproveitando das vantagens que ele oferece. Após a geração dos superpixels, um grafo de adjacência é construído, e de forma gulosa, operações de junção nos nós do grafo são executadas, até que, idealmente, os nós restantes representem os objetos da imagem.

1.1

Estrutura da dissertação

Esta dissertação está dividida da seguinte forma: Alguns conceitos básicos para este trabalho são apresentados no capítulo 2. No capítulo 3 apresentamos os trabalhos relacionados. Já no capítulo 4 apresentam-se a metodologia proposta para a segmentação e alguns detalhes de sua implementação. No capítulo 5 apresentamos e discutimos os resultados obtidos aplicando o método proposto em imagens fotográficas e de ultrassom, e por fim, no capítulo 6 apresentamos algumas conclusões e propomos alguns trabalhos futuros para a continuação da pesquisa.

2

Conceitos Básicos

Apresenta-se brevemente neste capítulo alguns dos conceitos básicos considerados importantes para o entendimento deste trabalho.

2.1

Imagem Digital

Segundo *Gonzalez e Woods* (6), uma imagem pode ser definida como uma função bidimensional $f(x, y)$, em que x e y são coordenadas espaciais discretas de um plano e a amplitude f em qualquer par de coordenadas (x, y) é chamada de intensidade ou nível de cinza da imagem naquele ponto. Uma imagem digital é portanto, um conjunto finito de elementos, comumente denominados *pixels*, dispostos regularmente em uma matriz com largura e altura definidas.

Para a representação de objetos volumétricos em imagens tridimensionais, como por exemplo, imagens sísmicas e imagens de ultrassom, torna-se necessário estender a definição de *Gonzalez e Woods* (6). A imagem tridimensional é então definida como uma função $f(x, y, z)$, em que x , y e z são coordenadas espaciais discretas de um volume, onde cada elemento pode ser denominado *voxel* (um *pixel* volumétrico).

O significado físico de cada *pixel* ou *voxel* depende do tipo de imagem e de como ela foi adquirida(6). Em uma imagem fotográfica, um *pixel* é caracterizado como a combinação da quantidade de iluminação incidente na cena com a quantidade de iluminação refletida pelos objetos, denominados iluminação e reflectância, respectivamente. Em um volume sísmico, cada *voxel* possui o valor da amplitude da onda registrada pelos sensores de ondas sísmicas no processo de aquisição. Já em imagens de tomografia computadorizada, um *voxel* representa a quantidade de raios X absorvidos por um material ou parte do corpo. Em imagens de ultrassonografia, um *voxel*, por sua vez, assim como na imagem sísmica, possui também o valor da amplitude da onda sonora refletida.

Segundo Umbaugh (13) existem quatro tipos principais de imagens:

- Imagens binárias: cada pixel pode assumir valores 0 ou 1 e ocupa 1 bit.
- Imagens em tons de cinza: cada pixel é representado por apenas um valor, que indica a intensidade do brilho. A quantidade de tons possíveis depende da quantidade n de bits utilizados para representar cada pixel, e

equivale a 2^n . Tipicamente, em imagens fotográficas, são usados 8 bits por pixel, o que permite até 256 tons diferentes, e em imagens médicas são usados 12 ou 18 bits por pixel. Convencionalmente, os *pixels* com os menores valores são apresentados em preto, clareando gradativamente até o branco, que representa os maiores valores.

- Imagens coloridas: neste tipo de imagem, cada pixel é representado por pelo menos três informações, uma para cada canal de cor (RGB, por exemplo). Baseado no modelo monocromático, que utiliza com 8 bits por pixel, a imagem RGB é representada então por 24 bits por pixel, 8 para cada um dos três canais de cor.
- Imagens multiespectrais: tipicamente possuem informações fora do que é perceptível ao olho humano, como raio-X, ultravioleta e outras bandas do espectro eletromagnético. A informação é representada de forma visual mapeando as bandas espectrais para os componentes RGB.

A aquisição de imagens pode ser feita com diversos tipos de dispositivos, que utilizam diferentes mecanismos. A seguir, explicaremos brevemente o mecanismo de aquisição de imagens fotográficas (2.1.1) e de ultrassonografia (2.1.2).

2.1.1 Imagens Fotográficas

Existem diversos mecanismos de captura de imagens. Os mais conhecidos são o Scanner, o Digitalizador de Sinal de Vídeo (Frame/Vídeo Grabber) e a Câmera Digital. Neste trabalho focaremos apenas nas imagens obtidas por câmeras digitais.

O dispositivo mais usado atualmente por câmeras é chamado CCD (*Charge Coupled Device*), que atua como se fosse o filme fotográfico (11). A resposta de cada sensor CCD é proporcional à integral da energia luminosa projetada sobre a sua superfície (6).

A Figura 2.1 ilustra o processo de aquisição de uma imagem digital. A energia de uma fonte (a) de iluminação é refletida por um objeto da cena (b). O sistema de aquisição de imagens (c) coleta a energia encontrada e projeta em um plano imagem. A entrada frontal do sistema de aquisição é uma lente ótica que projeta a cena vista sobre o plano focal da lente (d). O arranjo de sensores, que coincide com o plano focal, produz saídas proporcionais à integral da luz recebida em cada sensor. Circuitos digitais e analógicos realizam uma varredura nessas saídas e as convertem em um sinal analógico, que é então digitalizado por um outro componente do sistema de aquisição de imagens. A saída é uma imagem digital (e) (6).

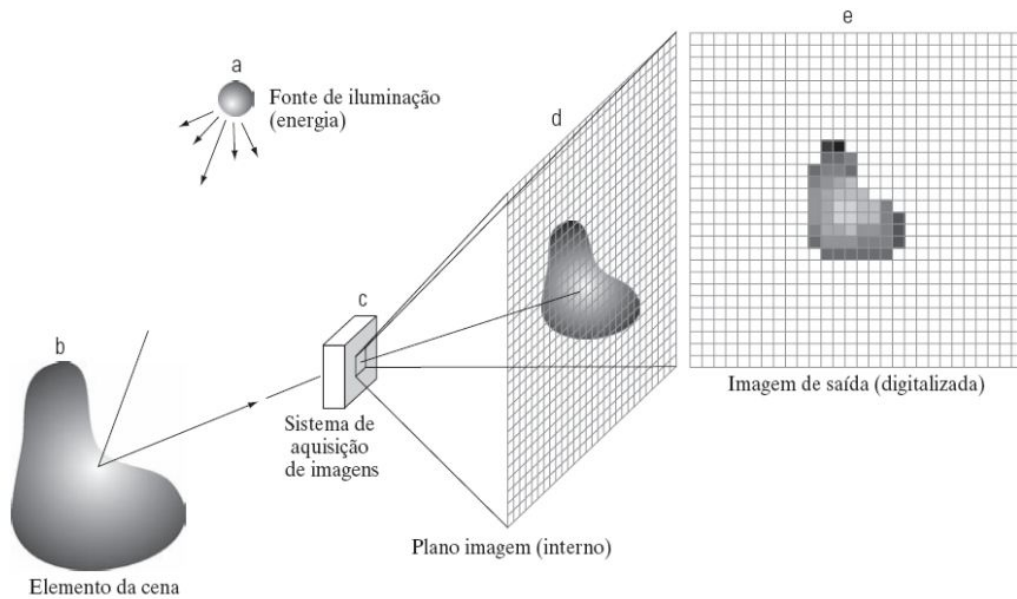


Figura 2.1: Exemplo do processo de aquisição de uma imagem digital. Fonte: *Gonzalez e Woods* (6).

A conversão do sinal analógico para uma imagem digital envolve dois processos: amostragem e quantização. Amostragem se refere a digitalização dos valores de coordenadas da imagem, enquanto a quantização se refere a digitalização dos valores de amplitude. A qualidade da imagem digital é claramente determinada, em grande parte, pelo número de amostras e de níveis discretos de intensidade utilizados nestes dois processos. A Figura 2.2 mostra a imagem após o processo de digitalização.

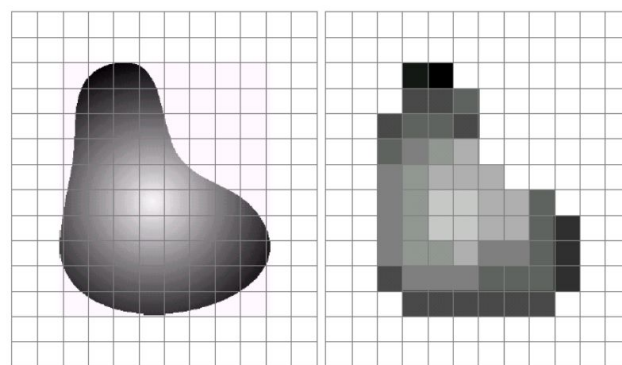


Figura 2.2: Imagem contínua, projetada em uma matriz e resultado da amostragem e quantização da imagem. Fonte: *Gonzalez e Woods* (6).

2.1.2

Imagens de Ultrassom de Feto

Segundo *Nunes* (9), a ultrassonografia é, dentre os exames de diagnóstico por imagem, um dos métodos mais utilizados, especialmente por não ser invasivo, ter boa sensibilidade e baixo custo. Por outro lado, tem como desvantagens o fato de que a qualidade da imagem depende da habilidade do operador, a baixa resolução espacial e a presença constante de ruídos na imagem.

É uma modalidade de exame médico baseada na reflexão de ondas sonoras de alta frequência, não audíveis para o ser humano. O aparelho emissor de ondas é manipulado por um operador sobre o corpo do paciente, permitindo obter imagens em tempo real. Assim que uma onda sonora encontra um tecido, uma parte dela é refletida, sendo o tempo que leva a regressar ao ponto de origem (eco) proporcional à distância a que se encontra o tecido. A amplitude do sinal de eco depende das propriedades acústicas dos tecidos e manifesta-se na imagem gerada sob a forma de diferentes intensidades no brilho produzido (12).

Nas imagens de ultrassom de feto em particular, que será a usada neste trabalho, podemos encontrar três principais elementos, que são o líquido amniótico, que aparece em tons mais escuros, e a placenta e o feto que aparecem em tons mais claros, como ilustrado na Figura 2.3.

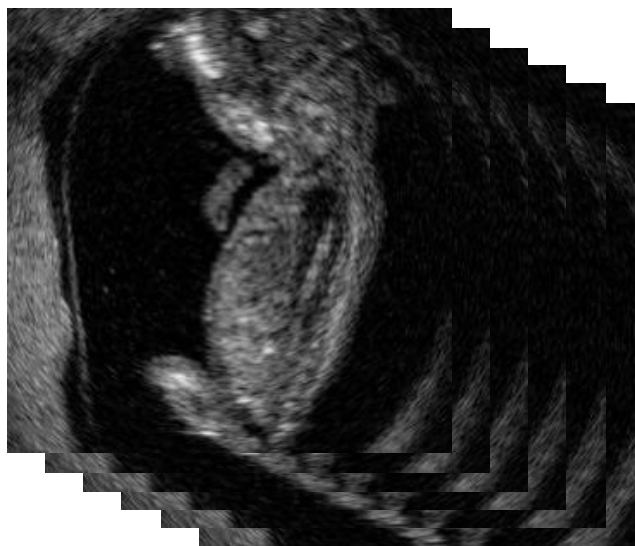


Figura 2.3: Exemplo de uma ultrassom de feto.

2.2

Espaço de Cor CIELAB

O espaço de cor $L^*a^*b^*$, também conhecido como espaço de cor CIELAB, é um espaço que procura ser perceptualmente mais uniforme que outros espaços existentes, como o RGB. Neste espaço as cores são descritas por uma coordenada

de luminosidade L^* e duas de cromaticidade a^* (conteúdo de vermelho a verde) e b^* (conteúdo de amarelo a azul).

É possível se calcular a diferença ΔE^* entre duas cores (L_1^*, a_1^*, b_1^*) e (L_2^*, a_2^*, b_2^*) como:

$$\Delta L^* = L_1^* - L_2^* \quad (2-1)$$

$$\Delta a^* = a_1^* - a_2^* \quad (2-2)$$

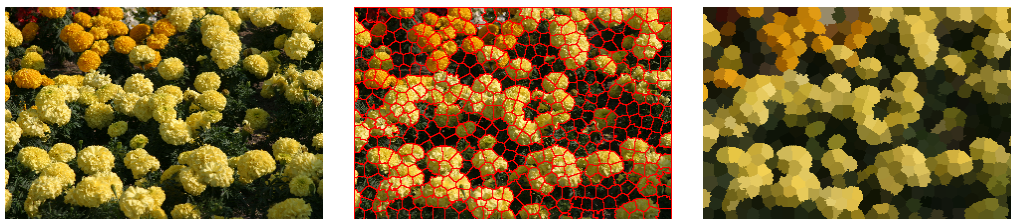
$$\Delta b^* = b_1^* - b_2^* \quad (2-3)$$

$$\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (2-4)$$

onde $\Delta E^* \approx 2.4$ corresponderia a um valor mínimo para percebermos as cores como diferentes (5).

2.3 Superpixel

Como explicado na Seção 2.1, uma imagem pode ser definida como uma grade de pixels. Porém, os pixels não são uma entidade natural e sim apenas uma consequência da representação discreta de imagens. Até em imagens com resolução média, o número de pixels é normalmente bastante alto. *Xiaofeng Ren e Jitendra Malik* (10) introduziram em 2003 o conceito de *superpixels*. Um superpixel pode ser entendido como um pequeno grupo de pixels conectados com características homogêneas. Estas características podem ser definidas como proximidade espacial e intensidade de cor. A Figura 2.4 mostra um exemplo de superpixels em uma imagem.



(a) Imagem inicial.

(b) Imagem com os superpixels.

(c) Imagem com os superpixels pintados com suas cores médias.

Figura 2.4: Exemplo de superpixels em uma imagem.

A representação da imagem como um conjunto de superpixels pode provocar uma redução significativa no tempo de processamento, uma vez que o número de superpixels é muito menor que o número de pixels, além de que é

muito mais interessante o processamento de regiões homogêneas e significativas do que pixel-a-pixel.

Segundo Achanta et al (3), algumas das propriedades normalmente desejadas para a geração de superpixels são:

1. Aderência as bordas: os superpixels devem respeitar e preservar as bordas dos objetos da imagem, uma vez que um superpixel representa uma parte de um objeto.
2. Eficiência: os superpixels devem ser gerados da forma mais rápida possível, para que não impacte negativamente no processo como um todo.
3. Eficiência em memória: uma vez que muitas vezes lidamos com imagens grandes, o método de geração de superpixels deve usar pouca memória. Ele deve ainda ser extensíveis a imagens tridimensionais.
4. Flexibilidade na escolha do número de superpixels gerados: idealmente os métodos devem permitir a escolha do número desejado de superpixels, a fim de, por exemplo, evitar uma super segmentação de uma imagem com poucos objetos.
5. Compacidade: Superpixels regulares e compactos são muita vezes desejáveis.
6. Redução da redundância: os superpixels devem ser perceptualmente significativos, ou seja, devem ser uniformes em cor e espacialmente, reduzindo assim a redundância dos pixels que os compõem.

3

Trabalhos Relacionados

Neste capítulo serão apresentados os trabalhos relacionados ao método proposto neste trabalho.

Em (Achanta *et al*, 2010) (2) é apresentado um algoritmo eficiente para a geração de superpixels, o *SLIC* - *Simple linear iterative clustering*. Este algoritmo utiliza uma estratégia baseada no K-means, porém com um espaço de busca limitado ao tamanho desejado do superpixel, o que torna a complexidade do algoritmo linear ao número de pixels.

Em (Achanta *et al*, 2012) (3), é feita uma comparação do SLIC com outros métodos existentes na literatura. Os superpixels gerados pelo SLIC mostram-se mais compactos e com melhor aderência a borda comparados aos outros métodos. Outra vantagem do SLIC é permitir ao usuário que escolha o número de superpixels desejados na imagem segmentada, dado através do parâmetro k . O usuário também é capaz de o nível de compactação dos superpixels gerados, através do parâmetro m , explicado mais adiante.

A Figura 3.1 apresenta um fluxograma com uma visão geral do algoritmo SLIC.

A primeira etapa do SLIC é a inicialização, onde os k clusters iniciais são criados de forma a ocupar espaços amostrados regularmente. Os clusters tem tamanhos iguais,

$$S = \sqrt{\frac{N}{k}} \quad (3-1)$$

onde N é o número total de pixels da imagem e S o tamanho de cada cluster, ou seja, o tamanho em pixels de cada superpixel. A seguir, os centros dos clusters são movidos para a posição de menor gradiente na vizinhança 3×3 para diminuir a chance de posicionar um centro sobre uma aresta ou ruído.

