



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

SEGMENTAÇÃO DE IMAGENS UTILIZANDO ELEMENTOS DE MORFOLOGIA MATEMÁTICA

Tiago Cordeiro de Melo Nascimento

Trabalho de Graduação

Recife
ABRIL DE 2013

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Tiago Cordeiro de Melo Nascimento

**SEGMENTAÇÃO DE IMAGENS UTILIZANDO
ELEMENTOS DE MORFOLOGIA MATEMÁTICA**

Trabalho apresentado ao Programa de GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO do CENTRO DE INFORMÁTICA da UNIVERSIDADE FEDERAL DE PERNAMBUCO como requisito parcial para obtenção do grau de Bacharel em ENGENHARIA DA COMPUTAÇÃO.

Orientador: Prof. Dr. Carlos Alexandre Barros de Mello

Recife
ABRIL DE 2013

*Dedico este trabalho aos meus pais,
Miramar e Elielson.*

Agradecimentos

Sou imensamente grato

A Deus, por ter me concedido a sabedoria necessária para concluir este trabalho.

À minha família, em especial meus pais, pela sua dedicação e amor.

Ao meu orientador, Carlos, pela paciência e disposição em passar seus conhecimentos e me orientar neste trabalho.

À minha namorada, Juliana, pois seu incentivo, apoio e compreensão foram fundamentais para a conclusão deste trabalho.

À professora Rosalie, supervisora do meu estágio, pela sua flexibilidade e compreensão.

Aos amigos, pela companhia.

Aos colegas de curso, por fazer de nossa caminhada mais amena.

Resumo

A interpretação dos dados de uma imagem é uma das tarefas mais importantes em um sistema de visão computacional. Seu objetivo é obter uma descrição da imagem que seja capaz de distinguir diferentes objetos de interesse, com o mínimo de intervenção humana, sendo feita de forma confiável. O processo de interpretação não é uma atividade simples, fazendo-se necessário um processo intermediário de segmentação. A segmentação subdivide uma imagem em regiões ou objetos de interesse de acordo com algum critério. As principais abordagens para segmentação estão baseadas na detecção de regiões de descontinuidades ou similaridade. Na primeira abordagem, o objetivo é particionar a imagem com base nas mudanças abruptas nos níveis de intensidade, como a detecção de bordas da imagem. A Morfologia Matemática tem sido utilizada como ferramenta para extrair informações da imagem, sendo aplicadas na filtragem, segmentação e detecção de bordas. Um método de detecção de bordas utilizando operações morfológicas foi desenvolvido por [4], porém este método possui algumas limitações, principalmente em imagens que possui uma região de textura evidente. Este trabalho utiliza este algoritmo de detecção de bordas como base para o desenvolvimento de um algoritmo que segmente imagens com baixa profundidade de campo (*low depth of field*). Tais imagens apresentam um melhor comportamento para este detector de bordas, pois os objetos de interesse estão focados, enquanto os demais desfocados. O método apresentado neste trabalho utiliza de filtros e operações morfológicas para realizar a segmentação. Os experimentos mostram que o algoritmo proposto possui um bom comportamento para imagens com baixa profundidade de campo, sendo feita uma comparação com outro algoritmo com mesmo propósito.

Palavras-chave: segmentação de imagens, detecção de bordas, morfologia matemática, baixa profundidade de campo.

Sumário

1.	Introdução	9
1.1	Motivação	9
1.2	Objetivos.....	10
1.3	Estrutura do documento.....	10
2.	Revisão Bibliográfica	11
2.1	Imagem Digital	11
2.2	Morfologia Matemática	14
3.	Segmentação de Imagens	22
3.1	Introdução.....	22
3.2	Detecção de descontinuidades	22
3.3	Detecção de similaridades	26
3.3.1	Crescimento de regiões (<i>region-growing</i>).....	27
3.3.2	Divisão de regiões (<i>quadtree partition</i>)	28
3.3.3	Divisão e junção de regiões (<i>split-and-merge</i>).....	29
3.3.4	Divisor de águas (<i>watershed</i>).....	30
3.4	Segmentação de Imagens <i>Low-Depth-of-Field</i> Usando Filtros Morfológicos e Fusão de Regiões	31
4.	Segmentação de Imagens Utilizando Elementos de Morfologia Matemática .	33
4.1	Detecção de Bordas Utilizando Múltiplos Elementos Estruturantes.....	33
4.2	Segmentação de Imagens <i>Low-Depth-of-Field</i>	35
5.	Experimentos	38
5.1	Experimento 1	38
5.1.1	Metodologia	38
5.1.2	Resultados	39
5.1.3	Discussão.....	43
5.2	Experimento 2	44
6.	Conclusão.....	48
6.1	Trabalhos Futuros	49
	Referências.....	50
	Anexos	51
	Resultados dos experimentos.....	52

Lista de Figuras

Figura 1 - Imagem em tons de cinza e imagem em preto e branco.....	11
Figura 2 - Canais vermelho, verde e azul de uma imagem.....	12
Figura 3 - Efeitos da redução da resolução sobre uma imagem	13
Figura 4 - Diferentes quantizações para a mesma imagem.	14
Figura 5 - Efeito da reflexão e translação sob um conjunto de pixels.....	15
Figura 6 - Elementos estruturantes e seus arranjos matriciais retangulares.	15
Figura 7 - Conversões para arranjo matricial retangular.	16
Figura 8 - Exemplo da operação de dilatação.....	17
Figura 9 - Exemplo da operação de erosão.....	17
Figura 10 - Operação de dilatação em uma imagem em tons de cinza	18
Figura 11 - Exemplo das operações de abertura e fechamento.	19
Figura 12 - Extração de fronteira através de operações morfológicas.	20
Figura 13 - Preenchimento morfológico de buracos	21
Figura 14 - Intensidade e direção da borda através do vetor gradiente.	23
Figura 15 - Máscaras usadas para implementar as Equações 15 e 16	24
Figura 16 - Máscara utilizada para implementar a Equação 20	24
Figura 17 - Máscaras de Roberts, Prewitt e Sobel.....	25
Figura 18 - Comparação das bordas entre Sobel e Roberts.	25
Figura 19 - Detecção das bordas utilizando o operador de Sobel.	26
Figura 20 - Exemplo do algoritmo de segmentação de crescimento de regiões.....	28
Figura 21 - Exemplo da partição <i>quadtree</i>	28
Figura 22 - Segmentação baseada na divisão de regiões.....	29
Figura 23 - Processo de inundação da segmentação por divisor de águas	30
Figura 24 - Segmentação baseada em <i>Watershed</i>	31
Figura 25 - Imagem do tipo <i>low-depth-of-field</i>	32
Figura 26 - Algoritmo de segmentação de imagens (<i>low-DOF</i>) de Kim	32
Figura 27 - Conjunto de oito elementos estruturantes em uma janela 5x5.....	34
Figura 28 - Resultado do algoritmo de detecção de bordas.....	35
Figura 29 - Segmentação da imagem pelo algoritmo proposto.	37
Figura 30 - Segmentação da imagem utilizando o algoritmo proposto.....	39
Figura 31- Segmentação da imagem utilizando o algoritmo proposto.....	39
Figura 32 - Segmentação da imagem utilizando o algoritmo proposto.....	40
Figura 33 - Segmentação da imagem utilizando o algoritmo proposto.....	40
Figura 34 - Segmentação da imagem utilizando o algoritmo proposto.....	40
Figura 35 - Comparação entre os métodos de segmentação de imagens <i>low-DOF</i> . 41	
Figura 36 - Comparação entre os métodos de segmentação de imagens <i>low-DOF</i> . 42	
Figura 37 - Imagem padrão ouro usada para análise da robustez do algoritmo	44
Figura 38 - Composições geradas entre o padrão ouro e um <i>background</i>	45
Figura 39 – Segmentação das imagens da Figura 38.....	46
Figura 40 - Comparação entre os métodos de segmentação de imagens <i>low-DOF</i> . 52	
Figura 41 - Comparação entre os métodos de segmentação de imagens <i>low-DOF</i> . 53	
Figura 42 - Comparação entre os métodos de segmentação de imagens <i>low-DOF</i> . 54	
Figura 43 - Comparação entre os métodos de segmentação de imagens <i>low-DOF</i> . 55	

Lista de Tabelas

Tabela 1 - Valores utilizados nos parâmetros do algoritmo de segmentação.....	38
Tabela 2 - Avaliação quantitativa da segmentação aplicada sobre composições.....	46
Tabela 3 - Valores utilizados nos parâmetros do algoritmo de segmentação.....	51

1. Introdução

1.1 Motivação

Segundo [1], a capacidade humana de captar, processar e interpretar o grande volume de dados de natureza visual tem estimulado, nas últimas décadas, o desenvolvimento de técnicas e sistemas computacionais cada vez mais avançados. Entretanto, desenvolver um sistema computacional capaz de reagir a tais estímulos visuais, ainda é um grande desafio para os pesquisadores. Entender como funciona o sistema visual humano, sua capacidade de aprendizagem, e sua habilidade de inferir a partir de estímulos visuais, são de grande importância para o aprimoramento dessas técnicas para análise automática por um computador.