Na segunda etapa, intitulada rotulagem, cada pixel da imagem receberá um rótulo indicando a qual cluster pertence. Esta rotulagem é feita com base na distancia do *pixel* em questão com o cluster mais próximo dentro de uma região limitada $2S \times 2S$. Este procedimento é similar ao *k-means*, em que cada *pixel* deve ser comparado com todos os clusters, porém a limitação da região de busca faz com que o SLIC seja mais eficiente quanto ao tempo de execução. A complexidade do SLIC é linear em relação ao número de pixels da imagem $O(N)$ devido a busca ser feita apenas em um região limitada, enquanto o *k-means* é

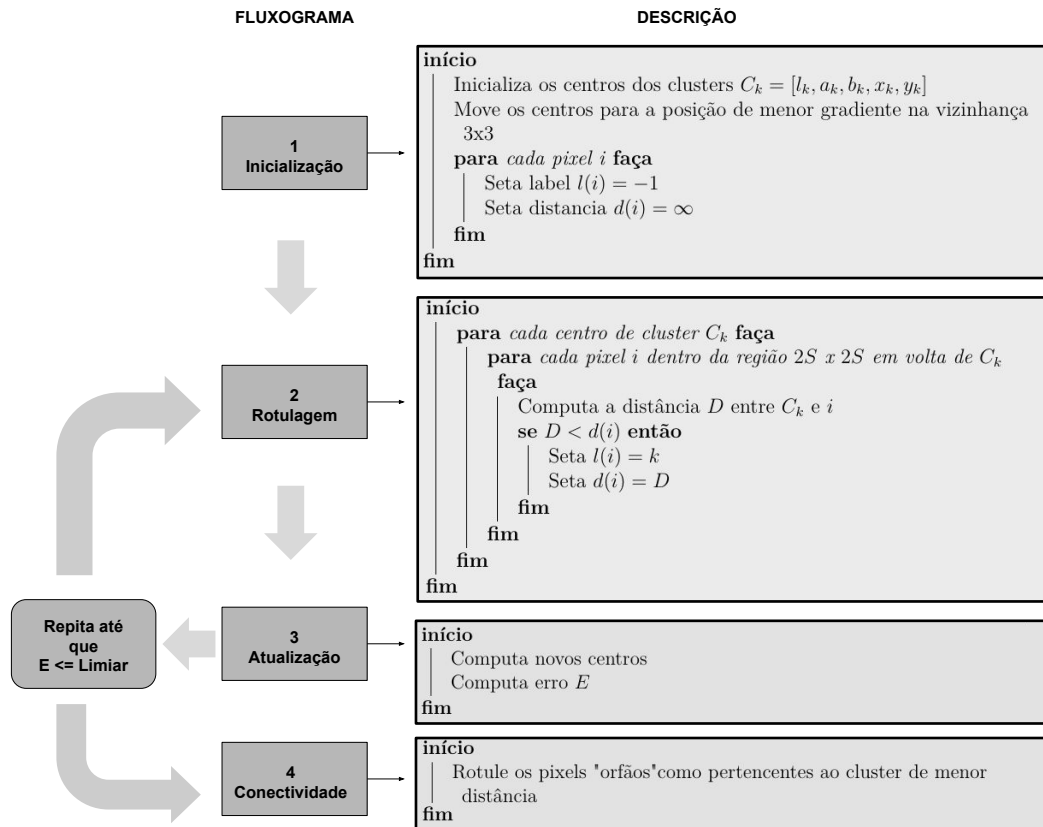
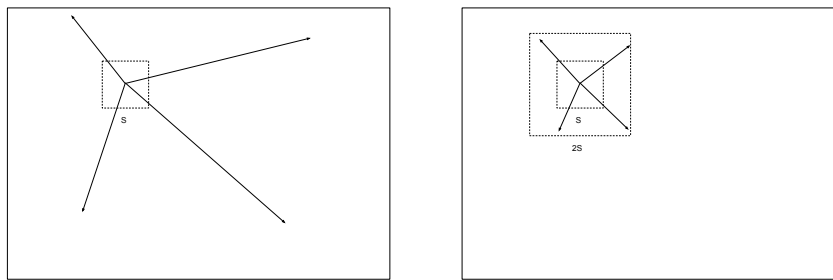


Figura 3.1: Fluxograma do SLIC. Fonte: Adaptado de *da Rosa*(4).

$O(kNI)$, onde I corresponde ao número de iterações. A Figura 3.2 ilustra os espaços de busca para o k -means e SLIC, respectivamente.



(a) Espaço de busca no k-means.

(b) Espaço de busca no SLIC.

Figura 3.2: Diferença do espaço de busca do k-means e do SLIC. Fonte: Adaptado de *Achanta et al, 2012* (3).

A medida de distância D utilizada para definir a distância do centro de um cluster a um pixel leva em consideração as 5 dimensões de um pixel: $labxy$, onde l, a, b são as componentes de cor do pixel representada no espaço $CIELAB$ e x, y são as componentes espaciais do pixel. Para combinar as distâncias de cor

d_c e espacial d_s em uma única medida, é necessário que elas sejam normalizadas. Sendo assim, a distância utilizada é dada por:

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \quad (3-2)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (3-3)$$

$$D = \sqrt{(d_c/N_c)^2 + (d_s/N_s)^2} \quad (3-4)$$

Uma vez que todos os clusters devem assumir valores máximos de tamanho S , podemos assumir que $N_s = S$. Para a definição de N_c , que corresponde a maior variação de intensidade em cada cluster, é usado o parâmetro que pode ser definido pelo usuário m . Este parâmetro corresponde ao nível de compactação que cada superpixel terá. Aplicando estes valores na equação 3-4, temos:

$$D = \sqrt{(d_c)^2 + (f \cdot d_s)^2} \quad (3-5)$$

$$f = (m/S)^2 \quad (3-6)$$

Para valores mais altos de m , os superpixels gerados serão mais compactos, uma vez que a distância espacial passa a ser destacada, enquanto para valores menores, os superpixels gerados terão maior aderência às bordas, uma vez que a distância de cor passa ser destacada.

Na terceira fase, a de atualização, os novos centros de cada cluster são atualizados como a média de cada uma de suas componentes (l, a, b, x e y) e o erro E entre os novos centros de clusters e os anteriores é calculado. As etapas 2 e 3 (rotulagem e atualização) são repetidas até que este erro calculado seja menor que um limiar estipulado, porém, (Achantá et al, 2010) estipulou que dez iterações são ideais para resultados satisfatórios.

Ao final das dez iterações, alguns pixels ainda podem estar sem rótulo. A etapa final do algoritmo, a de conectividade, é responsável por rotular estes pixels "órfãos" com o rótulo do cluster mais próximo.

Existe uma versão do SLIC em que o parâmetro m é calculado de forma automática e adaptativa, e que produz superpixels mais regulares, independente se a imagem tem regiões mais suaves e outras mais texturizadas, denominada SLICO. Esta melhoria vem com quase nenhum compromisso sobre a eficiência computacional. É também possível utilizar a versão para imagens tridimensionais que gera supervoxels. A implementação destas três versões está disponível¹ online.

¹Executável e código fonte podem ser encontrados em: <https://ivrl.epfl.ch/research/superpixels>

As Figuras 3.3 mostram a diferença no resultado da aplicação do SLIC e SLICO em três imagens. Nas três primeiras imagens, o SLIC foi aplicado, gerando superpixels menos regulares. Nas três últimas imagens, o SLICO foi aplicado, gerando superpixels mais regulares, porém com menos aderência às bordas.

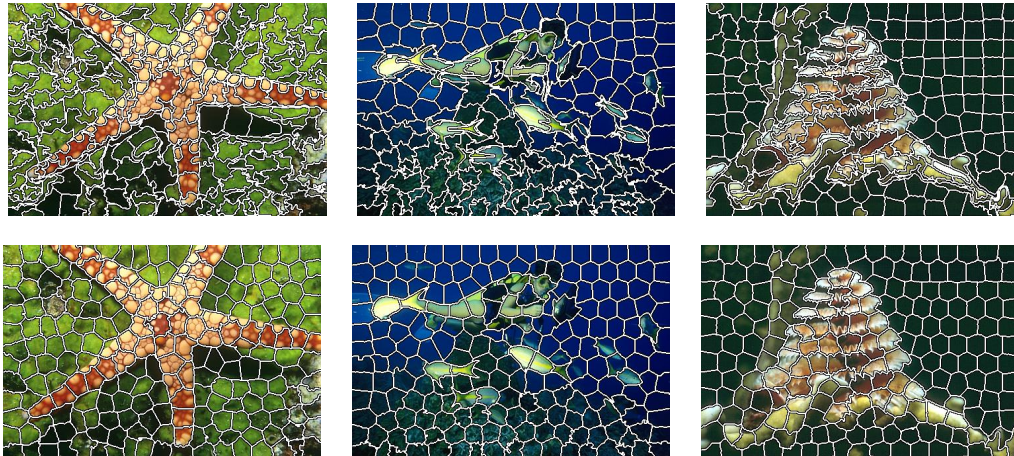


Figura 3.3: Diferença do resultado da aplicação do SLIC(imagens de cima) e SLICO(imagens de baixo).

Em (1) é apresentado, em resumo, um método intitulado "*Image Segmentation Using Superpixel Based Split and Merge Method*" que discute a ideia de utilizar *merge* e *split* na geração de superpixels a fim de obter uma boa segmentação de imagens fotográficas, que separa o fundo dos objetos em primeiro plano. Para isso, o autor diz fazer uso de uma *quad-tree*, porém é pouco específico e não fornece detalhes da implementação. O método porém, não obtém bons resultados.

Em *Image Segmentation Using Hierarchical Merge Tree* (7), *Liu et al* apresentam um método que aborda também segmentação de imagens, bem como este trabalho. O autor apresenta no artigo bons resultados para suas imagens de teste, porém, além do método ser complexo, o autor não disponibiliza uma implementação do mesmo.

4 Metodologia Proposta

Este capítulo descreve a metodologia de segmentação proposta, cujas etapas estão apresentadas no fluxograma da Figura 4.1.

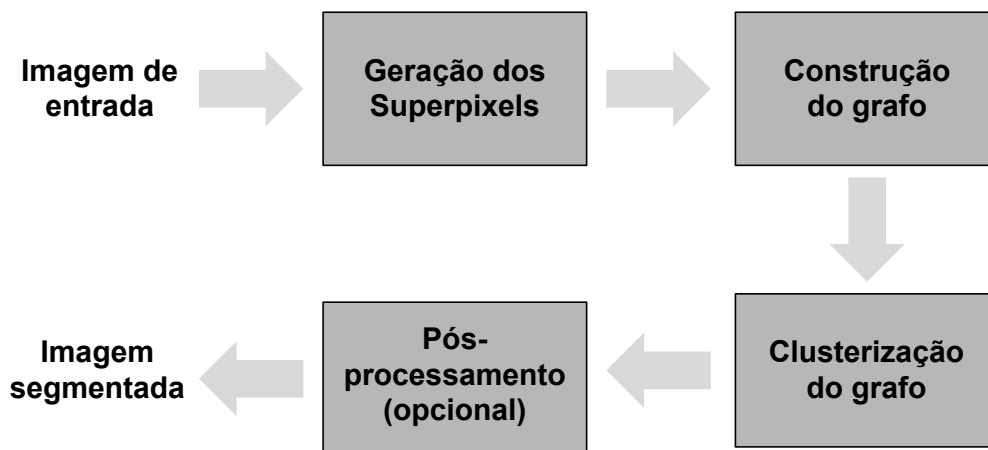


Figura 4.1: Fluxograma da metodologia proposta.

Como entrada, espera-se uma imagem, que pode ser uma imagem fotográfica bidimensional colorida, ou uma imagem de ultrassom tridimensional em tons de cinza. Dessa imagem serão extraídos superpixels. A seção 4.1 explica como os superpixels são extraídos. A seguir, um grafo é construído, como detalhado na seção 4.1 e depois é clusterizado, utilizando um processo de *merge* dos nós almejando que restem apenas nós que correspondem a objetos na imagem. O processo de clusterização do grafo é explicado na seção 4.3. Por fim, a seção 4.4 explica o pós-processamento que pode ou não ser utilizado. Diferentemente de outros métodos, este método não necessita que o usuário determine a priori a quantidade de objetos presentes na imagem. Porém, no pós-processamento, que é opcional, essa informação é utilizada para melhorar os resultados da segmentação.

Para exemplificar o método, a imagem da Figura 4 será utilizada.



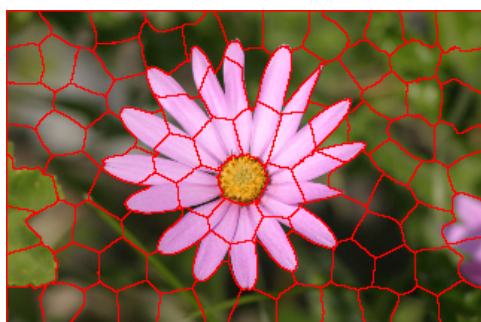
4.1 Geração dos Superpixels/Supervoxels

A primeira etapa do método é de inicialização dos clusters como superpixels. Os superpixels são gerados, usando a implementação disponibilizada por (2), para o caso de imagens fotográficas e do SLIC supervoxel para imagens tridimensionais, que é o caso das imagens de ultrassom, e formam uma clusterização inicial. Esta técnica foi escolhida dentre as demais disponíveis por ser de rápida execução, ter bons resultados, e ser expansível a três dimensões.

Esta estratégia é adotada para aproveitar dos benefícios dos superpixels. Com isso, o tamanho da entrada é reduzido significativamente, uma vez que sem a utilização deles a entrada seria todos os pixels, e com o uso deles a entrada passa a ser regiões significativas, já separadas pelas bordas dos objetos da imagem, e com características semelhantes, o que facilita a segmentação.

A escolha do número de superpixels gerados varia com o tamanho e com nível de detalhe da imagem usada. Em imagens com mais detalhes, pode-se perder algumas bordas e detalhes ao gerar poucos superpixels, enquanto em imagens com poucos detalhes, pode-se destacar ruídos gerando muitos superpixels.

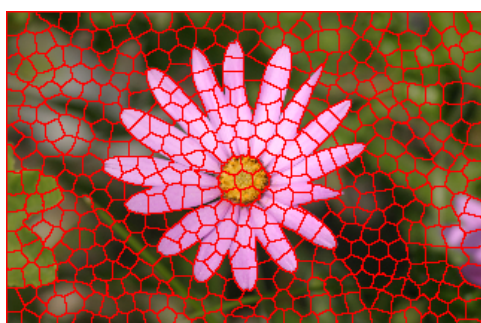
As imagens da Figura 4.2 a seguir ilustram a aplicação deste primeiro passo com diferentes quantidades de superpixels:



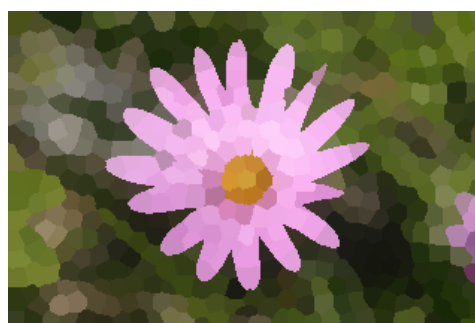
(a) 100 superpixels.



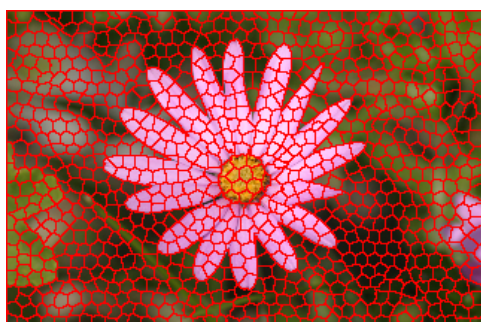
(b) 100 superpixels coloridos com sua cor média.



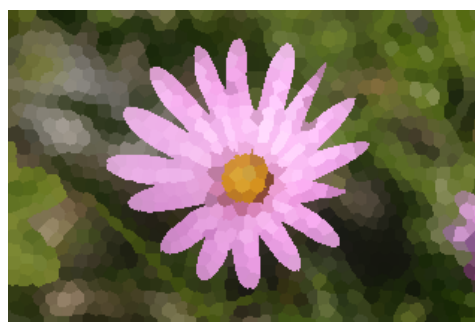
(c) 500 superpixels.



(d) 500 superpixels coloridos com suas cores médias.



(e) 1000 superpixels.



(f) 1000 superpixels coloridos com suas cores médias.

Figura 4.2: Exemplo da geração diferentes números de superpixels.

4.2

Construção do Grafo

Após a geração dos superpixels, no caso das imagens bidimensionais, ou dos supervoxels, no caso das imagens tridimensionais, um grafo é construído de maneira que cada nó do grafo representa um superpixel ou supervoxel. Para isso, cada nó do grafo deve ter informação da cor média dos pixels que pertencem ao superpixel correspondente e quais são esses pixels.

Cada nó do grafo tem arestas para todos os superpixels vizinhos. Um superpixel é considerado vizinho de todos os superpixels que tem pelo menos

um pixel vizinho a um pixel dele.

O algoritmo 1 detalha esta etapa de construção do grafo.

Algoritmo 1: CONSTROI GRAFO

```

1 inicio
2   para cada superpixel  $s$  gerado hacer
3     Cria um novo nó  $n$ 
4     para cada pixel  $p$  pertencente a  $s$  hacer
5       Adicione  $p$  a lista de pixels de  $n$ 
6     fin
7     Calcula a média de cor dos pixels e armazena em  $n$ 
8     Adiciona  $n$  no grafo
9   fin
10  para cada pixel  $p1$  da imagem hacer
11    para cada pixel  $p2$  vizinho de  $p1$  hacer
12      se  $p1$  e  $p2$  pertencem a nós diferentes então
13        Adiciona o nó de  $p2$  a lista de vizinhos do nó de  $p1$ 
14      fin
15    fin
16  fin
17 fin

```

Um pixel p de coordenadas (x, y) , considerando uma *vizinhança-8*, têm dois vizinhos horizontais, dois verticais, cujas coordenadas são dadas respectivamente por:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

e quatro vizinhos diagonais cujas coordenadas são dadas por:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1).$$

Um voxel de coordenadas (x, y, z) , por sua vez, contém uma vizinhança composta por 27 voxels. A figura 4.3 ilustra, em uma imagem bidimensional (a) a vizinhança de um pixel e em uma tridimensional a vizinhança de um voxel (b).

Quando os superpixels gerados são regulares, eles tendem a ter no máximo oito vizinhos cada (exceto os de borda, que tem menos), porém quanto menos regulares forem, mais vizinhos podem ter. O mesmo ocorre com os supervoxels, que se regulares, tendem a ter no máximo 27 vizinhos, e quanto menos regulares, a tendência é que tenham ainda mais vizinhos.