Num sistema de visão computacional, uma das tarefas mais importantes é a de interpretação de uma imagem. Seu objetivo principal é obter uma descrição da imagem que contenha informação suficiente para distinguir entre diferentes objetos de interesse, com o mínimo de intervenção humana, e de forma confiável.

A interpretação dos dados nas imagens não é uma tarefa trivial, para isso é necessário um processamento intermediário, a segmentação. Esta etapa consiste em subdividir a imagem em regiões ou objetos a fim de que sua representação seja simplificada, auxiliando o processo de interpretação das imagens. O nível de detalhe em que a subdivisão é realizada depende de cada problema.

É de fundamental importância que o processo de segmentação identifique corretamente a localização, a forma, e a topologia dos objetos para que as informações resultantes do processo de análise de imagens sejam confiáveis. Após a segmentação, cada objeto é representado a partir de suas propriedades geométricas e topológicas.

As principais abordagens para segmentação são baseadas nas propriedades básicas dos níveis de cinza da imagem, visando detectar regiões de descontinuidade ou similaridade. Na primeira categoria, o objetivo é particionar a imagem com base nas mudanças abruptas nos níveis de cinza, como a detecção de pontos isolados, linhas, ou bordas na imagem. Na segunda categoria, o objetivo é agrupar em regiões pontos na imagem que possuam valores similares, de acordo com um conjunto de características [1].

A morfologia matemática é uma ferramenta utilizada no processamento de imagens para extrair informações relativas à forma e topologia da imagem. Suas operações estão fundamentadas na teoria matemática dos conjuntos, sendo aplicadas em várias áreas como: realce, filtragem, segmentação, detecção de bordas, entre outras. As operações morfológicas aplicam um conjunto chamado elemento estruturante na imagem, a fim de examinar propriedades de interesse [2].

Dentro desse contexto, os autores em [4] desenvolveram um algoritmo de detecção de bordas que utilizava múltiplos elementos estruturantes a fim de extrair características complexas da borda da imagem. Esse algoritmo, porém, possuía limitações, detectando regiões de textura indesejadas na imagem, além de ser sensível ao ruído.

Analisando o algoritmo proposto em [4], percebeu-se que o mesmo possuía uma melhor precisão em imagens de baixa profundidade de campo (*low depth of field*), no qual o objeto de interesse encontra-se no foco da imagem, e os demais estão desfocados.

Assim, este trabalho consiste em propor um algoritmo de segmentação de imagens para detecção de bordas baseado na proposta apresentada em [4], utilizando operações de morfologia matemática, aplicado a imagens de baixa profundidade de campo.

1.2 Objetivos

Objetivo geral:

- Propor um algoritmo de segmentação de imagens baseado na detecção de bordas da imagem e operações de morfologia matemática.

Objetivos específicos:

- Estudar algoritmos de detecção de bordas e segmentação de imagens.
- Testar e comparar com outras abordagens de segmentação.
- Avaliar possíveis contribuições para melhorar o algoritmo desenvolvido.

1.3 Estrutura do documento

Capítulo 2 – Revisão Bibliográfica. Neste capítulo são introduzidos conceitos básicos sobre imagem digital, morfologia matemática e suas operações para melhor entendimento deste trabalho.

Capítulo 3 – Segmentação de Imagens. Neste capítulo é apresentado um breve resumo sobre os algoritmos clássicos de segmentação de imagens, e também apresenta uma breve descrição do algoritmo de segmentação de imagens com baixa profundidade de campo (*low depth of field*).

Capítulo 4 – Segmentação de Imagens Utilizando Elementos de Morfologia Matemática. Neste capítulo é apresentado o método de segmentação de imagens proposto neste trabalho e se divide em duas partes. Na primeira é feita uma descrição do algoritmo de detecção de bordas utilizado por este trabalho, sendo comentadas suas limitações. Na segunda parte, é descrita a nova metodologia proposta para segmentação de imagens.

Capítulo 5 – Experimentos. Este capítulo apresenta e discute os resultados dos experimentos realizados com o algoritmo de segmentação proposto, utilizando imagens *low depth of field*, fazendo também uma comparação com o algoritmo proposto em [5].

Capítulo 6 – Conclusão. Neste capítulo são apresentadas as considerações finais, contribuições, limitações e trabalhos futuros.

Anexos – Nesta Seção, são incluídas imagens adicionais mostrando os resultados do algoritmo proposto e outras comparações com o algoritmo proposto em [5].

2. Revisão Bibliográfica

Neste capítulo, são introduzidos alguns termos e conceitos básicos sobre processamento de imagem digital e morfologia matemática e suas operações, que são abordados durante este trabalho.

2.1 Imagem Digital

O processamento digital de imagens consiste em um conjunto de técnicas para capturar, representar e transformar imagens com o auxílio de um computador. O uso de tais técnicas permite extrair e identificar informações das imagens e melhorar a qualidade das informações visuais para facilitar a percepção humana e a interpretação automática por um computador [2].

Para que se possa representar e manipular uma imagem em um computador, é necessária a definição de um modelo matemático adequado da imagem. Uma definição natural é considerar uma imagem como uma função bidimensional de intensidade luminosa, denotada por $f(x, y)$, fornecendo o brilho, ou a intensidade luminosa naquele ponto [1].

Quando as coordenadas (x, y) e os níveis de intensidade luminosa f são quantidades finitas e discretas, chamamos de imagem digital. O processamento digital de imagens se dá por imagens digitais, que são compostas por um número finito de elementos, cada um com sua posição e nível de intensidade. Esses elementos são denominados *pixels* [2].

As imagens que possuem apenas um canal para representar o nível de intensidade de cada *pixel* são chamadas de imagens monocromáticas. Neste caso, a informação de cada *pixel* está associada à presença ou ausência de luz. Portanto, não existe informação suficiente para representar a cor de um *pixel*. Apesar de possuir apenas essa informação, tais imagens podem possuir diferentes tons, sendo classificadas como imagens em tons de cinza. Não devemos confundir tais imagens com as imagens em preto e branco, também conhecidas como binárias, que possuem apenas duas cores possíveis (Figura 1).



Figura 1 – (a) Imagem em tons de cinza, (b) imagem em preto e branco.

Uma imagem colorida é conhecida como imagem multibanda, pois possui mais de um canal para representar a cor de um *pixel*. Uma representação comum para imagens

coloridas é o sistema RGB. A composição dos canais R (Red – Vermelho), G (Green - Verde) e B (Blue - Azul), formam a cor de cada *pixel* numa imagem (Figura 2).

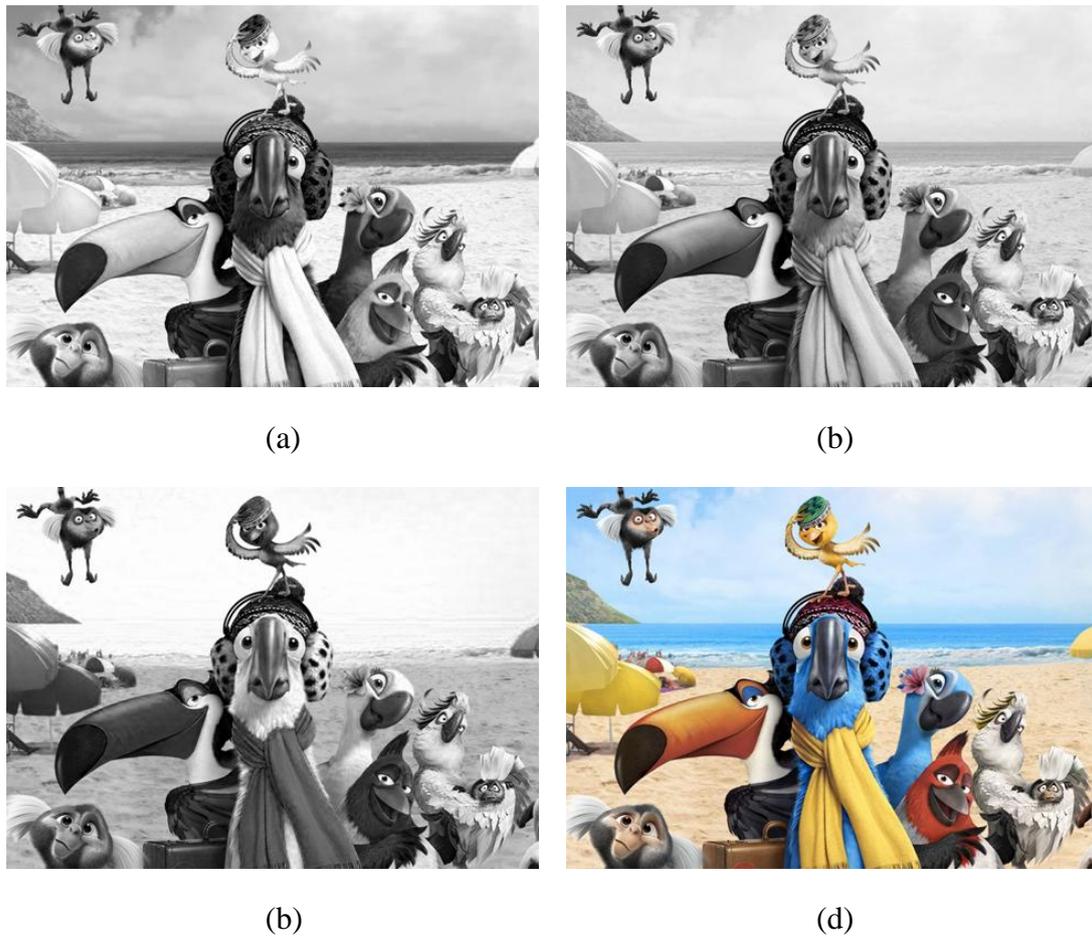


Figura 2 - Intensidade dos canais (a) vermelho, (b) verde, (c) azul e (d) imagem formada a partir da composição dos três canais.

As imagens digitais são geradas a partir de uma operação chamada digitalização, que é constituída de duas partes: amostragem e quantização.

A amostragem discretiza o domínio de definição da imagem contínua nas dimensões x e y , criando uma matriz de $M \times N$ amostras, com $0 \leq x \leq M - 1$ e $0 \leq y \leq N - 1$. Nessa operação, a imagem contínua é aproximada por uma matriz de dimensão de M *pixels* na horizontal, e N *pixels* na vertical [1]. A dimensão de um *pixel* ao longo do eixo x ou do eixo y , está associado com o espaçamento físico entre as amostras. O produto MN da matriz é definido como a resolução espacial, ou resolução da imagem. Pode-se perceber que quanto maior a resolução da imagem, maior a quantidade de amostras coletadas, maior é o nível de informações contidas naquela imagem digital (Figura 3).

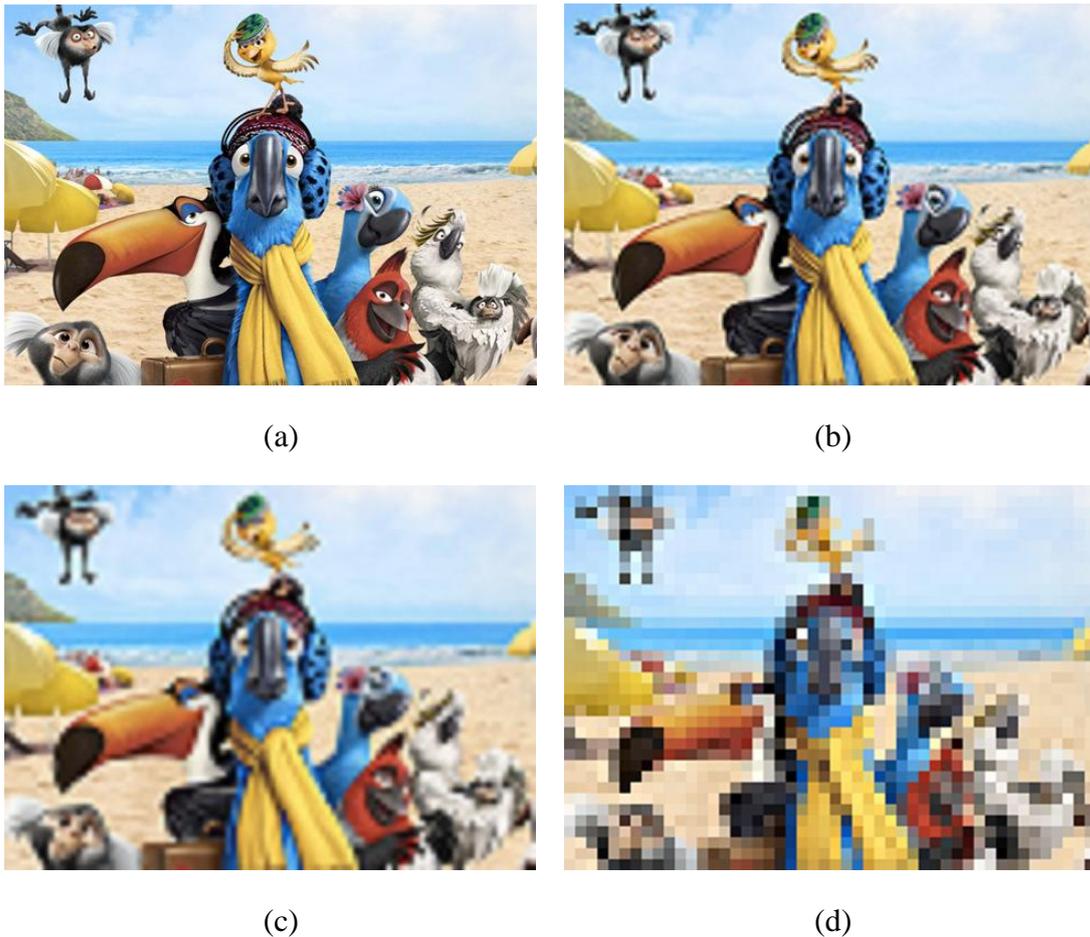


Figura 3 - Efeitos da redução da resolução sobre uma imagem (em *pixels*): (a) 480 x 347, (b) 168 x 122, (c) 96 x 70, (d) 48 x 35. As imagens de menores dimensões estão apresentadas ampliadas para melhor visualização dos detalhes.

Quantização é uma operação que converte cada nível de intensidade da matriz de amostras em quantidades discretas, escolhendo-se um número inteiro L de níveis de intensidade. Cada *pixel* possui um nível de intensidade $L_{min} \leq f(x, y) \leq L_{max}$, tal que o intervalo $[L_{min}, L_{max}]$ é denominado escala de cinza [1].

Em imagens digitais, a quantização é definida em função do número de bits utilizados para representar os níveis de intensidade. Normalmente os valores mínimo e máximo da escala de cinza estão compreendidos no intervalo $[0, L - 1]$, onde $L = 2^b$, e b é o número de bits utilizados na representação das cores da imagem. Em imagens em tons de cinza é comum que sejam utilizados 8 bits para representar os níveis de intensidade. Temos, então, $L = 2^8 = 256$ tons de cinza, logo, a escala de níveis de cinza está compreendida no intervalo $[0, 255]$. Uma convenção é atribuir a cor preta ao nível de cinza mais escuro (o valor 0) e atribuir a cor branca ao nível de cinza mais claro (o valor 255). O efeito de diferentes quantizações para uma mesma imagem pode ser observado na Figura 4, exemplificando em imagens coloridas.

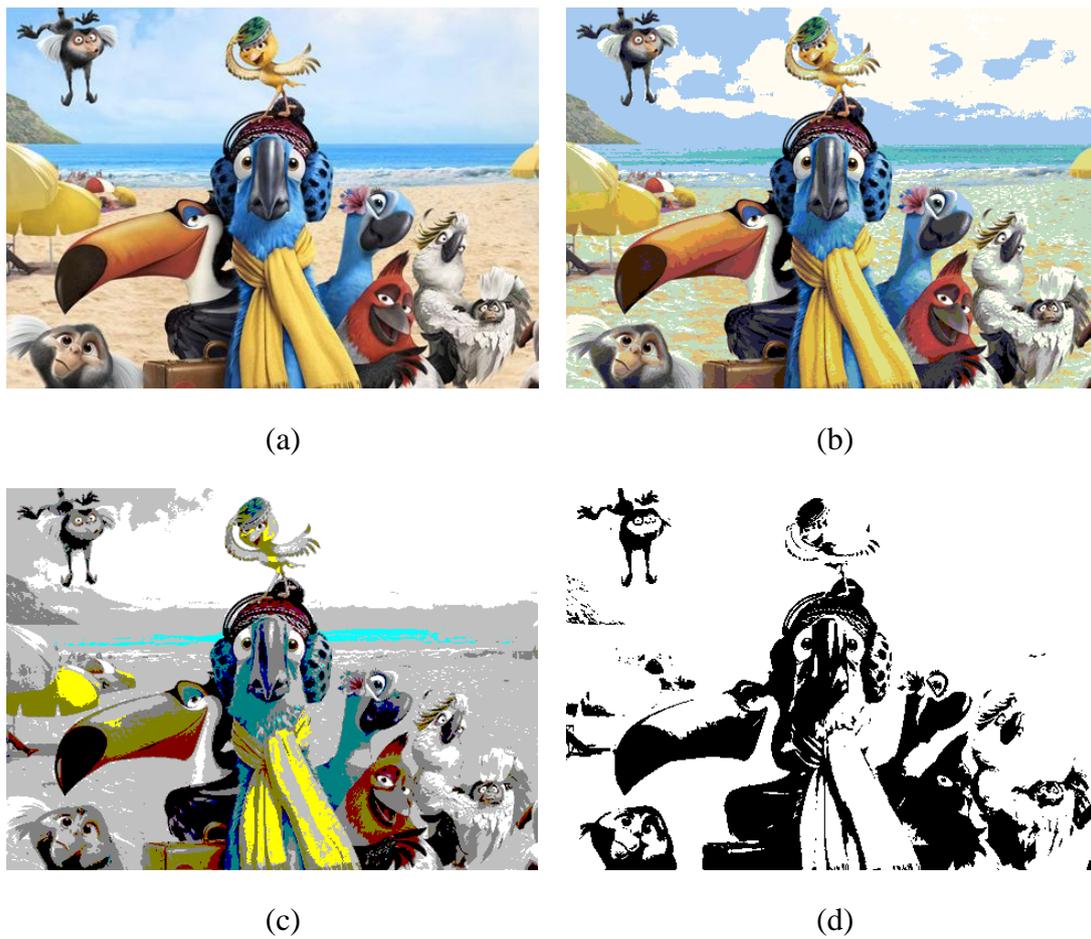


Figura 4 - Diferentes quantizações para a mesma imagem. (a) Imagem com 24 bits de cor, (b) 8 bits, (c) 4 bits e (d) 1 bit.