A Figura 4.4 ilustra como deve ser a saída desta etapa, onde, para simplificar a ilustração, a saída da etapa anterior é uma imagem dividida em apenas cinco superpixels.

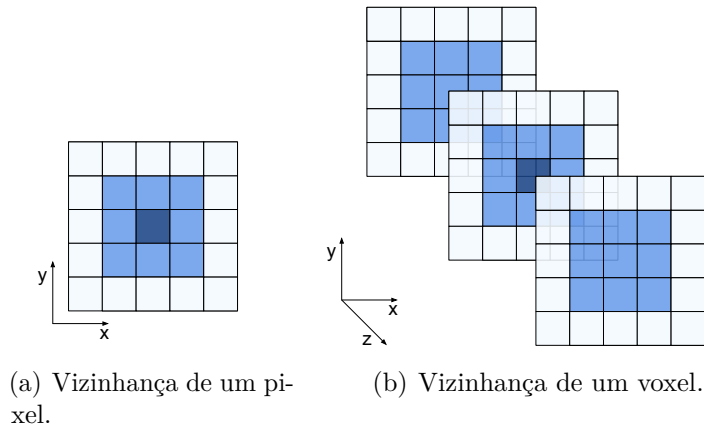


Figura 4.3: Vizinhos de um pixel e de um voxel, respectivamente.

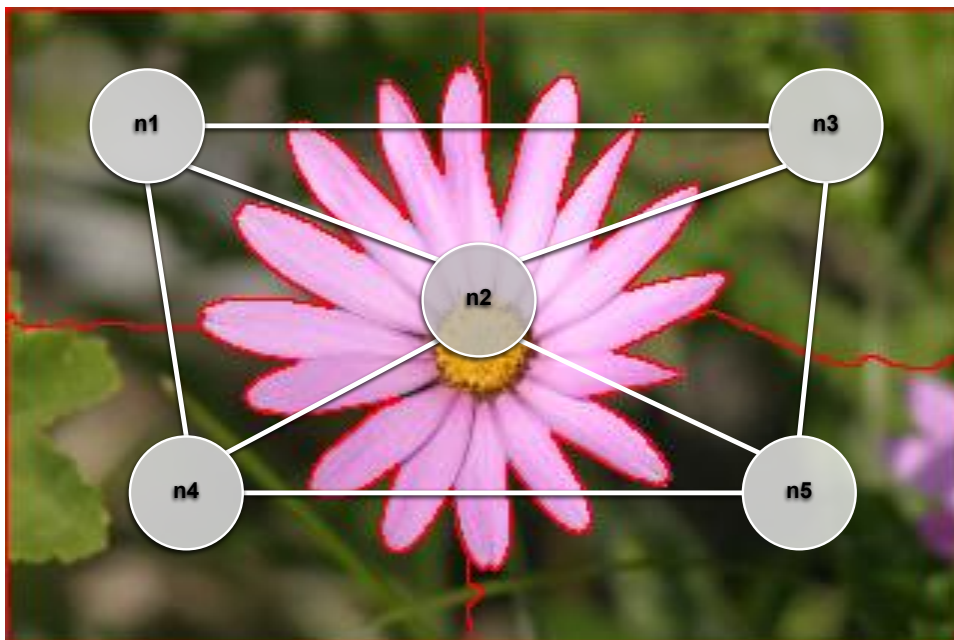


Figura 4.4: Exemplo de grafo para uma imagem com cinco superpixels.

4.3 Clusterização dos Nós do Grafo

Uma vez que o grafo de superpixel está construído, é desejado que os nós mais semelhantes do grafo sejam colapsados até que cada nó restante represente um objeto da imagem e é nesta etapa que isto ocorre.

Como mostrado no Algoritmo 2, inicialmente, as distâncias de todas as arestas são calculadas e armazenadas em um *heap* de mínimo. A forma como essa distância é calculada é discutida mais adiante. A seguir, a aresta de menor valor é removida do *heap* e tem seus nós colapsados. No procedimento *juntaNos(i, j)* de junção dos dois nós de uma aresta (Algoritmo 3), os pixels pertencentes ao nó j passam a pertencer ao nó i e a vizinhança do nó i é

atualizada com os antigos vizinhos do nó j . Após a junção, as novas arestas do nó i são calculadas e inseridas no *heap*. Este procedimento é realizado repetidamente até que a aresta com menor distância no *heap* seja maior que o erro aceito dado pelo usuário.

Algoritmo 2: CLUSTERIZAGRAFO(ERROACEITO)

```

1 inicio
2   Cria heap edgesHeap // heap de mínimo
3   para cada nó  $i$  do grafo hacer
4     para cada nó  $j$  vizinho de  $i$  hacer
5       Computa distância entre  $i$  e  $j$  :  $edge(i, j)$ 
6       Adiciona  $edge(i, j)$  a edgesHeap
7     fin
8   fin
9   para cada  $edge(i, j)$  de edgesHeap hacer
10    Remove  $edge(i, j)$  de edgesHeap
11    se  $edge(i, j) < erroAceito$  então
12      JuntaNos( $i, j$ )
13      para cada nó  $k$  vizinho do nó  $i$  hacer
14        Computa distância entre  $i$  e  $j$  :  $edge(i, k)$ 
15        Adiciona  $edge(i, k)$  a edgesHeap
16      fin
17    fin
18  fin
19 fin

```

Pode-se perceber que a cada iteração, a aresta que tem menor valor é a que tem seus nós colapsados, ou seja, o método adota uma estratégia gulosa.

Algoritmo 3: JUNTANÓS(I, J)

```

1 inicio
2   Adiciona ao nó  $i$  os pixels pertencentes ao nó  $j$ 
3   para cada nó  $k$  vizinho de  $j$  hacer
4     Remove o nó  $j$  da listas de vizinhos do nó  $k$ 
5     Adiciona o nó  $k$  na lista de vizinhos do nó  $i$ 
6     Adiciona o nó  $i$  na lista de vizinhos do nó  $k$ 
7   fin
8   Remove o nó  $j$  do grafo
9 fin

```

4.3.1

Métricas de distância

As métricas de distância utilizadas, consideram apenas a distância de cor dos clusters (não consideram a distância de posição), porém a utilização do grafo garante que apenas clusters vizinhos são colapsados.

Estas métricas são calculadas de formas diferentes para cada tipo de imagem. Para as imagens fotográficas, que são coloridas, a distância ΔE^* de dois nós n_1 e n_2 de uma aresta é dada pela diferença da cor média destes nós. Podemos considerar que a cor tem uma componente de luminância L e um de cromaticidade C , composta pelas componentes a e b , assim temos a diferença de cor:

$$\Delta E^* = \sqrt{\Delta L^{*2} + \Delta C^{*2}} \quad (4-1)$$

onde

$$\Delta L^{*2} = (L_{n_1}^* - L_{n_2}^*)^2 \quad (4-2)$$

e

$$\Delta C^{*2} = (a_{n_1}^* - a_{n_2}^*)^2 + (b_{n_1}^* - b_{n_2}^*)^2 \quad (4-3)$$

Geometricamente, se delimitarmos $\Delta E^* \leq 1$, por exemplo, temos que os valores de ΔE^* se encontram dentro de um círculo de raio 1 centrado em zero, como mostrado na Figura 4.5.

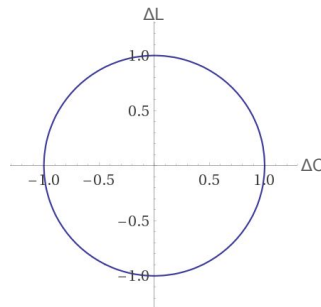


Figura 4.5: $\Delta E^* \leq 1$

Verificou-se porém, que um mesmo objeto, com a mesma cor em toda sua extensão, pode aparecer na imagem com variação em seu tom devido a iluminação. A fim de minimizar a influência da iluminação, a componente L é dividida por um fator α , e com isso temos:

$$\Delta L^{*'} = \frac{\Delta L^*}{\alpha} \quad (4-4)$$

$$\Delta E^{*'} = \sqrt{\frac{\Delta L^{*2}}{\alpha^2} + \Delta C^{*2}} \quad (4-5)$$

E assim, geometricamente, ao limitarmos $\Delta E^{*'} \leq 1$, com $\alpha^2 = 10$, temos que todos os valores de $\Delta E^{*'}$ agora estão dentro de uma elipse como mostrado na Figura 4.6.

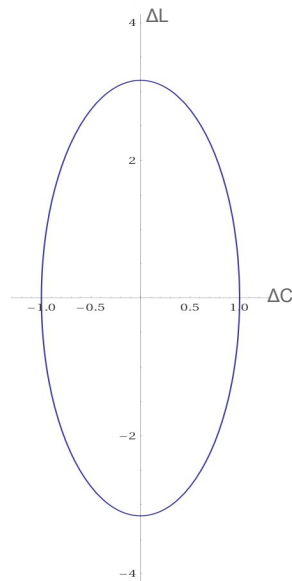


Figura 4.6: $\Delta E^{*'} \leq 1$ com $\alpha^2 = 10$

Já nas imagens de ultrassom, que são imagens em tons de cinza, a distância ΔE^* é calculada usando apenas a componente de luminosidade L^* de cada cor, uma vez que as componentes a^* e b^* são sempre zero. Assim, para esse tipo de imagem temos:

$$\Delta E^{*''} = \sqrt{(L_{n_1}^* - L_{n_2}^*)^2} \quad (4-6)$$

Ao final desta etapa, espera-se que cada cluster restante represente um objeto da cena, porém, algumas vezes, um cluster ainda representa uma parte de um objeto da cena. Nestes casos, o pós processamento pode ser útil.

Em muitos casos, o número de objetos da cena é subjetivo, sendo dependente do nível de detalhamento que o usuário deseja para a segmentação. A Figura 4.7 ilustra duas possíveis segmentações para a Figura 4, onde na primeira tem um nível menor de detalhes, onde apenas a flor é segmentada do fundo, já na segunda, o nível de detalhamento é melhor, e podemos observar que o miolo da flor e um pedaço de flor que aparece a direita também são segmentados.

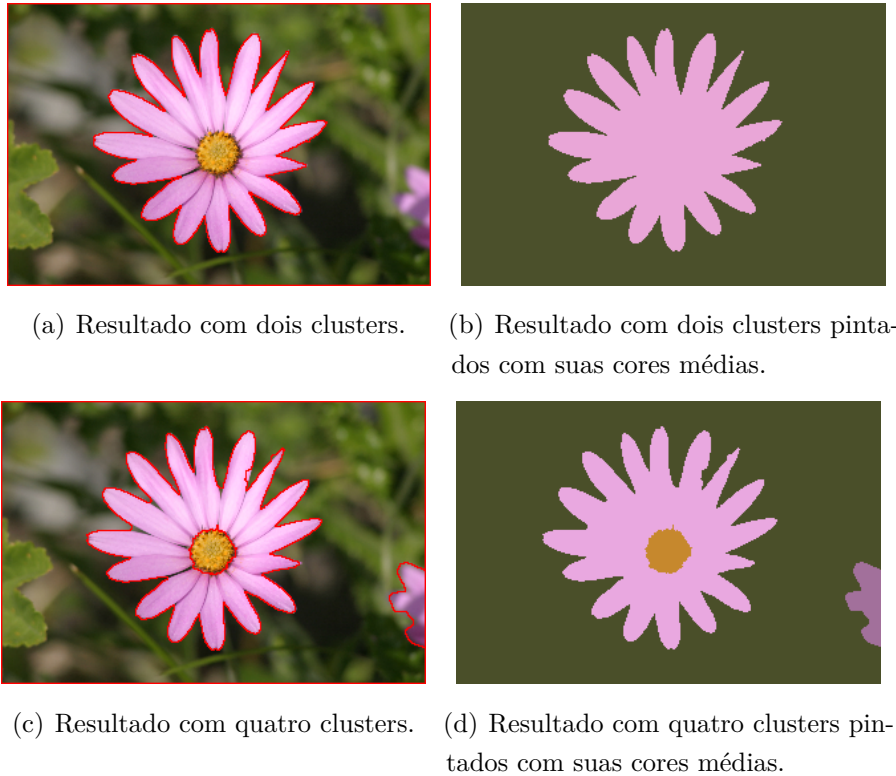


Figura 4.7: Duas possíveis segmentações para a Figura 4.

4.4 Pós-processamento

Nesta etapa, que não é obrigatória, a segmentação é continuada baseando-se na quantidade de objetos presentes na imagem.

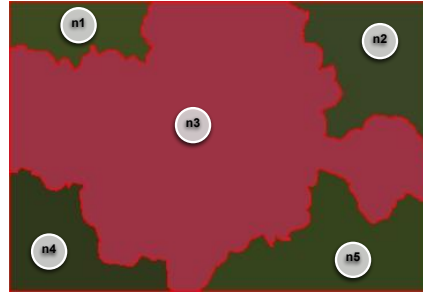
No caso da imagem de ultrassom, que é conhecido que tem-se dois objetos, sendo eles o feto ligado a placenta e o líquido amniótico, os nós são separados em dois grupos, os mais claros e os mais escuros, segundo o seus valores de luminância. Os nós pertencentes a cada um dos grupos são colapsados, mesmo que não existam arestas entre eles, restando assim apenas dois nós.

No caso de imagens fotográficas, o pós-processamento, que é opcional, deve receber do usuário o número de objetos na imagem. Para esta etapa, são criadas arestas de todos os nós para todos os nós do grafo. As arestas de menor valor vão tendo seus nós colapsados até que restem apenas a quantidade de nós equivalente ao número de objetos fornecido pelo usuário. Este procedimento pode ser interessante no caso de parte de um objeto estar sendo sobreposto por outro, fazendo com que ele apareça dividido. Na Figura 4.4 mostra-se um caso que este pós-processamento é útil. Considerando que o desejado neste caso é separar o que é flor do fundo, ao final da terceira etapa do algoritmo, a imagem é dividida em cinco nós, sendo que o fundo aparece dividido em

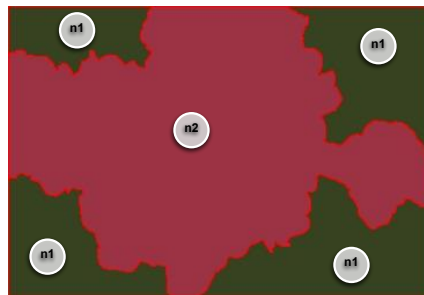
quatro nós que não são vizinhos entre si, porém tem suas cores médias muito próximas. Com o pós-processamento, estes quatro nós que representam o fundo são colapsados, restando no grafo apenas dois nós, um que representa as flores e um que representa o fundo.



(a) Imagem original.



(b) Resultado da terceira etapa com os 5 clusters enumerados e pintados com suas cores médias.



(c) Resultado do pós-processamento com apenas 2 clusters enumerados e pintados com suas cores médias.

Figura 4.8: Exemplo de imagem em que o pós-processamento pode ser interessante.

5 Resultados

Neste capítulo são apresentados e discutidos os resultados obtidos com a utilização da metodologia descrita no Capítulo 4.

Toda a implementação foi feita utilizando a linguagem de programação *c++*.

Para realização dos testes em imagens fotográficas foram utilizadas imagens das bases MSRC¹ e *The Berkeley Segmentation Dataset and Benchmark*² (8).

Nos testes com imagens fotográficas, foi utilizado $\alpha^2 = 10$ na equação 4-5 referente a métrica de distância de dois nós, ficando a distância entre dois nós como:

$$\Delta E^{*'} = \sqrt{\frac{\Delta L^{*2}}{10} + \Delta C^{*2}} \quad (5-1)$$

5.1 Dataset MSRC

Nesta seção mostraremos e discutiremos a aplicação do método em quatro imagens na base *MSRC*. Esta base é disponibilizada pela *Microsoft Research em Cambridge* e é composta por 591 imagens e 23 classes de objetos. É um dataset comumente usado para segmentação e reconhecimento de objetos, usando aprendizado de máquina, e por isto está dividido em um conjunto de imagens de treinamento e um conjunto de imagens para teste. Para cada imagem da base, é disponibilizado um *ground truth* com os objetos pertencentes a imagem marcados. Para estes testes, a geração dos superpixels foi feita usando o SLICO, pois foi a que mostrou melhores resultados.

5.1.1 Imagem do cachorro na grama

Na primeira imagem de teste, Figura 5.1, tem uma fotografia de um cachorro em um fundo de grama e a Figura 5.2 contém o *ground truth* fornecido pelo *dataset* para esta imagem. Como o dataset é feito para segmentação e

¹<https://www.microsoft.com/en-us/research/project/image-understanding/>

²<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

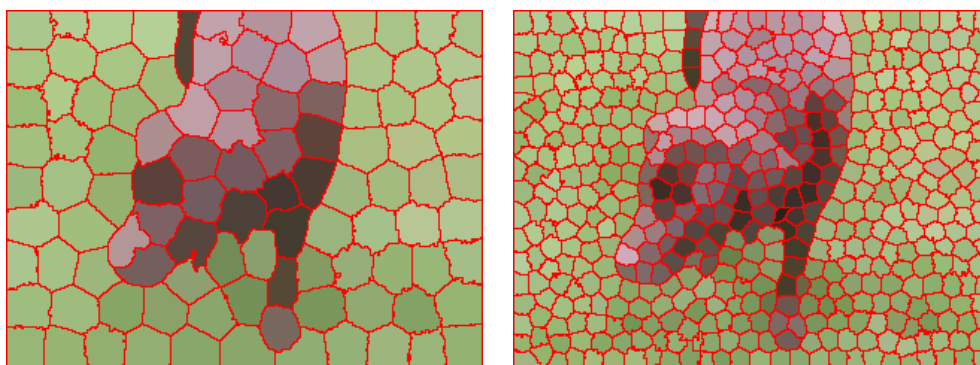
reconhecimento de objetos, o *ground truth* tem a marcação do objeto "cachorro" e do objeto "grama".