2.2 Morfologia Matemática

A palavra morfologia pode ser compreendida como a forma e a estrutura de um objeto, ou o arranjo e as inter-relações entre as partes de um objeto. O estudo da morfologia matemática tem como objetivo descrever ou analisar a forma de uma imagem digital, permitindo a construção de operadores úteis para realizar tais objetivos [3].

A morfologia matemática é uma área relativamente recente, sua teoria foi formulada na década de 1960, ao passo que os computadores puderam torná-la prática. Inicialmente foi desenvolvida para manipular imagens binárias, e posteriormente estendida para imagens em níveis de cinza [1]. Seus fundamentos estão baseados na teoria dos conjuntos para representar os objetos em uma imagem. A ideia por trás da morfologia matemática é considerar que as imagens são constituídas por um conjunto de elementos, os *pixels*, que se agrupam numa estrutura bidimensional (forma) [3].

As operações morfológicas ajudam a melhorar características das formas dos objetos presentes numa imagem, podendo ser, por exemplo, contados ou reconhecidos. Além das definições básicas da teoria dos conjuntos, é importante mencionar as definições de reflexão e translação de conjuntos (Figura 5).

A reflexão pode ser definida a partir de um conjunto de *pixels* B que representa um objeto em uma imagem, então \hat{B} é o conjunto de pontos de B cujas coordenadas (x, y) foram substituídas por $(-x, -y)$ [2]:

$$\hat{B} = \{w \mid w = -b, \text{para } b \in B\} \quad (\text{Eq. 1})$$

A translação pode ser definida a partir de um conjunto de *pixels* B que representa um objeto em uma imagem, então $(B)_z$ é o conjunto de pontos em B cujas coordenadas (x, y) foram substituídas por $(x + z_1, y + z_2)$ [2]:

$$(B)_z = \{c \mid c = b + z, \text{para } b \in B\} \quad (\text{Eq. 2})$$

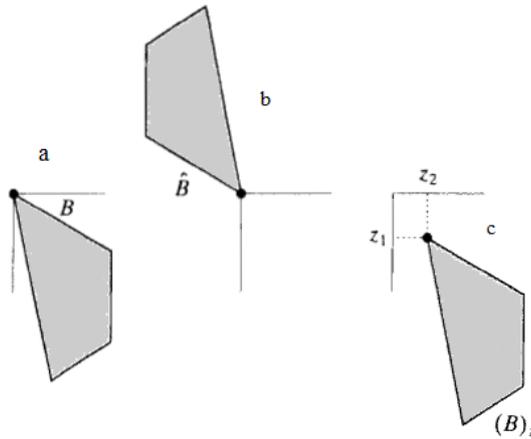


Figura 5 – Efeito das operações de reflexão e translação sob um conjunto de *pixels*. (a) Conjunto original, (b) sua reflexão em torno da origem, (c) sua translação por z .

A reflexão e a translação são muito utilizadas nas operações baseadas nos elementos estruturantes (ES). Os elementos estruturantes podem ser entendidos como conjuntos que são utilizados para explorar e detectar propriedades de interesse em uma imagem. A Figura 6 mostra, em sua primeira linha, alguns exemplos de elementos estruturantes, em que cada quadrado sombreado representa um membro do ES (ou uma região ativa), e os pontos pretos representam sua origem. A escolha da origem nem sempre é o seu centro de gravidade, porém quando não é explicitada sua posição, através do ponto, e o ES é simétrico, assumimos que o ponto de simetria coincide com sua origem.

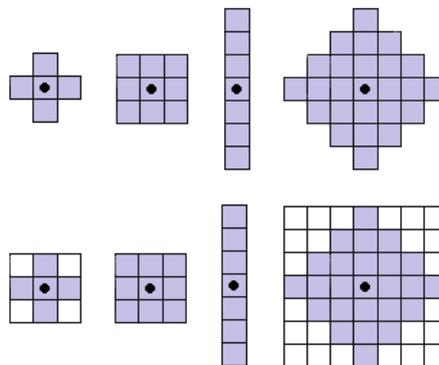


Figura 6 - Na primeira linha: exemplos de elementos estruturantes. Na segunda linha: os mesmos elementos estruturantes em arranjos matriciais retangulares.

Ao se trabalhar com imagens digitais, é necessário dispor os elementos estruturantes num arranjo matricial retangular. Para isso, é acrescentada a menor quantidade possível de *pixels* de fundo (ou região inativa). A segunda linha da Figura 6 mostra esse arranjo [2]. De forma semelhante, o conjunto dos *pixels* que formam a imagem precisa ser arranjado na forma matricial, adicionando *pixels* de fundo (Figura 7).

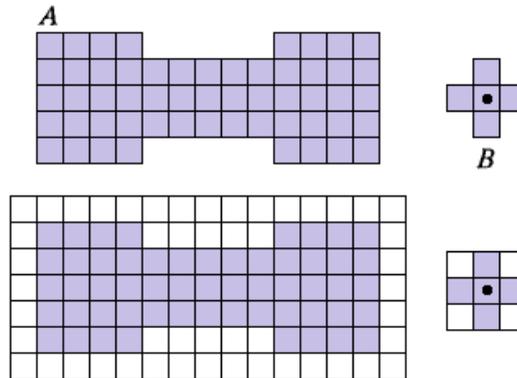


Figura 7 - Conversão da imagem (A) e do elemento estruturante (B) para o arranjo matricial retangular (figura copiada de [2]).

As operações básicas da morfologia matemática são a dilatação, no qual um conjunto de *pixels* é adicionado à fronteira da imagem de acordo com um determinado padrão, e a erosão, no qual os *pixels* correspondentes a um determinado padrão são removidos da fronteira da imagem. Entretanto, as definições dessas operações dependem do tipo de imagem a ser processada, como binária ou em tons de cinza [3].

Em se tratando de imagens binárias, a dilatação de uma imagem A por um elemento estruturante B pode ser entendida como o conjunto de todos os deslocamentos, z , de forma que \hat{B} e A se sobreponham pelo menos por um elemento.

$$A \oplus B = \{z \mid [\langle \hat{B}_z \rangle \cap A] \subseteq A\} \quad (Eq. 3)$$

A Figura 8 mostra um exemplo da dilatação de uma imagem binária A por um elemento estruturante B . A linha tracejada mostra as fronteiras do conjunto original para referência, já a linha contínua mostra os limites além do qual os deslocamentos da origem de \hat{B} por z resultariam em uma intersecção vazia entre \hat{B} e A .

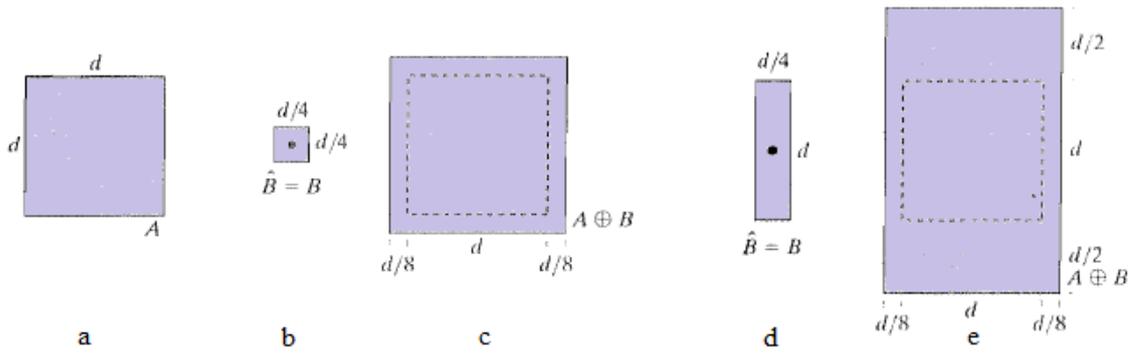


Figura 8 - (a) Conjunto A . (b) Elemento estruturante B . (c) Dilatação de A por B . (d) Outro elemento estruturante B alongado. (e) Dilatação de A por esse ES (figura copiada de [2]).

A erosão de uma imagem A por um elemento estruturante B pode ser entendida como o conjunto de todos os pontos z de forma que B , transladado por z , está contido em A .

$$A \ominus B = \{z \mid \langle B_z \rangle \subseteq A\} \quad (\text{Eq. 4})$$

A Figura 9 mostra um exemplo de erosão com os mesmos conjuntos A e B . A linha sólida mostra o limite além do qual os deslocamentos adicionais da origem de B fariam com que o elemento estruturante deixasse de estar completamente contido em A .