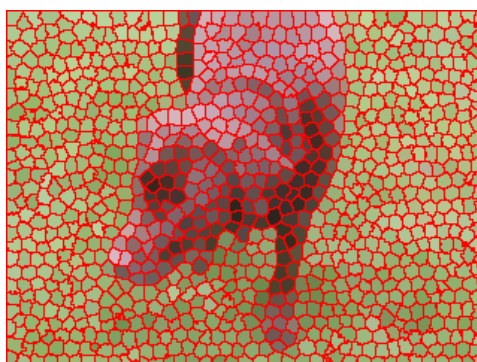
Figura 5.1: Imagem do cachorro na grama.

Figura 5.2: *Ground truth* fornecido pelo dataset para imagem do cachorro.

A primeira etapa do método, explicada na Seção 4.1 é a geração dos superpixels. A Figura 5.3 ilustra a geração de diferentes quantidades de superpixels para esta imagem.



(a) Imagem com 100 superpixels coloridos com suas cores médias. (b) Imagem com 500 superpixels coloridos com suas cores médias.



(c) Imagem com 1000 superpixels coloridos com suas cores médias.

Figura 5.3: Geração de 100, 500 e 1000 superpixels, respectivamente, para a imagem do cachorro.

A Tabela 5.1 contém os resultados da aplicação do método, sem o pós processamento, para diferentes números de superpixels e erros aceitos. Como podemos perceber, quanto maior o erro aceito, menos clusters no final e quanto maior o número de superpixels no início, mais detalhada a clusterização.

Como esta imagem não possui muitos detalhes que devem ser destacados após a clusterização, a geração de 100 superpixels é suficiente para este caso. Pode-se observar que o melhor resultado foi obtido com 100 superpixels e erro 4. Neste resultado foram gerados dois clusters, um de fundo e um com o cachorro, com as bordas mais suaves do que o resultado com 1000 superpixels e erro 4.

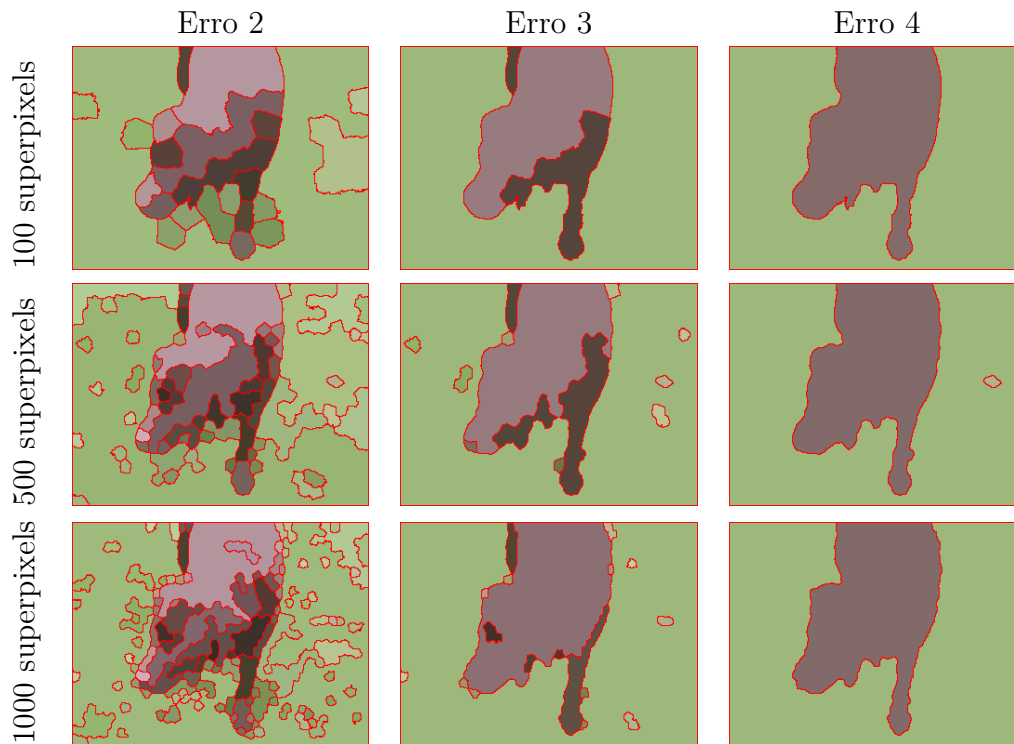


Tabela 5.1: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do cachorro.

5.1.2

Imagem de duas vacas no pasto

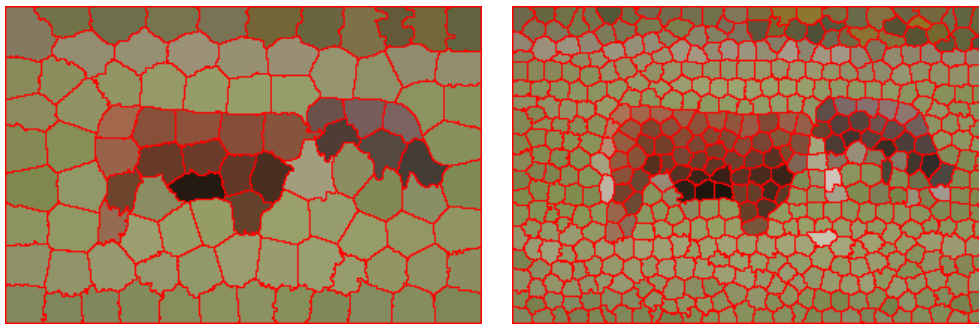
Na segunda imagem de teste (Figura 5.4), existem duas vacas em um fundo de grama e a Figura 5.5 contém o *ground truth* fornecido pelo *dataset* para esta imagem, onde o objeto "vaca" e "grama" são marcados.



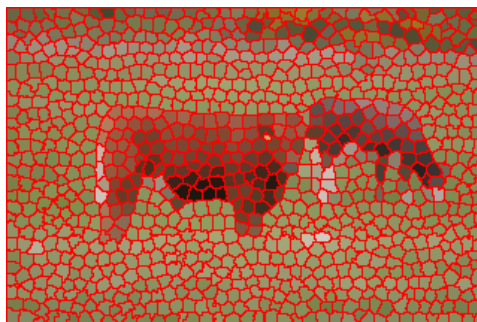
Figura 5.4: Imagem das duas vacas no pasto. Figura 5.5: *Ground truth* fornecido pelo dataset para imagem das vacas.

A primeira etapa do método, explicada na Seção 4.1 é a geração dos superpixels. A Figura 5.6 ilustra a geração de diferentes quantidades de

superpixels para esta imagem.



(a) Imagem com 100 superpixels coloridos com suas cores médias. (b) Imagem com 500 superpixels coloridos com suas cores médias.



(c) Imagem com 1000 superpixels coloridos com suas cores médias.

Figura 5.6: Geração de 100, 500 e 1000 superpixels, respectivamente, para imagem das vacas.

A Tabela 5.2 contém os resultados da aplicação do método, sem o pós processamento, para diferentes números de superpixels e erros aceitos. Como podemos perceber, quanto maior o erro aceito, menos clusters no final e quanto maior o número de superpixels no início, mais detalhada a clusterização.

Para este caso, o melhor resultado foi obtido com 100 superpixels e erro 4. Neste resultado foram gerados quatro clusters: o fundo, as vacas, sendo uma delas dividida em dois clusters.

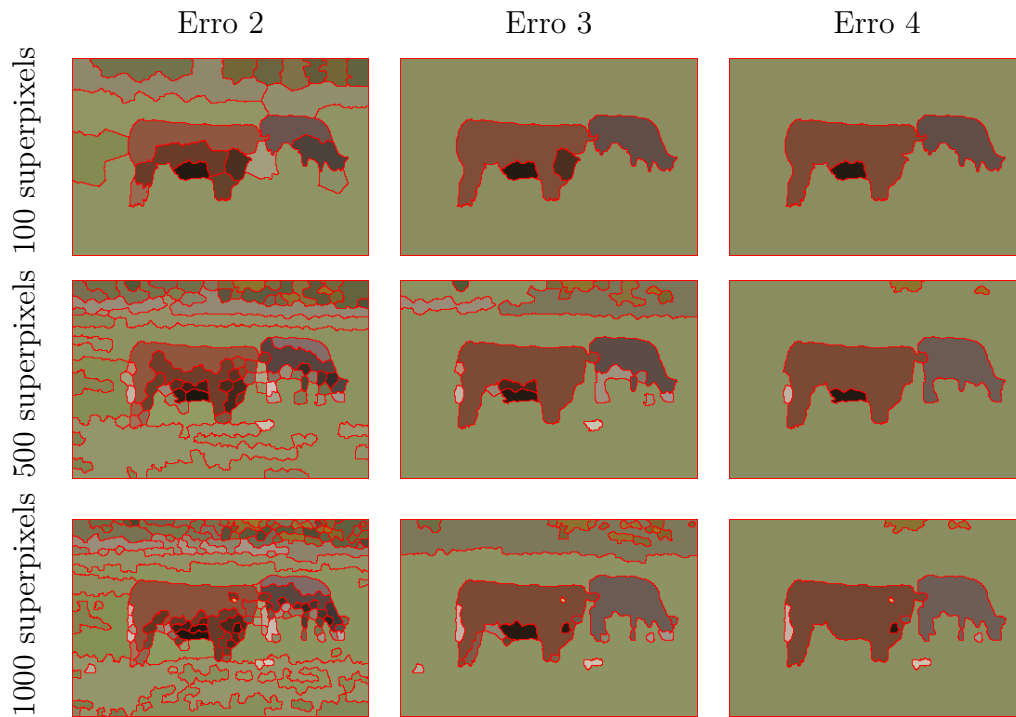


Tabela 5.2: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem das vacas.

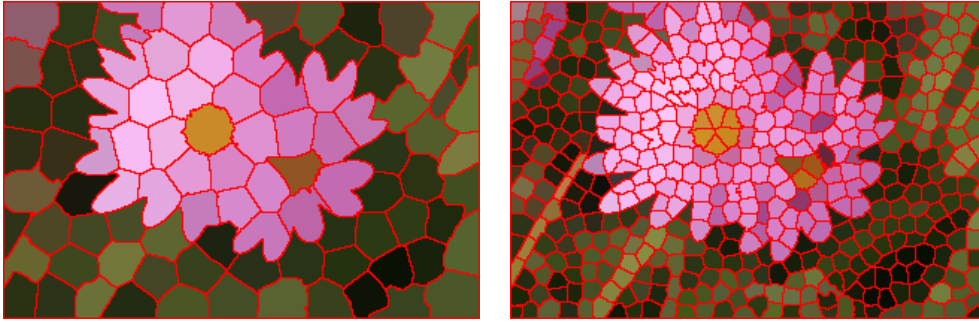
5.1.3 Imagem das flores sobrepostas

A terceira imagem de teste (Figura 5.7) possui flores com um fundo com folhas. O *ground truth* fornecido pelo dataset está ilustrado na Figura 5.8, onde apenas o objeto "flor" está marcado.

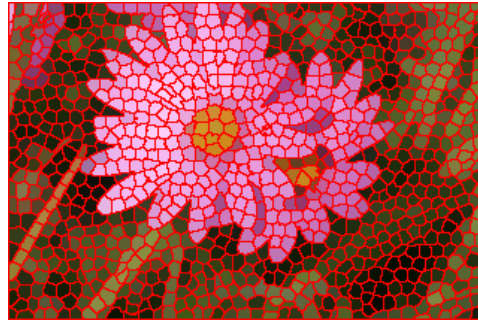


Figura 5.7: Imagem das flores sobrepostas. Figura 5.8: *Ground truth* fornecido pelo dataset para a imagem das flores.

A seguir, na Figura 5.1.3, a geração de 100, 500 e 1000 superpixels, respectivamente para a imagem de entrada.



(a) Imagem com 100 superpixels coloridos com suas cores médias. (b) Imagem com 500 superpixels coloridos com suas cores médias.



(c) Imagem com 1000 superpixels coloridos com suas cores médias.

Figura 5.9: Geração de 100, 500 e 1000 superpixels, respectivamente, para imagem das flores.

A Tabela 5.3 mostra diferentes resultados do método, também sem a necessidade do pós-processamento, para diferentes quantidades de superpixels e erros aceitos. Para esta imagem, os melhores resultados são com um erro maior que a do anterior, com valor 5. Este erro maior é necessário devido a maior diferença entre as cores da flor e do fundo.

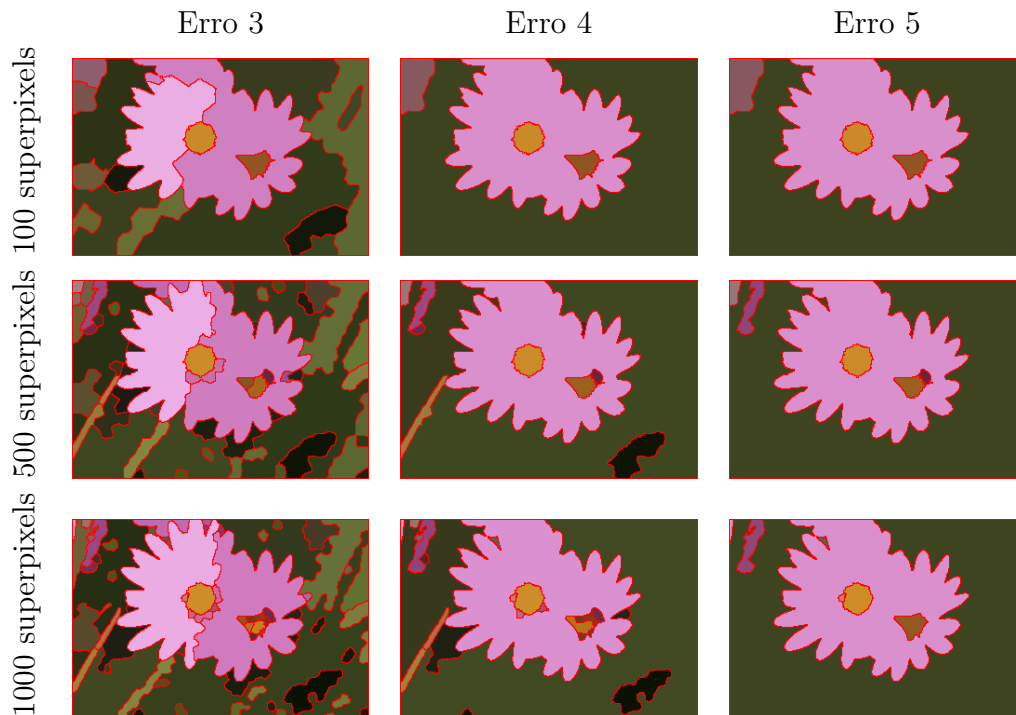


Tabela 5.3: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem das flores.

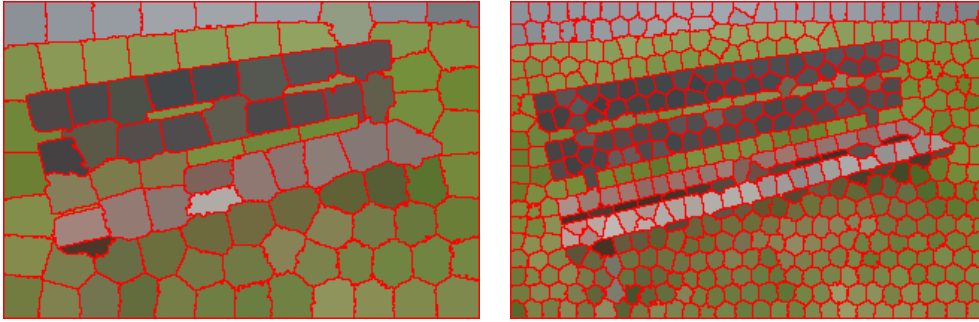
5.1.4 Imagem do banco de jardim

A quarta imagem de teste (Figura 5.10) possui um banco com um fundo de grama e acima uma faixa de asfalto. O *ground truth* fornecido pelo dataset está ilustrado na Figura 5.11, onde apenas os objetos "banco" e "grama" estão marcados.

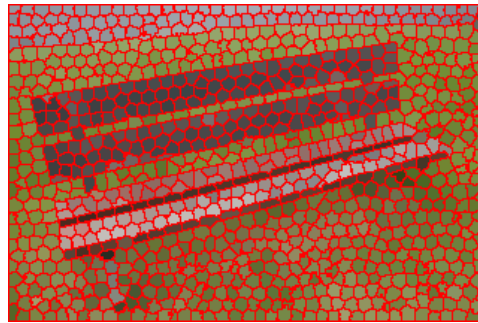


Figura 5.10: Imagem do banco de jardim. Figura 5.11: *Ground truth* fornecido pelo dataset para a imagem do banco.

A seguir, na Figura 5.12, a geração de 100, 500 e 1000 superpixels, respectivamente para a imagem de entrada.



(a) Imagem com 100 superpixels coloridos com suas cores médias. (b) Imagem com 500 superpixels coloridos com suas cores médias.



(c) Imagem com 1000 superpixels coloridos com suas cores médias.

Figura 5.12: Geração de 100, 500 e 1000 superpixels, respectivamente, para imagem do banco.