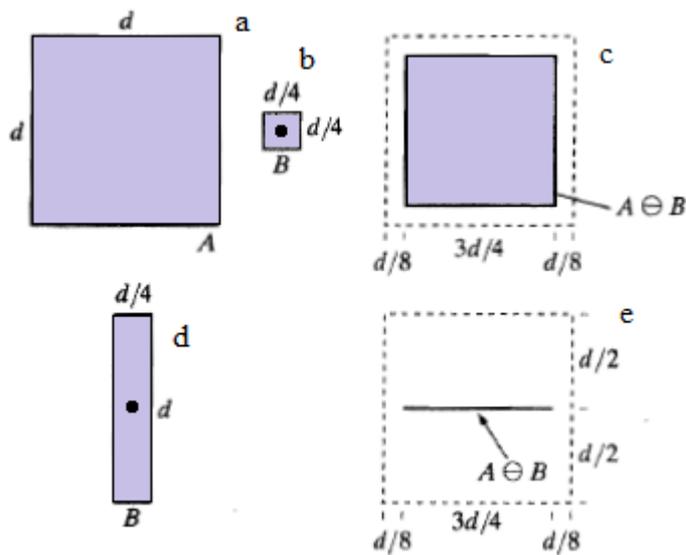


Figura 9 - (a) Conjunto A . (b) Elemento estruturante B . (c) Erosão de A por B . (d) Outro elemento estruturante B alongado. (e) Erosão de A por esse ES (figura copiada de [2]).

A dilatação tem como efeito aumentar ou engrossar os objetos em uma imagem binária. Já a erosão provoca a diminuição ou afinamento desses objetos. A forma de como esses objetos vão ser dilatados ou erodidos, e a extensão dessas operações, depende do formato e tamanho do elemento estruturante.

Para imagens em tons de cinza, as operações de dilatação e erosão podem ser definidas de uma maneira diferente. Na dilatação, o tom do *pixel* de saída é o maior tom de todos os *pixels* que estão contidos na região ativa do elemento estruturante. Já na erosão, o tom do *pixel* de saída é definido como o tom mínimo de todos os *pixels* que estão contidos na região ativa definida pelo elemento estruturante. Podemos expressar matematicamente as operações de dilatação e erosão, respectivamente, como se segue [2].

$$(A \oplus B)(x, y) = \max_{(s,t) \in B} \{A(x - s, y - t)\} \quad (Eq. 5)$$

$$(A \ominus B)(x, y) = \min_{(s,t) \in B} \{A(x + s, y + t)\} \quad (Eq. 6)$$

A Figura 10 ilustra a dilatação de uma imagem em tons de cinza. Note como o elemento estruturante define a vizinhança do *pixel* de interesse, que está circulado. Pode-se observar também que é selecionado o *pixel* de maior tom dentro da vizinhança definida pelo elemento estruturante para o *pixel* correspondente da imagem de saída [6].

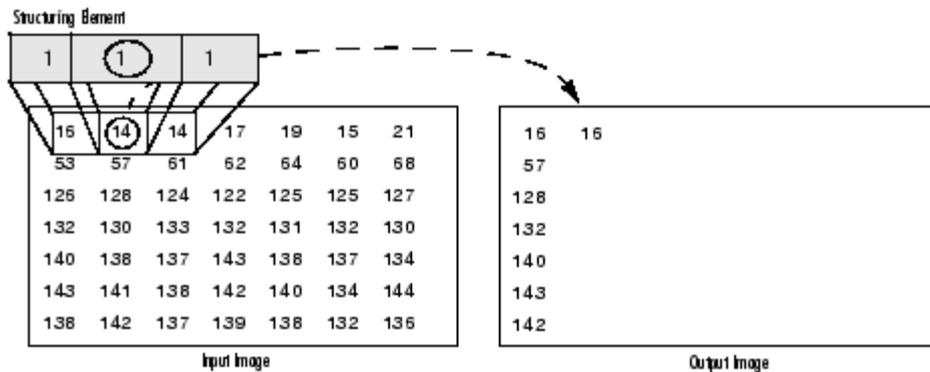


Figura 10 - Operação morfológica de dilatação em uma imagem em tons de cinza (figura copiada de [6]).

Outras operações morfológicas importantes são as de abertura e fechamento. A abertura geralmente suaviza o contorno de um objeto, abre istmos e elimina saliências finas. O fechamento também possui a tendência de suavizar contornos, mas em geral, tende a fundir discontinuidades dos objetos, preenchendo lacunas, eliminando pequenos buracos. Ambas as operações são definidas a partir das definições já mencionadas de dilatação e erosão.

A abertura, denotada por $A \circ B$, de um conjunto A por um elemento estruturante B , consiste da erosão de A por B , seguida de uma dilatação do resultado por B :

$$A \circ B = (A \ominus B) \oplus B \quad (Eq. 7)$$

De maneira semelhante, o fechamento, denotado por $A \cdot B$, consiste da dilatação de A por B , seguida da erosão do resultado por B :

$$A \cdot B = (A \oplus B) \ominus B \quad (Eq. 8)$$

A Figura 11 mostra o funcionamento das operações de abertura e fechamento. Em (a) é mostrado o conjunto original A . (b) Mostra as diferentes posições de um elemento estruturante B , em forma de disco, durante o processo de erosão. Em (c) é mostrado o

resultado final da erosão: a descontinuidade entre as partes principais do conjunto A . Isso ocorreu devido à ponte ser menor que o tamanho do elemento estruturante, assim o disco não pode estar completamente contido nessa parte da imagem. (d) Mostra o processo de dilatação do conjunto erodido. Em (e) é mostrado o resultado final da abertura.

Semelhantemente, as Figuras 11 (f) até (i) mostram as etapas que consistem o fechamento de A pelo mesmo elemento estruturante. Em (f) e (g) é mostrada a operação de dilatação de A por B . Em (h) e (i) é mostrada o resultado da erosão do conjunto dilatado, tendo assim, o fechamento de A por B . Repare que a baía a esquerda do objeto teve seu tamanho reduzido. Pode-se perceber também a suavização de algumas partes da imagem.

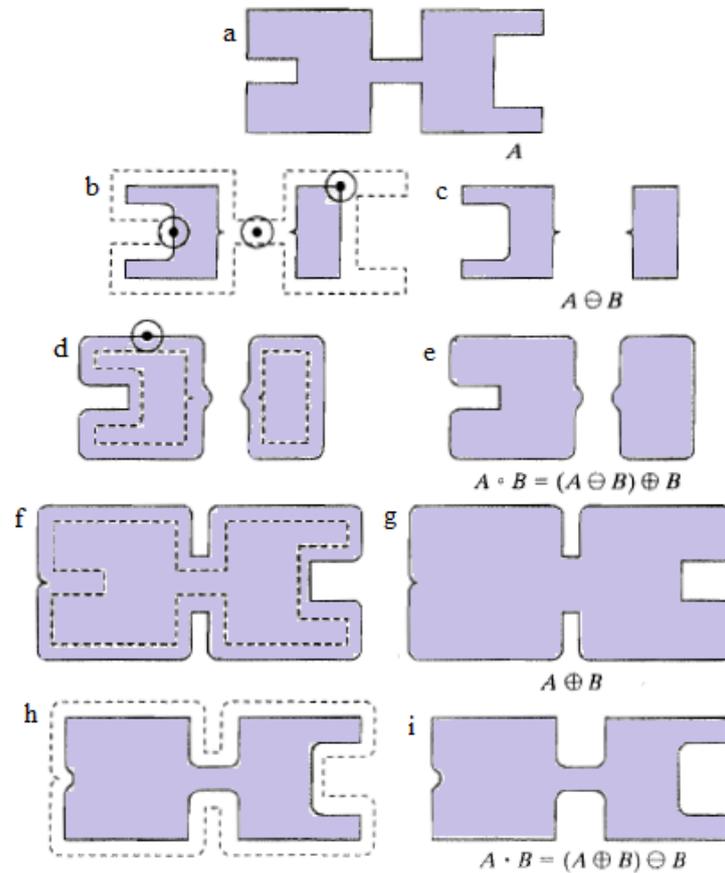


Figura 11 - Exemplo das operações de abertura e fechamento morfológicas, por um elemento estruturante em forma de disco (figura copiada de [2]).

A fronteira de um conjunto A , denotada por $\beta(A)$, também pode ser extraída pelas operações morfológicas através da dilatação ou erosão de A por um elemento estruturante B :

$$\beta(A) = (A \oplus B) - A \quad (\text{Eq. 9})$$

$$\beta(A) = A - (A \ominus B) \quad (\text{Eq. 10})$$

Ambas as Equações 9 e 10 extraem a informação de borda do conjunto A . Na primeira equação a imagem é dilatada por um elemento estruturante B , e em seguida é

subtraída do conjunto original A . Já na segunda, o conjunto original A é subtraído de sua erosão por um elemento estruturante B .

A Figura 12 ilustra o funcionamento da extração da fronteira de uma imagem em preto-e-branco, por um elemento estruturante quadrado de dimensões 3×3 . Nesse caso, foi utilizada a Equação 9, porém um resultado similar poderia ser conseguido com a Equação 10.