A Tabela 5.4 mostra diferentes resultados do método, sem o pós-processamento, para diferentes quantidades de superpixels e erros aceitos. Para esta imagem, os melhores resultados são com 1000 superpixels e erro 4, pois a imagem é mais detalhada e com as outras segmentações iniciais, perde-se informações.

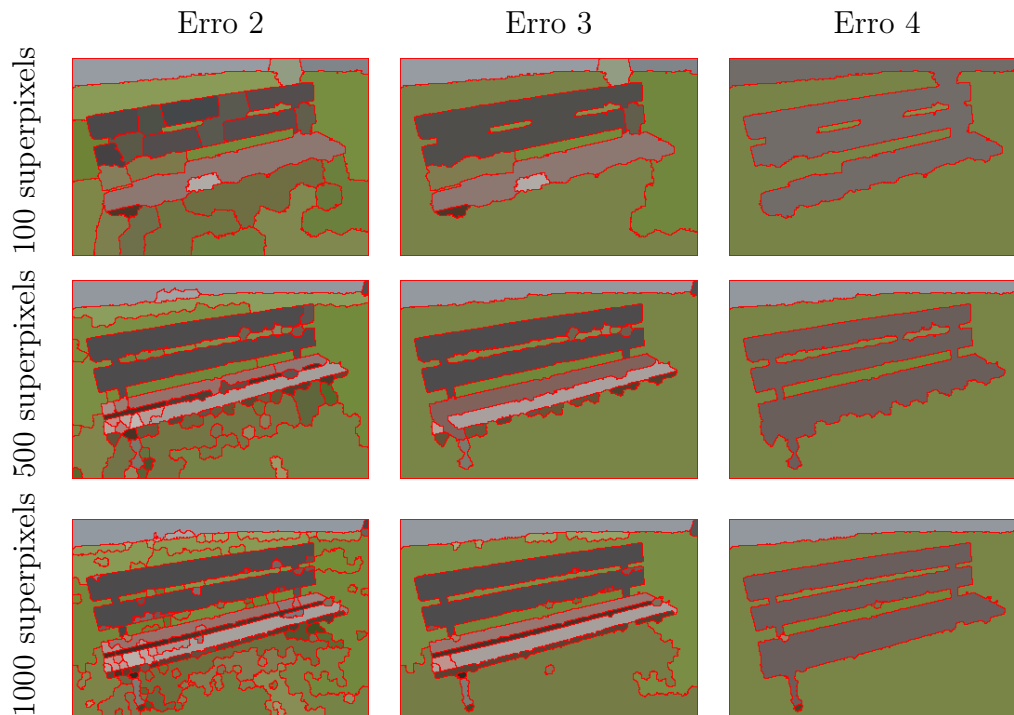


Tabela 5.4: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do banco.

Para as quatro imagens testadas e analisadas deste dataset, os resultados foram satisfatórios e a segmentação gerada foi melhor que a do *ground truth*.

5.2 Dataset Berkeley

Nesta seção primeiramente serão mostrados os resultados da aplicação do método em algumas imagens do dataset Berkeley (*The Berkeley Segmentation Dataset and Benchmark*), e depois os mesmos serão discutidos. Este dataset foi construído para pesquisas de segmentação e detecção de bordas e é composto por 1000 imagens. O *ground truth* fornecido para cada imagem são as bordas dos objetos da imagem. Este trabalho não foi desenvolvido a fim de detectar bordas, porém, para fins comparativos, uma imagem foi gerada apenas com as bordas dos clusters. Uma vez que o SLIC tem uma maior aderência as bordas, aplicamos o método utilizando como segmentação inicial também o SLIC. As imagens deste dataset, em geral, são mais complexas que as do que do MSRC, que eram basicamente do formato "objeto-fundo".

A seguir os testes são apresentados, para as algumas imagens, com diferentes números de clusters iniciais e erros aceitos, e no final são discutidos.

5.2.1 Imagem da mergulhadora com peixes

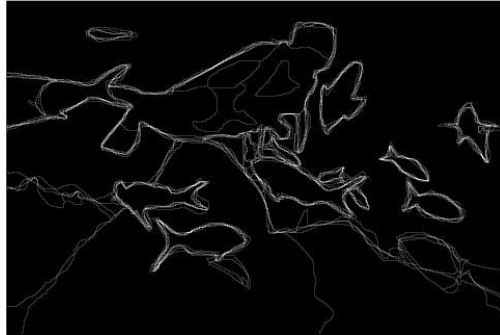


Figura 5.13: Imagem da mergulhadora. Figura 5.14: *Ground truth* fornecido pelo dataset para imagem da mergulhadora.

PUC-Rio - Certificação Digital Nº 1512331/CA

	SLIC	SLICO
100 superpixels		
500 superpixels		
1000 superpixels		

Tabela 5.5: Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem da mergulhadora.

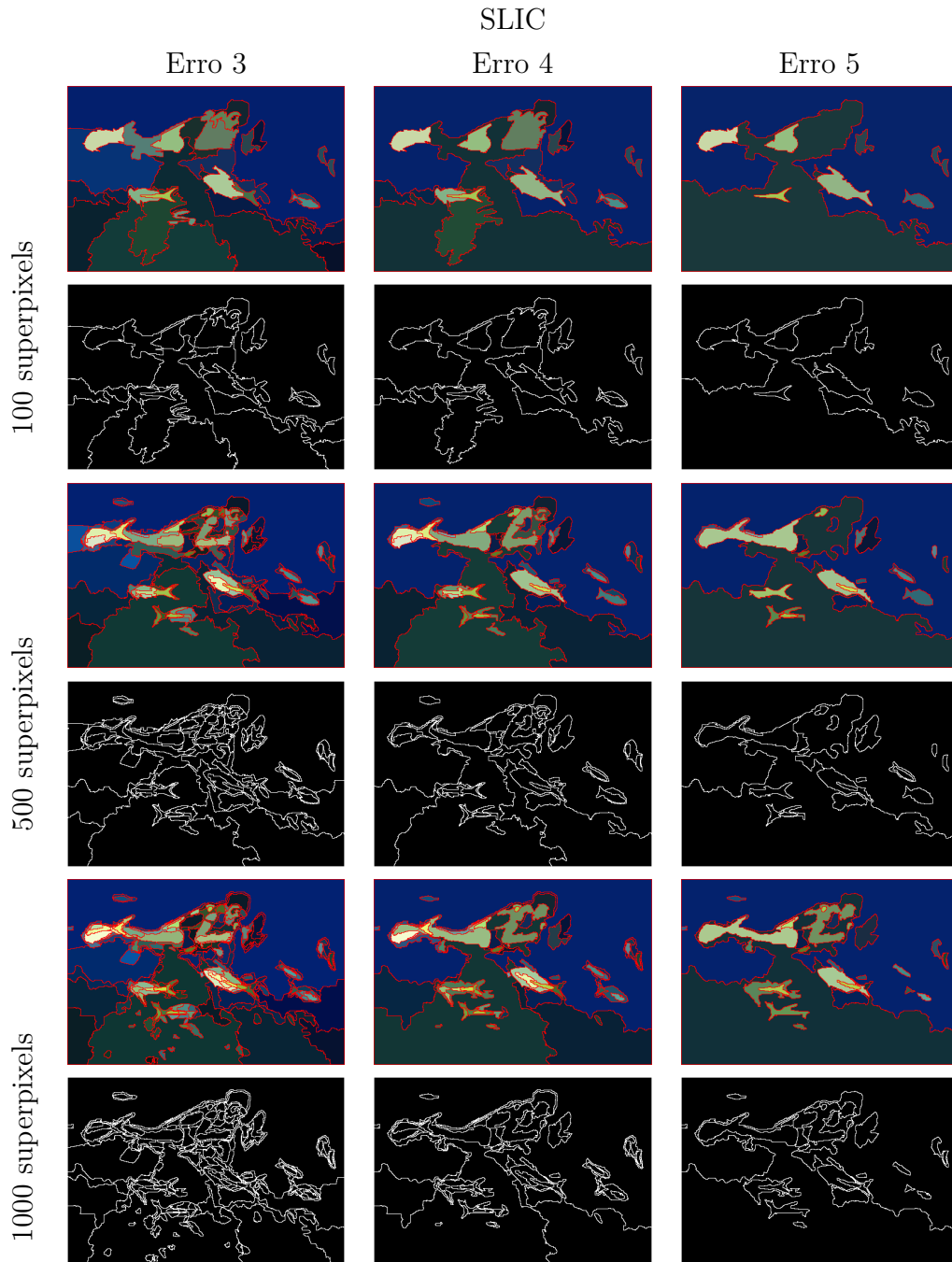


Tabela 5.6: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da mergulhadora utilizando o SLIC como segmentação inicial.

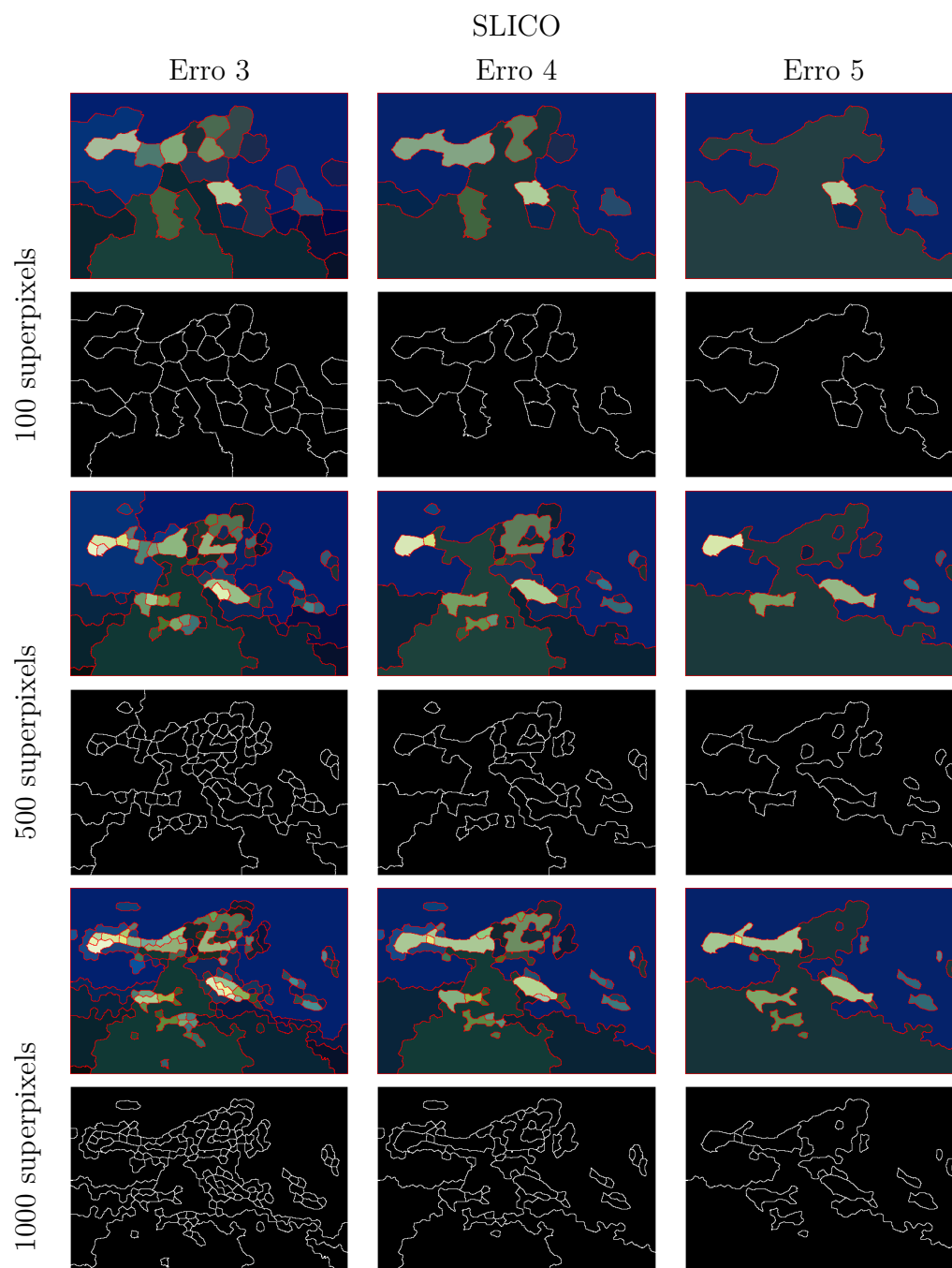


Tabela 5.7: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da mergulhadora utilizando o SLICO como segmentação inicial.

5.2.2
Imagem do campanário de igreja



Figura 5.15: Imagem do campanário de igreja.

Figura 5.16: *Ground truth* fornecido pelo dataset para imagem do campanário de igreja.

PUC-Rio - Certificação Digital Nº 1512331/CA

	SLIC	SLICO
100 superpixels		
500 superpixels		
1000 superpixels		

Tabela 5.8: Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem da igreja.

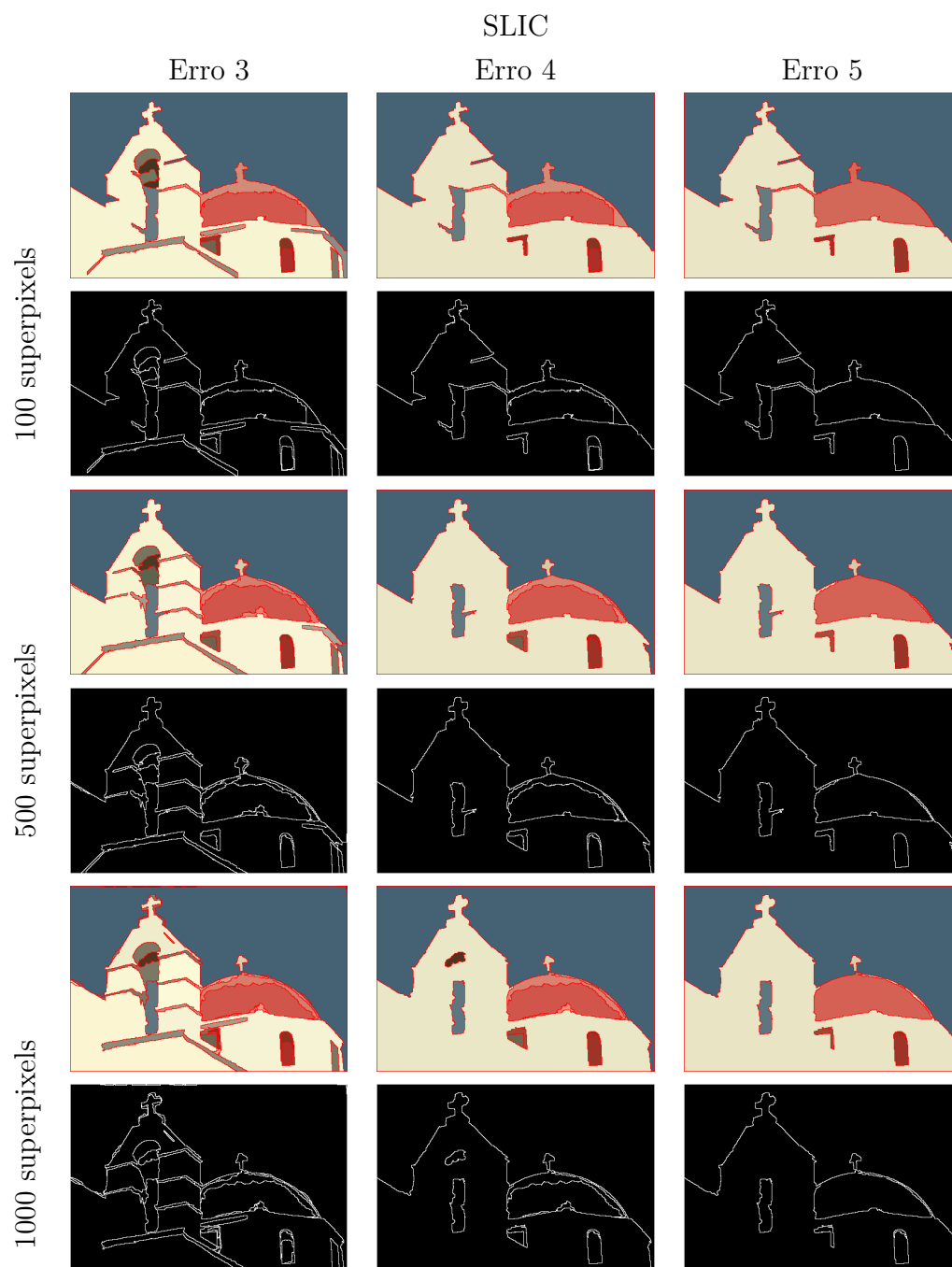


Tabela 5.9: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da igreja utilizando o SLIC como segmentação inicial.

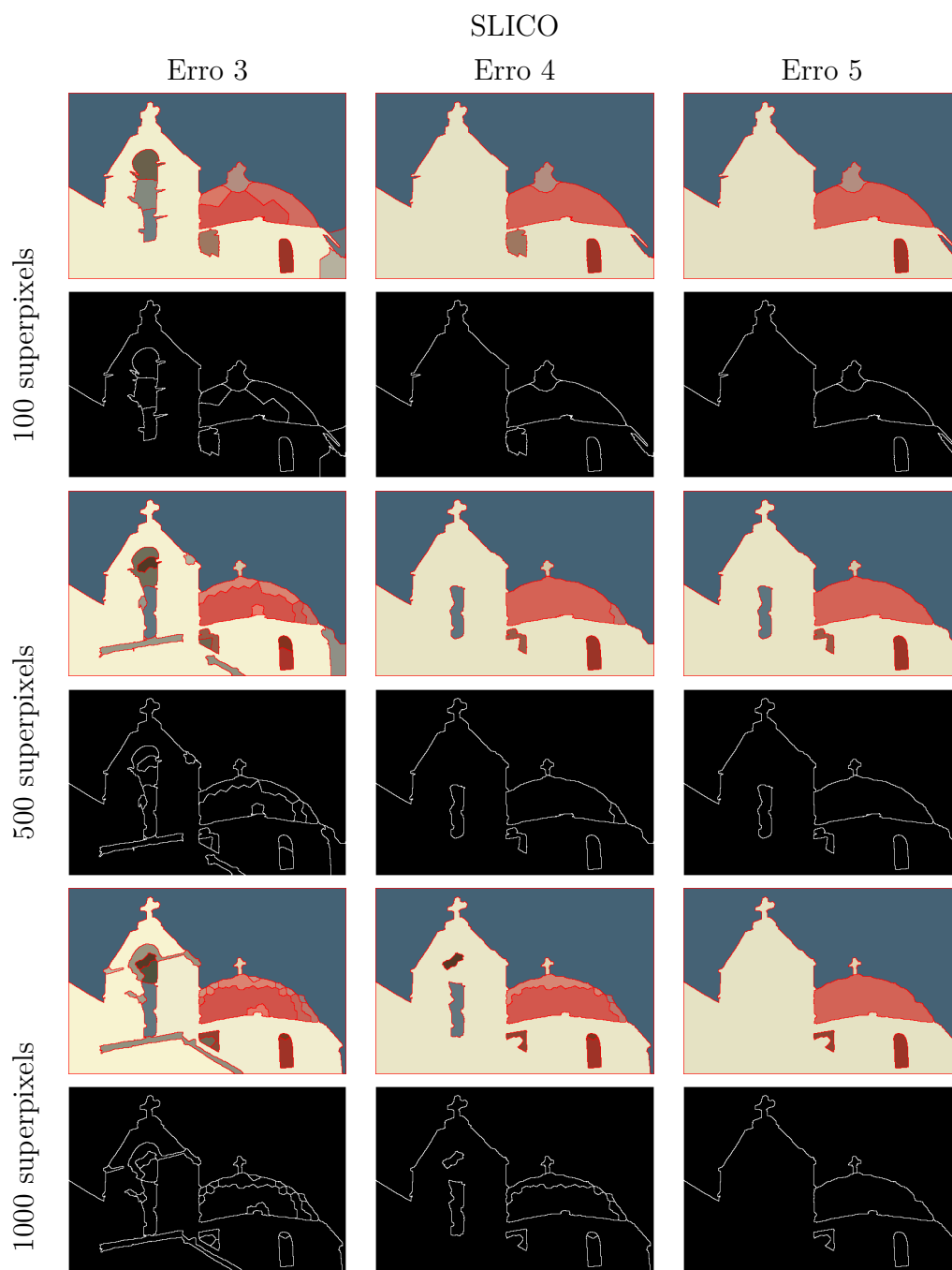


Tabela 5.10: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da igreja utilizando o SLICO como segmentação inicial.

5.2.3
Imagem da estrela do mar



Figura 5.17: Imagem da estrela do mar. Figura 5.18: *Ground truth* fornecido pelo dataset para a imagem da estrela do mar.

	SLIC	SLICO
100 superpixels		
500 superpixels		
1000 superpixels		

Tabela 5.11: Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem da estrela do mar.

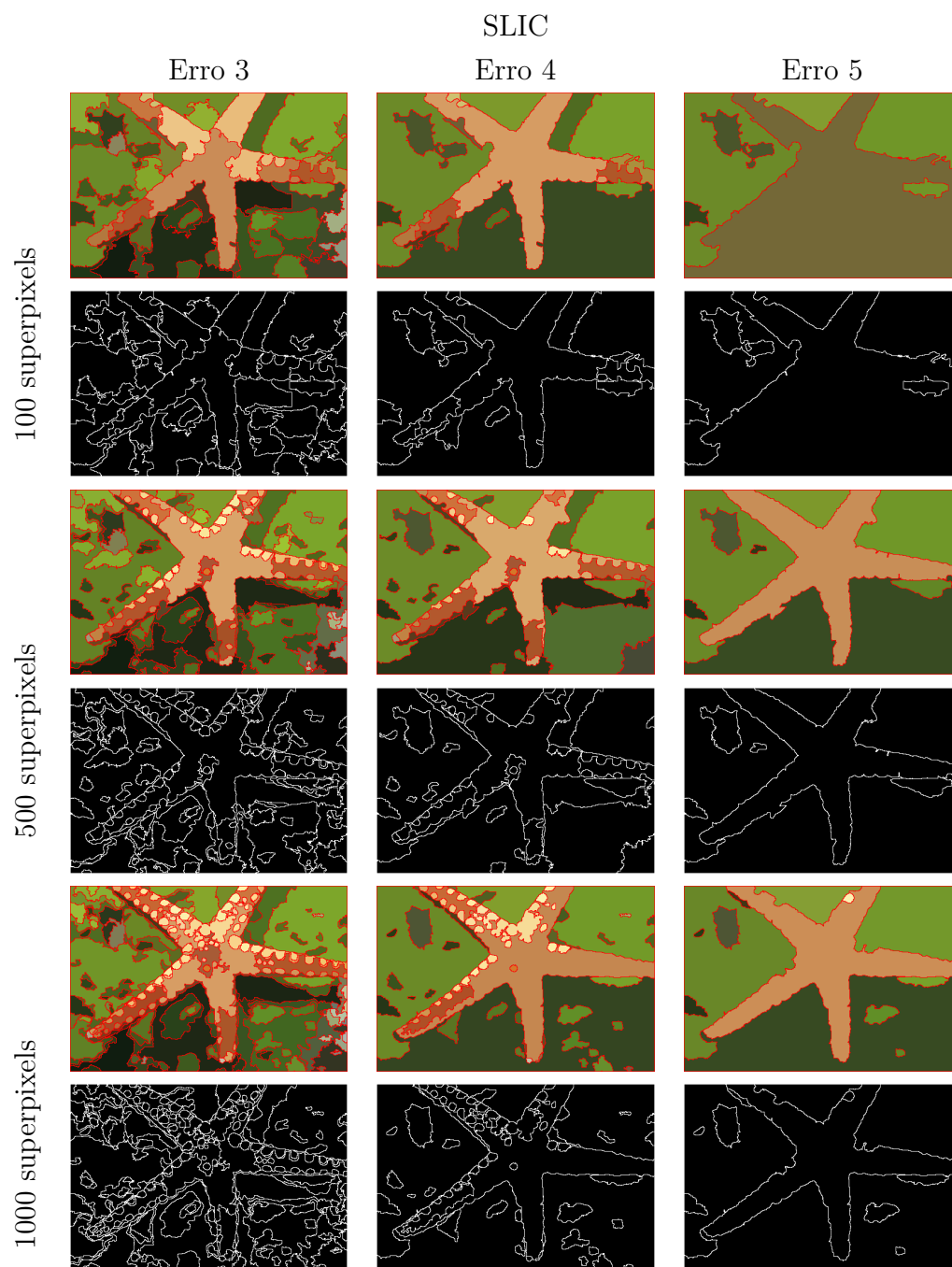


Tabela 5.12: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da estrela do mar utilizando o SLIC como segmentação inicial.

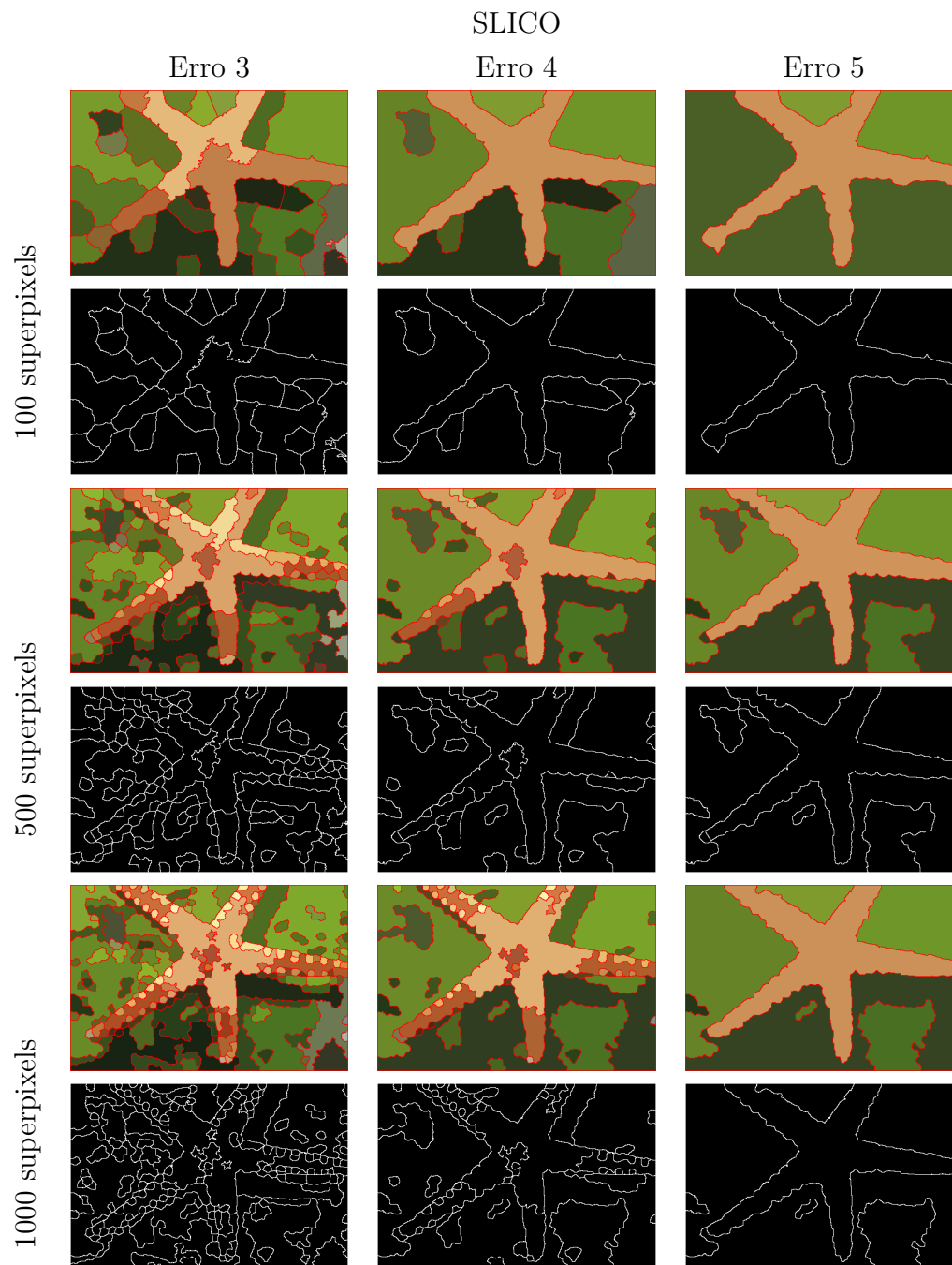


Tabela 5.13: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da estrela do mar utilizando o SLICO como segmentação inicial.

5.2.4 Imagem do coral marinho



Figura 5.19: Imagem G.

Figura 5.20: *Ground truth* fornecido pelo dataset para imagem da estrela do mar.

PUC-Rio - Certificação Digital Nº 1512331/CA

	SLIC	SLICO
100 superpixels		
500 superpixels		
1000 superpixels		

Tabela 5.14: Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem do coral marinho.

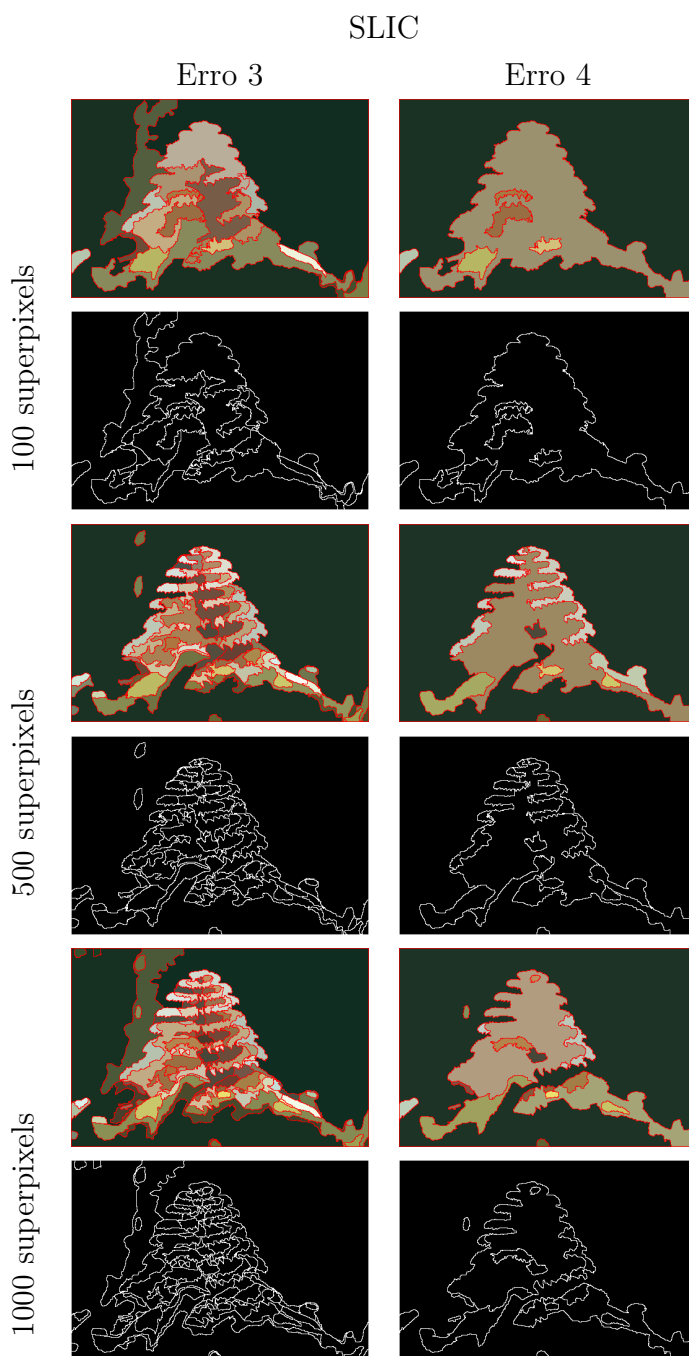


Tabela 5.15: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do coral utilizando o SLIC como segmentação inicial.

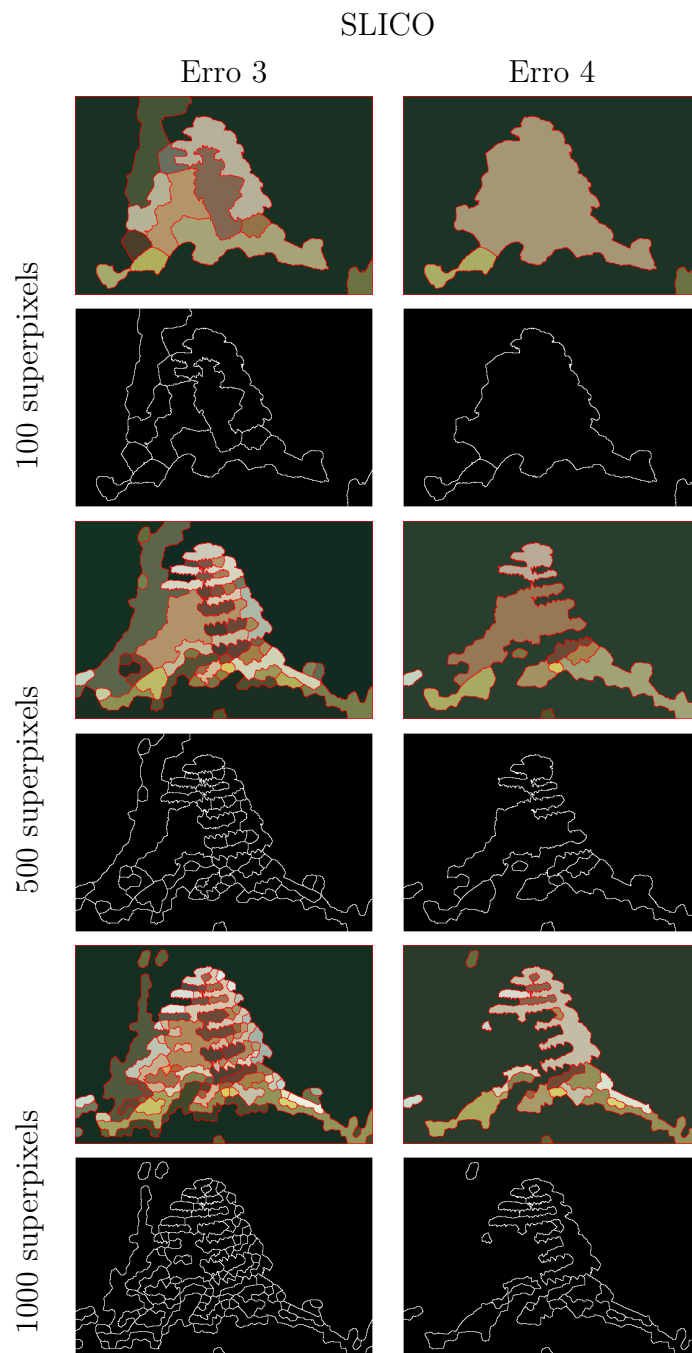


Tabela 5.16: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do coral utilizando o SLICO como segmentação inicial.