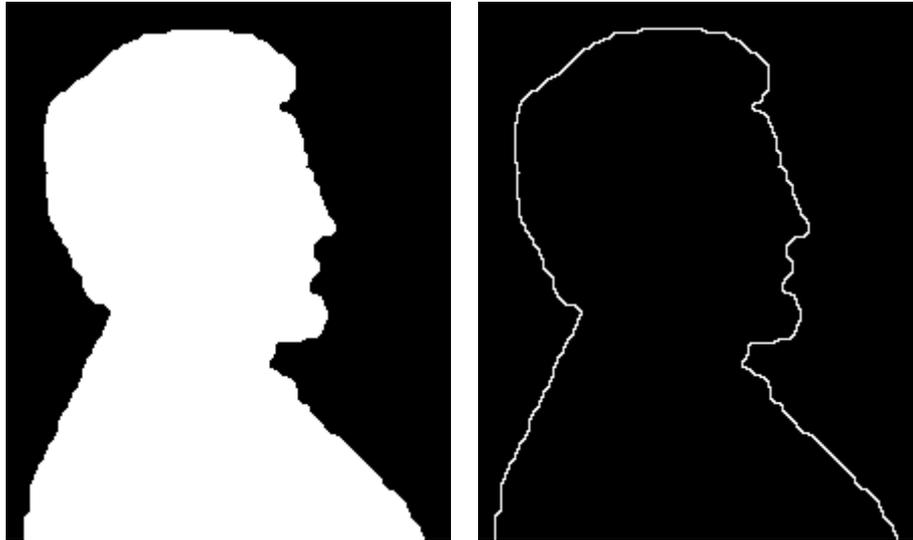


Figura 12 - Extração de fronteira de uma imagem binária, através de operações morfológicas. (a) Imagem original. (b) Fronteira extraída pela Equação 10.

Um buraco pode ser entendido como uma região de *background* contornada por um conjunto de *pixels* de *foreground* conectados. O preenchimento de buracos pode ser realizado por intermédio da operação morfológica de dilatação e as operações de interseção e complementação de conjuntos.

Seja A uma fronteira conectada por vizinhança-8 conectada que englobe uma região de *background*: o processo de preenchimento se inicia por um arranjo matricial X_0 de 0s, que possui um *pixel* semente de valor igual a 1 em cada buraco. Esse arranjo é dilatado por um elemento estruturante B , seguido da interseção desse resultado com o complemento do conjunto A . Podemos escrever esse procedimento a partir da Equação 11.

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots \quad (Eq. 11)$$

A interseção previne que os pontos dilatados ultrapassem a fronteira. O algoritmo termina quando não existir mais mudança entre duas iterações consecutivas, ou seja, $X_k = X_{k-1}$. O preenchimento dos buracos se dá pela união do conjunto da última iteração X_k com o conjunto A . A Figura 13 ilustra o funcionamento do algoritmo de preenchimento morfológico de buracos.

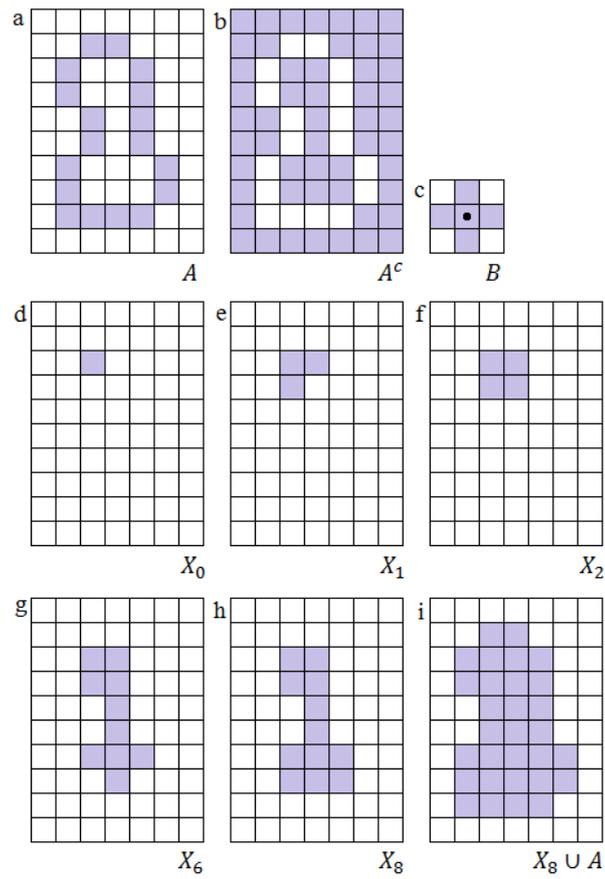


Figura 13 - Preenchimento morfológico de buracos. (a) Conjunto A (sombreado). (b) Complemento de A. (c) Elemento estruturante B. (d) Ponto semente no interior da fronteira. (e) – (h) várias iterações da Equação 11. (i) Preenchimento do buraco obtido pela união de (a) e (h) (figura copiada de [2]).

3. Segmentação de Imagens

Este capítulo apresenta um breve resumo sobre algoritmos clássicos de segmentação de imagens. Também faz uma breve descrição de um algoritmo de segmentação de imagens com baixa profundidade de campo (*low depth of field*).

3.1 Introdução

A interpretação dos dados em imagens digitais é uma tarefa complexa. Para isso, é necessário um procedimento intermediário de segmentação. Seu objetivo é extrair a partir de um conjunto de dados de entrada, estruturas com conteúdo semântico relevante para uma aplicação de interesse. É de grande importância que o procedimento de segmentação identifique corretamente propriedades básicas como a localização e forma dos objetos [1].

A segmentação subdivide a imagem em objetos ou regiões que a constitui. Em imagens de cenas reais, a segmentação pode se tornar uma tarefa muito complexa. Ruídos podem fazer com que os métodos de segmentação distorçam a forma dos objetos, prejudicando seu posterior reconhecimento. Regiões distintas poderiam ser classificadas como uma única região, ou uma região homogênea ser classificadas em mais de uma região [1].

É importante que a segmentação pare quando os objetos ou as regiões de interesse sejam detectados. A precisão da segmentação caracteriza o sucesso ou o fracasso da análise por meio de um computador. Em algumas situações, é possível fazer o controle do ambiente em que a imagem será capturada, ou utilizar de outros tipos de sensores, que realcem objetos de interesse, e diminua a seleção de detalhes que não são relevantes [2].

Os métodos mais comuns para segmentação de imagens são normalmente baseados nas propriedades básicas das imagens, buscando detectar descontinuidade ou similaridades na imagem. Na primeira categoria, a abordagem visa dividir a imagem com base nas mudanças bruscas de intensidade, caracterizadas pela presença de pontos, linhas e bordas. Na segunda categoria, os métodos estão baseados na divisão da imagem em regiões que apresentam semelhanças para um conjunto de características. A escolha das características está relacionada com o domínio da aplicação, fazendo com que se produza um conjunto de regiões homogêneas.

3.2 Detecção de descontinuidades

As três principais detecções por descontinuidades são pontos, linhas e bordas. Os *pixels* que compõem as bordas mudam abruptamente sua intensidade, e fazem parte de um conjunto conexo. As bordas são detectadas por métodos de processamento de imagens conhecidos como detectores de borda. Uma linha pode ser considerada como um segmento de uma borda, no qual as intensidades dos *pixels* de fundo são muito menores que as intensidades dos *pixels* da borda da imagem. De forma semelhante, um ponto pode ser visto como uma linha de comprimento e largura de um *pixel*.

A detecção de bordas pode ser descrita por meio de derivadas, visto que mudanças abruptas do nível de intensidade entre dois *pixels* são facilmente detectadas por essa operação. Como o modelo de uma imagem digital é uma função bidimensional $f(x, y)$,

se faz necessário analisar as bordas em várias direções. Por essa razão, são utilizadas as derivadas parciais com respeito às direções x e y . Um operador que é muito utilizado na diferenciação de imagens é o vetor gradiente, que pode ser expresso na forma matricial por:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (Eq. 12)$$

Esse vetor possui uma importante propriedade de apontar no sentido da maior taxa de variação de f no ponto (x, y) . A magnitude, denotada por $mag(\nabla f)$, do vetor gradiente corresponde ao valor da taxa de variação na direção do vetor gradiente, no qual:

$$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (Eq. 13)$$

Outra importante medida do vetor gradiente é a sua direção. O ângulo da direção do vetor ∇f é dado por:

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (Eq. 14)$$

onde o ângulo α é medido em relação ao eixo x . A direção da borda em um ponto (x, y) é sempre perpendicular à direção $\alpha(x, y)$ do vetor gradiente (Figura 14).

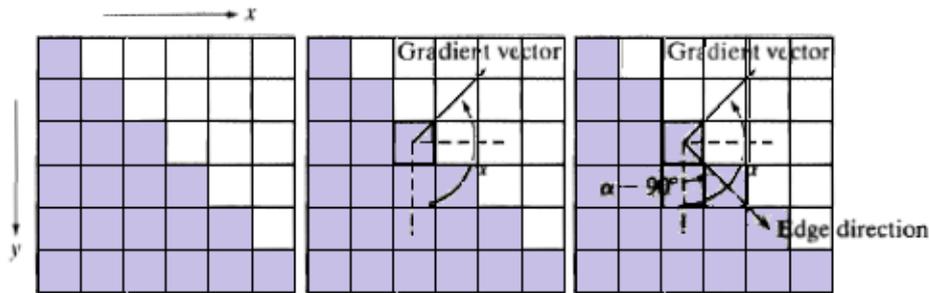


Figura 14 - Determinação da intensidade e direção da borda em um ponto através do vetor gradiente. O vetor de direção da borda é perpendicular ao vetor gradiente.