5.2.5
Imagem do menino no campo



Figura 5.21: Imagem do menino no campo. Figura 5.22: *Ground truth* fornecido pelo dataset para imagem do menino no campo .

PUC-Rio - Certificação Digital Nº 1512331/CA

	SLIC	SLICO
100 superpixels		
500 superpixels		
1000 superpixels		

Tabela 5.17: Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem do menino.

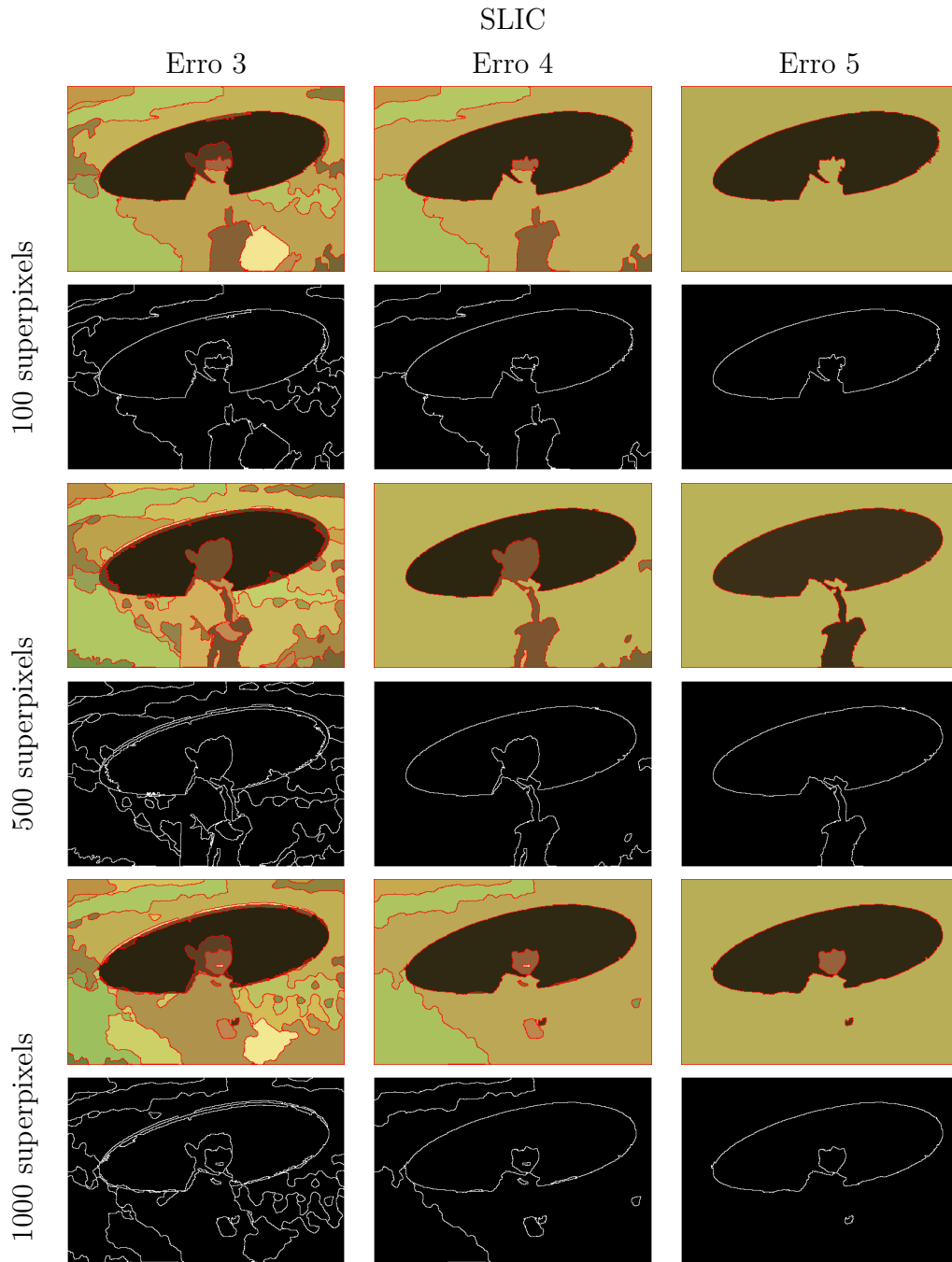


Tabela 5.18: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do menino utilizando o SLIC como segmentação inicial.

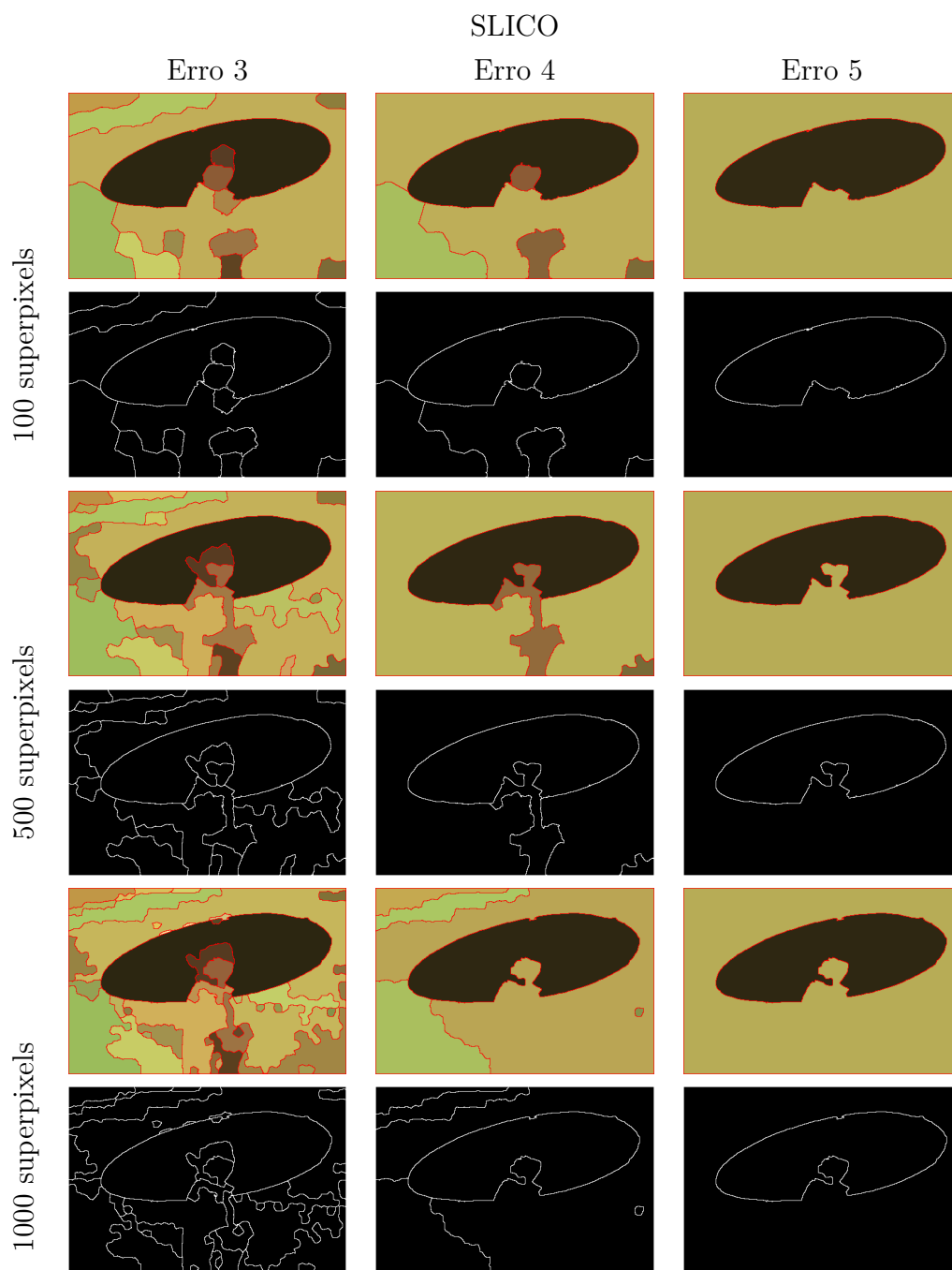


Tabela 5.19: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do menino utilizando o SLICO como segmentação inicial.

5.2.6
Imagem do peixe Nemo

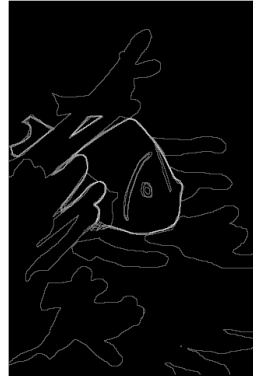


Figura 5.23: Imagem do nemo.

Figura 5.24: *Ground truth* fornecido pelo dataset para a imagem do nemo.

	SLIC	SLICO
100 superpixels		
500 superpixels		
1000 superpixels		

Tabela 5.20: Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem do peixe.

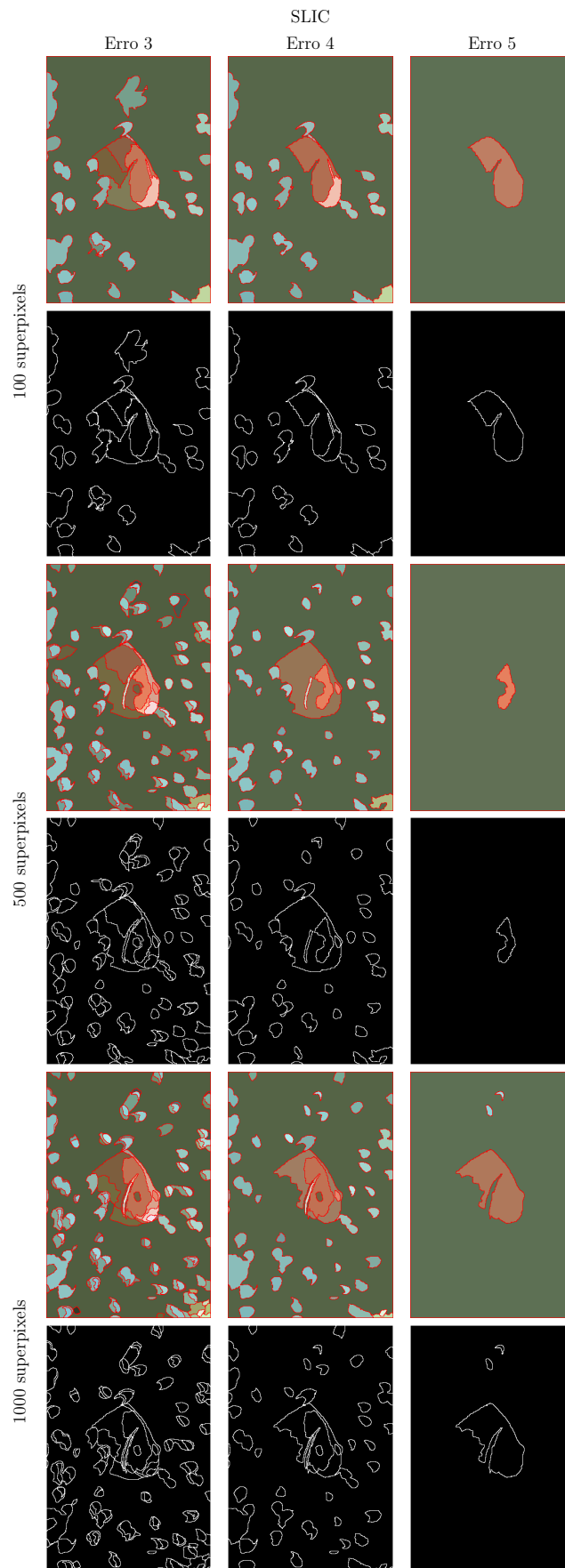


Tabela 5.21: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do peixe utilizando o SLIC como segmentação inicial.

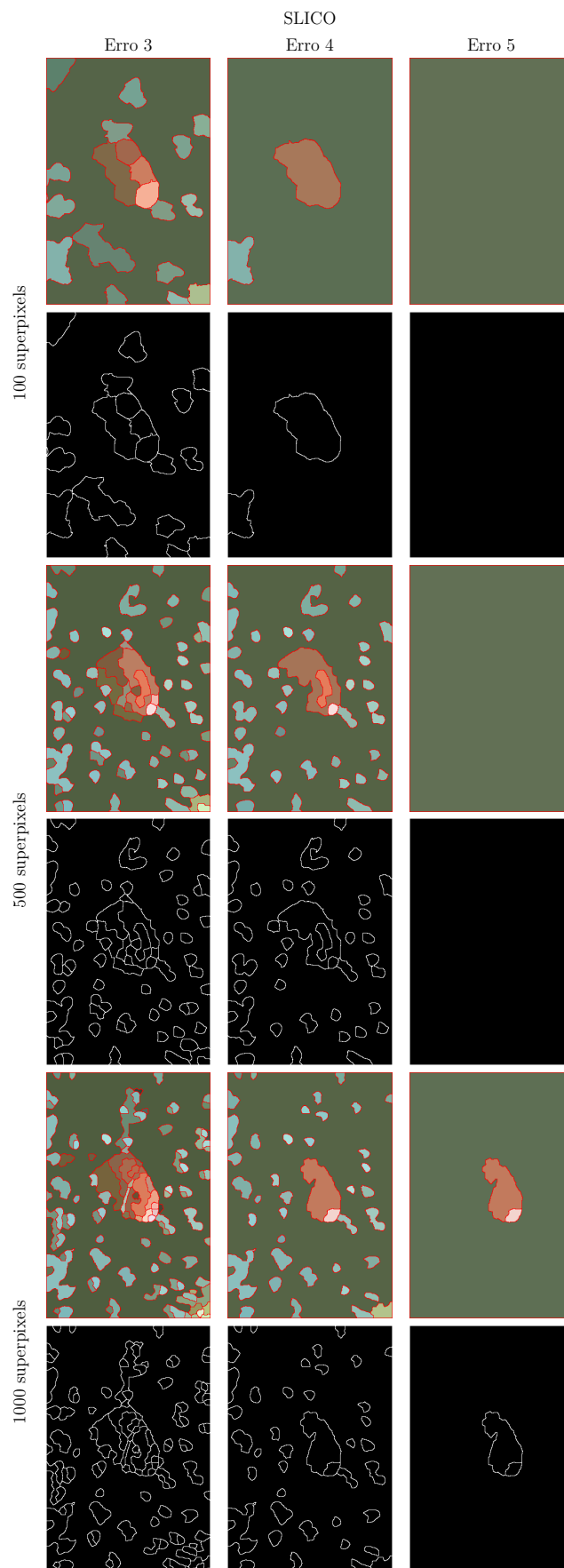


Tabela 5.22: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem do peixe utilizando o SLICO como segmentação inicial..

5.2.7
Imagem da mulher



Figura 5.25: Imagem da mulher.

Figura 5.26: *Ground truth* fornecido pelo dataset para a imagem da mulher.

	SLIC	SLICO
100 superpixels		
500 superpixels		
1000 superpixels		

Tabela 5.23: Geração de 100, 500 e 1000 superpixels, respectivamente, utilizando o SLIC e o SLICO para imagem da mulher.



Tabela 5.24: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da mulher utilizando o SLIC como segmentação inicial.

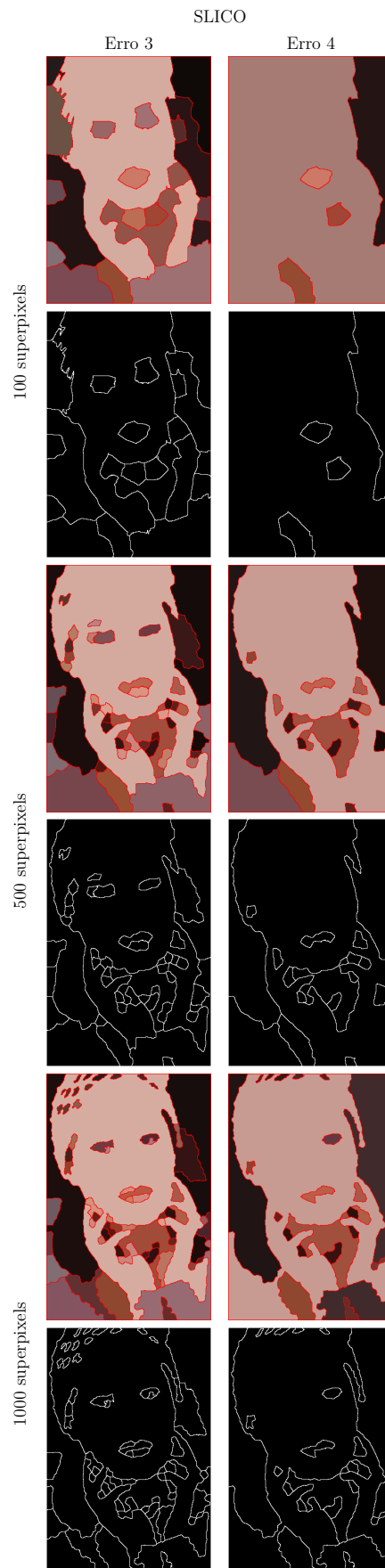


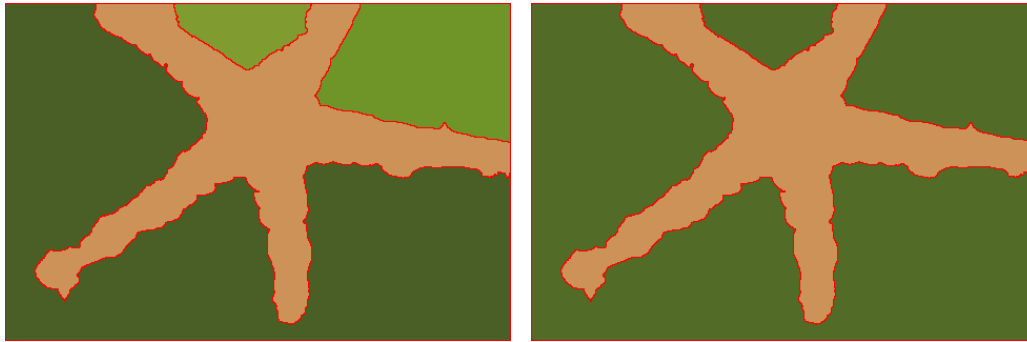
Tabela 5.25: Resultado da aplicação do método para diferentes erros e número de superpixels iniciais para a imagem da mulher utilizando o SLICO como segmentação inicial.

Analisando os resultados para as imagens selecionadas para este dataset, podemos perceber que na tarefa de detectar as bordas, os resultados usando o SLIC são melhores do que os usando o SLICO, como poderia ser esperado, dado a forma mais regulador dos superpixels do SLICO.

Alterando o número de superpixels iniciais e o erro aceito para o merge de clusters, podemos obter diferentes segmentações das imagens, gerando mais ou menos bordas. Quanto maior o erro aceito, menos bordas são geradas, perdendo algumas bordas relevantes da imagem, enquanto quanto menor o erro aceito mais bordas são geradas, porém, muitas vezes bordas que não existem na imagem original (falsos-positivos).

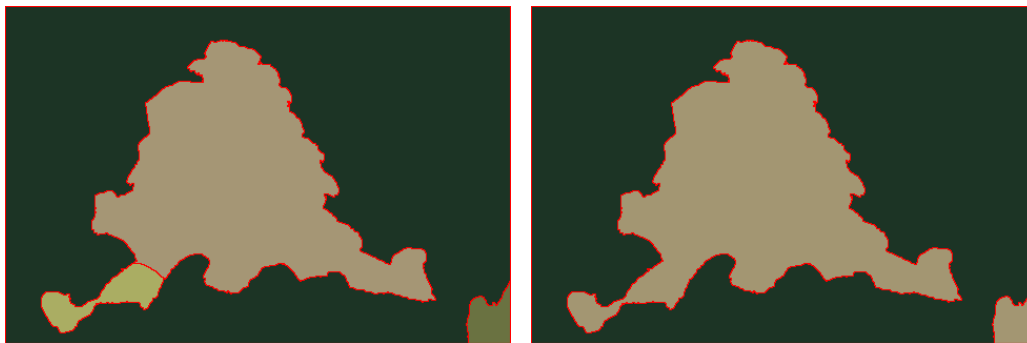
Isto pode ser observado por exemplo na imagem da igreja (Figura 5.15), na Tabela 5.9, a segmentação com 500 superpixels e erro 2, consegue detectar as bordas dos detalhes da parede do campanário, porém também identifica erroneamente bordas dentro da parte vermelha do telhado. A segmentação com 500 superpixels e erro 4 por sua vez, não identifica bordas nos detalhes da parede, nem no telhado vermelho. O mesmo acontece na imagem da mulher (Figura 5.25), onde com um erro menor o método foi capaz de identificar as bordas dos olhos e sobancelha, porém criou bordas na mão (Tabela 5.24).

Buscando apenas a segmentação de objetos, as imagens do coral (Figura 5.19) e da estrela do mar (Figura 5.17) são mais parecidas as do dataset MSRC, onde existe um objeto com um fundo, mesmo que estes sejam complexos. Para esta tarefa, pode-se observar que a segmentação utilizando o SLICO obtém melhores resultados. Para o coral, na Tabela 5.16 temos uma boa separação de objeto-fundo na segmentação com 100 superpixels e erro 4. Para a estrela do mar, por sua vez, podemos observar na Tabela 5.13 que a melhor segmentação é dada com 100 superpixels iniciais e erro 5. Para estas imagens, o pós-processamento explicado na seção 4.4 pode ser interessante, uma vez que na imagem da estrela o fundo aparece em 3 clusters separados, que no grafo que os representam não tem arestas entre si, e na imagem do coral, o coral ainda está separado em 3 clusters. O resultado do pós-processamento para estes casos são mostrados a seguir nas Figuras 5.27 e 5.28



(a) 100 superpixels e erro 5 - antes do pós-processamento. (b) 100 superpixels e erro 5 - depois do pós-processamento.

Figura 5.27: Resultado após o pós-processamento para a imagem da estrela do mar.



(a) 100 superpixels e erro 4 - antes do pós-processamento. (b) 100 superpixels e erro 4 - depois do pós-processamento.

Figura 5.28: Resultado após o pós-processamento para a imagem do coral marinho.

5.3 Imagem de ultrassom

Esta seção apresenta o resultado da aplicação do método em uma imagem de ultrassom fetal. O ultrassom utilizado é armazenado em 183 imagens de dimensões 259x259, com um byte de luminância. A Figura 5.29 ilustra algumas das camadas deste volume. Na imagem temos dois objetos principais: o feto, que aparece em tons mais claros, e o líquido amniótico que aparece em tons mais escuros. Como pode-se perceber, as imagens são de baixa resolução, muito ruidosas, e com pouco contraste entre as partes que se deseja segmentar, o que dificulta o algoritmo. Além disso, podemos perceber algumas "sombras" dentro do feto que se apresentam com cor mais semelhante à do líquido amniótico.

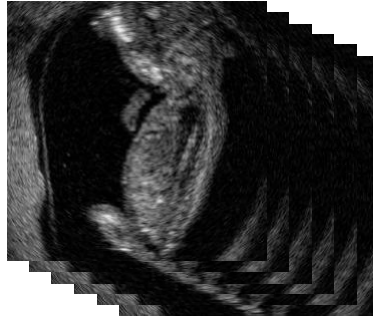
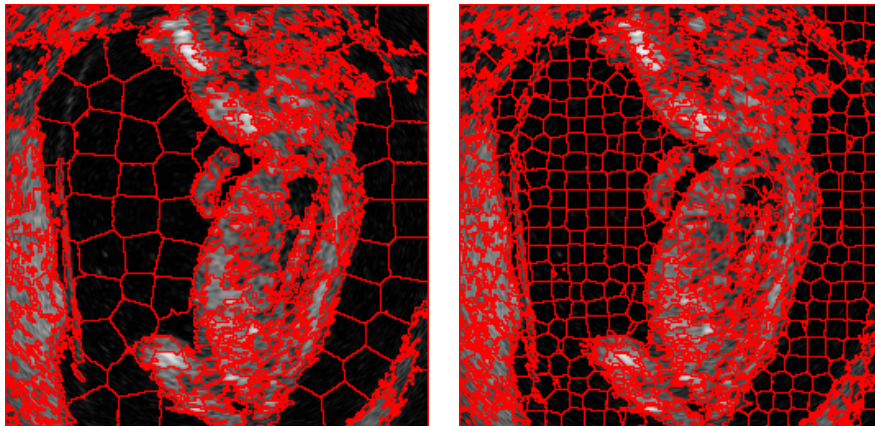


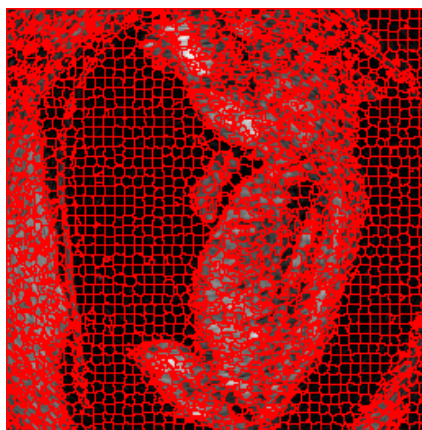
Figura 5.29: Algumas camadas do volume de ultrassom.

Nas imagem 5.30 é apresentado o resultado do SLIC supervoxel para 1000, 10000 e 50000 supervoxels, respectivamente, em uma camada do volume. As bordas dos supervoxels que interceptam esta camada são pintadas de vermelho, e por isso, a visualização pode não ser muito clara



(a) 1000 supervoxels.

(b) 10000 supervoxels.

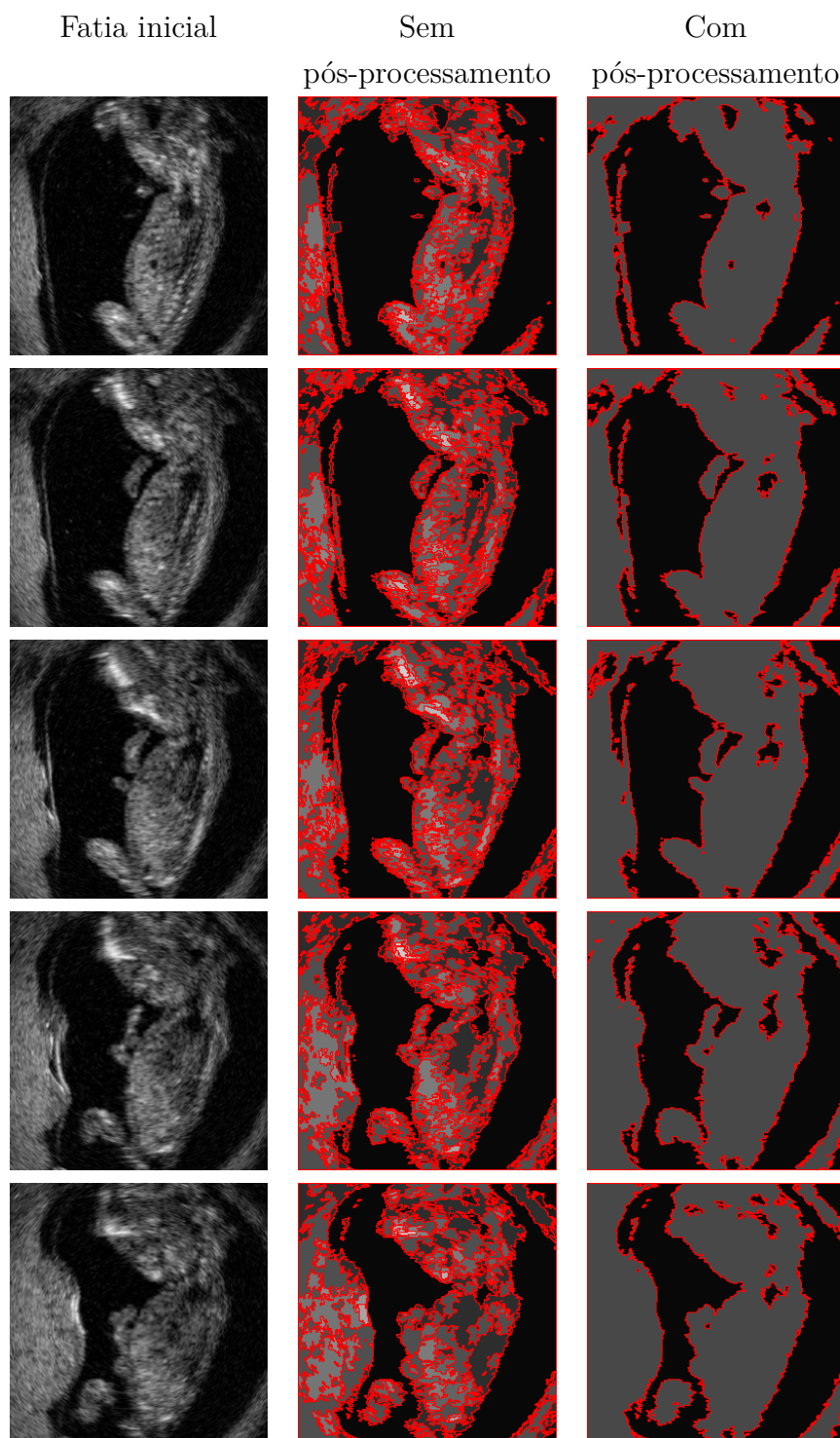


(c) 50000 supervoxels.

Figura 5.30: Geração de diferentes números de supervoxels em uma camada do volume.

Nas imagens da Tabela 5.26, os resultados antes e depois do pós-

processamento, para diferentes fatias do volume, utilizando 10000 supervoxels iniciais, que foi a quantidade que obteve melhor resultado, e erro 2.5. Este valor de erro foi escolhido pois com um valor maior, nós que não deveriam ser colapsados acabam sendo, gerando uma segmentação errada, e por isto, o pós-processamento se torna essencial.



PUC-Rio - Certificação Digital Nº 1512331/CA

Tabela 5.26: Resultado da aplicação do método apresentado em diferentes fatias.

6

Conclusão e Trabalhos Futuros

Este trabalho propôs uma metodologia para segmentação de imagens que utiliza como segmentação inicial superpixels, e a partir desta segmentação um grafo é construído, onde cada nó do grafo representa um superpixel, e cada nó tem aresta para todos os superpixels vizinhos, e através de um procedimento guloso de *merge* de nós que tem menor distância, o grafo é clusterizado até que cada nó represente um objeto contido na imagem. No procedimento de *merge* utilizamos uma distância de cores que reduz a influência da luminosidade. Na etapa de pós-processamento, a vizinhança é ignorada em benefício da união de partes desconexas dos objetos.

O método proposto necessita de supervisão para determinar a quantidade inicial de superpixels, e o erro máximo aceito para o *merge* de nós. Quanto maior o número de superpixels iniciais, mais detalhes podem ser destacados na segmentação e mais bordas são capturadas. Já quanto ao valor do erro, quanto menor for, menos clusters são colapsados, resultando em uma segmentação com mais clusters finais.

A partir dos testes apresentados, pode-se verificar que o método tem bons resultados principalmente na tarefa de segmentar objetos em imagens do tipo objeto/fundo, ou imagens mais simples. Como estes resultados são em geral muito bons, o método pode ser usado para identificar objetos, e não apenas como uma etapa de pré-processamento.

Em imagens com muitos objetos ou com muitos detalhes em diferentes escalas de tamanho, como as imagens apresentadas no dataset de Berkeley, os resultados apresentam algumas falhas.

Para a tarefa de detectar bordas proposta pelo dataset de Berkeley, que não foi o objetivo deste trabalho, o método proposto neste trabalho também apresentou alguns bons resultados e algumas falhas. Variando o erro aceito na clusterização, obtivemos resultados que detectaram as bordas principais da imagem, mas deixaram de detectar outras bordas, e resultados que detectaram a maioria das bordas da imagem, porém algumas bordas que não existem.

Já para a imagem de ultrassom, o resultado foi satisfatório, mas poderia ser melhorado fazendo um pré-processamento a fim de suavizar os ruídos, bem como um pós-processamento para eliminar, por exemplo, clusters muito escuros

dentro no bebê. Outra possível modificação que melhoraria o resultado seria a implementação de um SLICO supervoxel, que geraria uma clusterização inicial mais regular. Tais sugestões ficam como trabalhos futuros.

Ainda como trabalho futuro, sugerimos a melhoria do método para detecção de bordas de forma mais eficaz.

Como um trabalho futuro interessante, sugerimos uma modificação no algoritmo e métricas de distância para a segmentação de horizontes em imagens sísmicas.

Referências bibliográficas

- [1] ABOOD, L. K.; MOHAMMED, R. A.. **Image Segmentation Using Superpixel Based Split and Merge Method**. Abood and Mohammed Iraqi Journal of Science, 56(1A):233–237, 2015.
- [2] ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P. ; SUSSTRUNK, S.. **SLIC Superpixels**. EPFL Technical Report 149300, (June):15, 2010.
- [3] ACHANTA, R.; SHAJI, A.; SMITH, K.; LUCCHI, A.; FUA, P. ; SUSSTRUNK, S.. **Slic superpixels compared to state-of-the-art superpixel methods**. IEEE Trans. Pattern Anal. Mach. Intell., 34(11):2274–2282, Nov. 2012.
- [4] DA ROSA, R. P.. **Uma técnica otimizada de clusterização para segmentação de imagens de tc de torax de alta-resolução**. Dissertação de Mestrado, Universidade Federal de Santa Maria, 2016.
- [5] GATTASS, M.. **Cor em Sistemas Digitais**, 2018.
- [6] GONZALEZ, R. C.; WOODS, R. E.. **Digital Image Processing (2nd Edition)**. Pearson, 2002.
- [7] LIU, T.; SEYEDHOSSEINI, M. ; TASHDIZEN, T.. **Image segmentation using hierarchical merge tree**. IEEE Transactions on Image Processing, 25(10):4596–4607, Oct 2016.
- [8] MARTIN, D.; FOWLKES, C.; TAL, D. ; MALIK, J.. **A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics**. In: PROC. 8TH INT'L CONF. COMPUTER VISION, volumen 2, p. 416–423, July 2001.
- [9] NUNES, F. L. S.. **Introdução ao processamento de imagens médicas para auxílio ao diagnóstico – uma visão prática**. In: BREITMAN, Karin e ANIDO, R., editor, LIVRO DAS JORNADAS DE ATUALIZAÇÕES EM INFORMÁTICA, p. 73–126. PUC-Rio, 2006.

- [10] REN, X.; MALIK, J.. **Learning a classification model for segmentation.** In: PROC. 9TH INT'L. CONF. COMPUTER VISION, volumen 1, p. 10–17, 2003.
- [11] SCURI, A. E.. **Fundamentos da Imagem Digital.** Tecgraf/PUC-Rio, 2002.
- [12] SILVA, A. C.. **Algoritmos para Diagnostico Assistido de Nodulos Pulmonares Solitarios em Imagens de Tomografia Computadorizada.** Tese de Doutorado, PUC-Rio, Jan. 2004.
- [13] UMBAUGH, S. E.. **Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools, Second Edition.** CRC Press, Inc., Boca Raton, FL, USA, 2nd edition, 2010.