A operação do gradiente de uma imagem requer o uso de derivadas parciais. Como uma imagem digital é uma matriz formada por um conjunto discreto de *pixels*, ela não pode ser diferenciada da maneira usual. As derivadas são aproximadas por diferenças entre *pixels* de uma determinada região. Uma aproximação muito utilizada para as derivadas parciais de primeira ordem discreta é dada por:

$$\frac{\partial f}{\partial x}[x, y] \approx f[x + 1, y] - f[x, y] \quad (Eq. 15)$$

$$\frac{\partial f}{\partial y}[x, y] \approx f[x, y + 1] - f[x, y] \quad (Eq. 16)$$

As Equações 15 e 16 podem ser implementadas através da filtragem da imagem $f(x, y)$ pelo uso das máscaras 1-D mostradas na Figura 15.

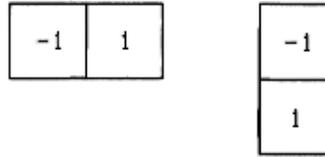


Figura 15 - Máscaras unidimensionais usadas para implementar as Equações 15 e 16.

Um operador útil quando se deseja detectar pequenos detalhes, como pequenos segmentos, pontos isolados numa imagem, ou até mesmo se um *pixel* está no lado claro ou escuro de uma imagem, é o Laplaciano. O Laplaciano é definido por uma derivada de segunda ordem:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (Eq. 17)$$

Da mesma forma como no caso do gradiente, a Equação 17 precisa ser aproximada para uma forma discreta. Como o Laplaciano é formado por derivadas de segunda ordem, podemos utilizar as Equações 15 e 16 para essa aproximação, chegando em:

$$\frac{\partial^2 f}{\partial x^2} [x, y] \approx f[x + 1, y] + f[x - 1, y] - 2f[x, y] \quad (Eq. 18)$$

$$\frac{\partial^2 f}{\partial y^2} [x, y] \approx f[x, y + 1] + f[x, y - 1] - 2f[x, y] \quad (Eq. 19)$$

Combinando as Equações 17, 18 e 19, o Laplaciano pode ser escrito como:

$$\nabla^2 f(x, y) = f[x + 1, y] + f[x - 1, y] + f[x, y + 1] + f[x, y - 1] - 4f[x, y] \quad (Eq. 20)$$

Similarmente, como foi feito com o gradiente, o Laplaciano pode ser descrito através da máscara bidimensional mostrado na Figura 16.

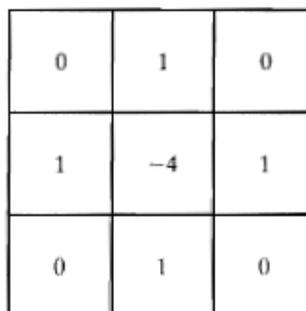


Figura 16 - Máscara bidimensional utilizada para implementar a aproximação do operador Laplaciano descrito pela Equação 20.

Apesar do operador Laplaciano detectar a grandes variações no nível de intensidade, ele é pouco utilizado para detecção de bordas. Pelo fato de utilizar derivadas de segunda

ordem, ele tende a detectar com muita facilidade mudanças bruscas, o que faz dele sensível ao ruído de uma maneira inaceitável.

Existe uma grande variedade de operadores que são utilizados na detecção de bordas. A Figura 17 mostra as máscaras dos operadores de Roberts, Prewitt e Sobel. Os operadores de Roberts foram a primeira tentativa de utilizar máscaras 2-D com uma direção de preferência diagonal. Esses operadores utilizam máscaras de tamanho 2x2, porém carregam menos informação de borda do que em máscaras simétricas em relação ao ponto central, como nas máscaras de tamanho 3x3 (Figura 18). As máscaras de Sobel são mais utilizadas na prática, pois elas apresentam melhor supressão de ruído.

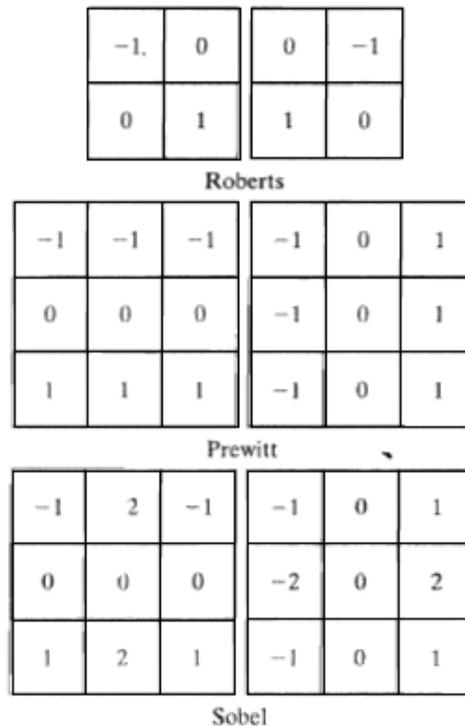


Figura 17 - Máscaras utilizadas para calcular os gradientes de Roberts, Prewitt e Sobel.

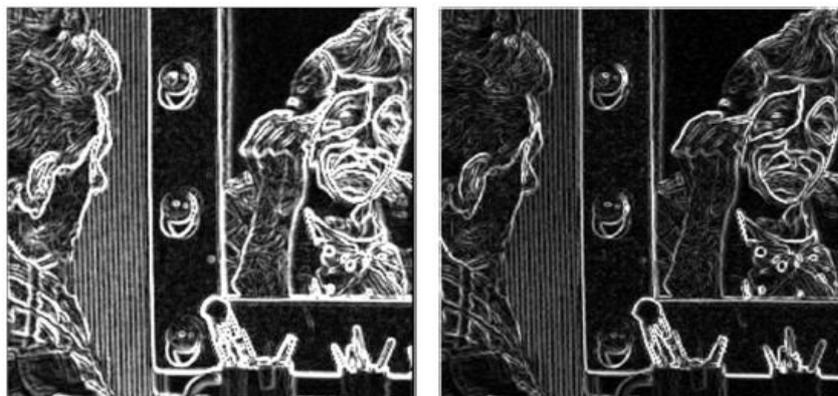


Figura 18 - Comparação das bordas detectadas pelo operador de Sobel (esquerda) e de Roberts (direita). O operador de Sobel detecta bordas mais finas, enquanto a detecção pelo operador de Roberts perdeu mais informações de borda.

A Figura 19 mostra o resultado do uso do operador gradiente de Sobel. A imagem original em (a) é filtrada pelas máscaras de Sobel mostradas na Figura 17. Em (b) é mostrado o componente G_x do gradiente, enfatizando as bordas horizontais. Em (c) é mostrado o componente G_y do gradiente, destacando as bordas na direção vertical. Finalmente, em (d) é mostrada a imagem resultante do cálculo da magnitude do vetor gradiente, a partir dos componentes G_x e G_y , como definida na Equação 13.

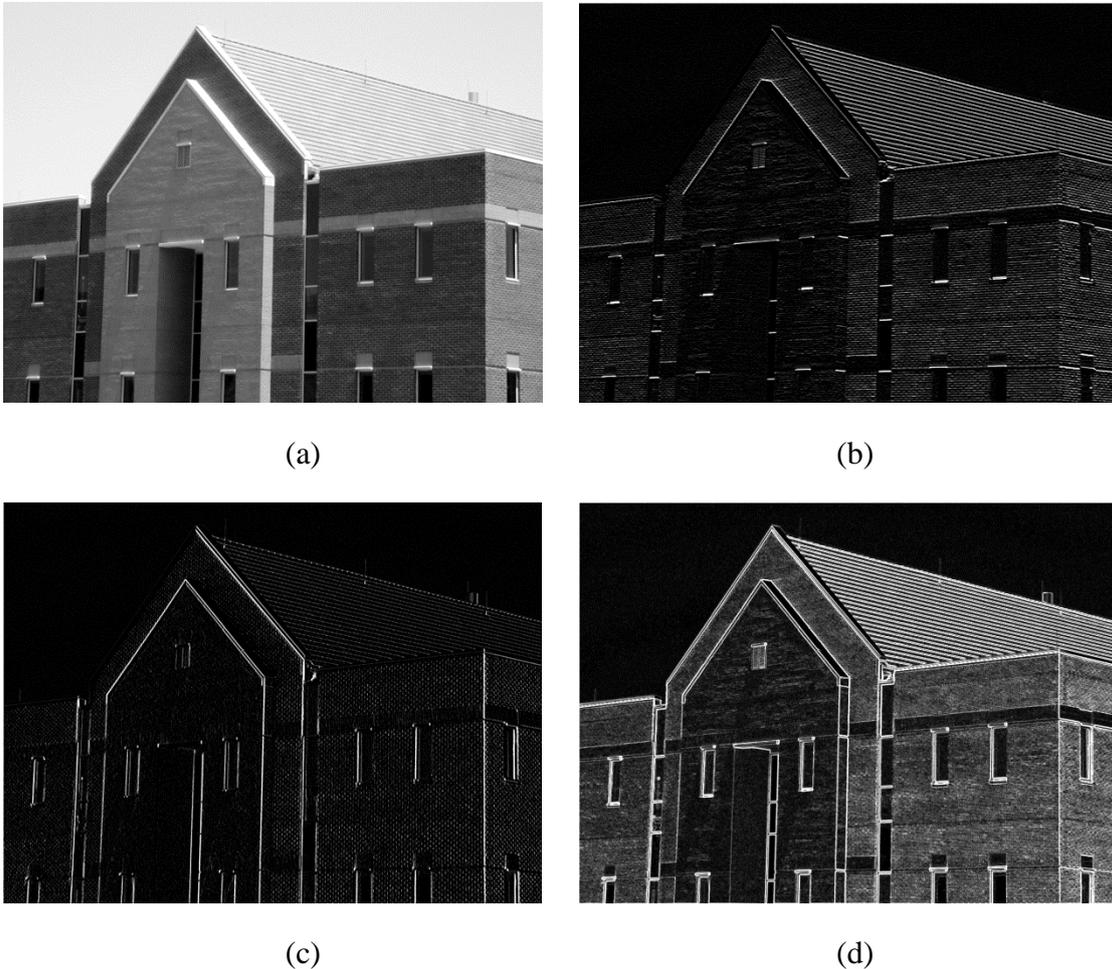


Figura 19 - Detecção das bordas utilizando o operador de Sobel. (a) Imagem original. (b) Componente G_x do gradiente. (c) Componente G_y do gradiente. (d) Magnitude do gradiente de Sobel calculado a partir de G_x e G_y .

3.3 Detecção de similaridades

Como já visto, a segmentação de imagens visa particionar uma imagem em regiões. Na seção anterior, o problema foi abordado considerando que as fronteiras entre as regiões possuem uma descontinuidade no seu nível de intensidade. Na detecção por similaridades, as técnicas de segmentação tentam encontrar as regiões de forma direta, agregando *pixels* com propriedades similares. Existem diversas propriedades que podem caracterizar uma região, tais como o nível de intensidade, cor ou textura.

O processo de segmentação de uma imagem R em n regiões R_1, R_2, \dots, R_n deve satisfazer as seguintes condições:

- (a) $\bigcup_{i=1}^n R_i = R$
- (b) R_i é uma função conexa, $i = 1, 2, \dots, n$
- (c) $R_i \cap R_j = \emptyset$ para todo i e j , $i \neq j$
- (d) $P(R_i) = VERDADEIRO$ para $i = 1, 2, \dots, n$
- (e) $P(R_i \cup R_j) = FALSO$ para $i \neq j$ e R_i adjacente a R_j .

no qual $P(R_i)$ é um propriedade lógica definida sobre o conjunto R_i e \emptyset é o conjunto vazio. Essa propriedade serve como uma medida de similaridade, com o objetivo de agrupar *pixels* em uma região.

A condição (a) indica que cada *pixel* deve pertencer a uma região da imagem segmentada. A condição (b) exige que os *pixels* pertencentes a uma região façam parte de um conjunto conexo. A condição (c) requer que as regiões sejam disjuntas. A condição (d) requer que os *pixels* pertencentes a uma mesma região possuam uma mesma propriedade em comum, tal como o nível de intensidade, por exemplo. A condição (e) indica que se duas regiões forem adjacentes, não podem possuir a mesma propriedade.

Os principais métodos de segmentação baseado em similaridades são os de crescimento de regiões (*region growing*), divisão de regiões (*quadtree partition*), divisão e fusão de regiões (*split-and-merge*) e divisor de águas (*watershed*).

3.3.1 Crescimento de regiões (*region-growing*)

Na segmentação baseada no crescimento de regiões, como o próprio nome sugere, *pixels* com propriedades similares são agrupados em regiões. O procedimento se dá a partir de um conjunto de *pixels* chamados sementes, no qual *pixels* que possuam propriedades similares são anexados às sementes, fazendo com que as regiões cresçam. A seleção das sementes pode ser feita de forma aleatória, determinística ou pelo usuário [1].

A seleção da propriedade de similaridade a não depende apenas do problema que está sendo abordado, mas também do tipo de dados presentes a imagem, como cor, textura e nível de intensidade. É comum utilizar a propriedade da diferença absoluta entre os níveis de cinza do *pixel* semente com os *pixels* de sua vizinhança, verificando se esta diferença é menor que um limiar T , ou seja

$$P(R) = \begin{cases} VERDADEIRO, & \text{se } |f(x, y) - f(r, s)| \leq T \\ FALSO, & \text{caso contrário} \end{cases}$$

onde $f(r, s)$ corresponde ao *pixel* semente e $f(x, y)$ representa os *pixels* conectados ao *pixel* semente por vizinhança-8.

A figura 20 mostra o funcionamento do método de segmentação crescimento de regiões. Os valores das células da matriz correspondem aos níveis de cinza dos *pixels*. Os dois *pixels* sementes escolhidos estão representados através de um círculo, como mostrado em (a). Como foram escolhidas apenas duas sementes, a segmentação resultará em no máximo duas regiões: R_1 associada à semente na posição (1,1) da

matriz, e R_2 associada à semente de coordenadas (5,1). Em (b) é mostrado o resultado obtido da segmentação utilizando $T = 4$. Qualquer *pixel* que satisfaça a propriedade $P(R)$ simultaneamente é atribuído à região R_1 . Para esse valor de T , a imagem foi dividida em duas regiões. Em (c) um novo valor do limiar é escolhido, atribuindo-se $T = 8$, o que resulta em uma única região.

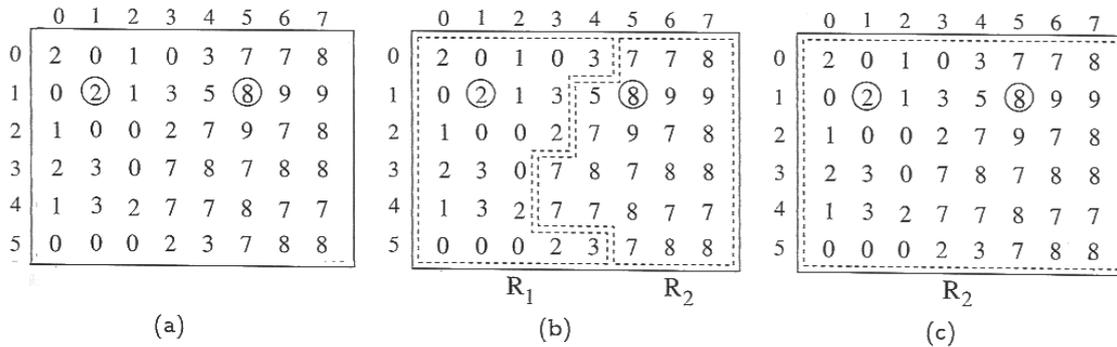


Figura 20 - Exemplo do algoritmo de segmentação baseado no crescimento de regiões. (a) Imagem original com a seleção dos *pixels* semente. (b) Segmentação utilizando diferença absoluta com um limiar menor que 4 níveis de intensidade. (c) Segmentação utilizando um limiar menor que 8 níveis de intensidade (figura copiada de [1]).

As principais dificuldades em utilizar esse método de segmentação estão na escolha das sementes que representem corretamente as regiões de interesse, assim como a escolha das propriedades para a inclusão dos *pixels* às diversas regiões, durante o processo de crescimento.

3.3.2 Divisão de regiões (*quadtree partition*)

O método de segmentação por divisão de regiões considera uma imagem por inteira como uma região. Então, é aplicada sobre essa região uma propriedade de similaridade. Caso o valor dessa propriedade não seja satisfeito para todos os *pixels* dessa região, a imagem é subdividida recursivamente até que todas as regiões satisfaçam o critério de similaridade. Uma técnica comum para subdividir a imagem em regiões homogêneas é a decomposição *quadtree*, dividindo a imagem em uma estrutura hierárquica em quadrantes. A representação dos *quadtrees* se dá por meio de uma árvore, onde cada nó possui quatro descendentes (Figura 21). A raiz da árvore representa a imagem inteira, e cada um dos quatro nós correspondem a uma subdivisão da imagem.

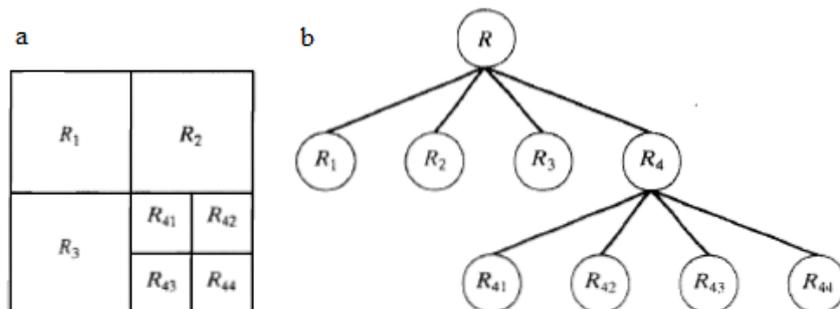


Figura 21 - (a) Imagem particionada, (b) *quadtree* correspondente (figura copiada de [2]).

A Figura 22 ilustra o funcionamento do algoritmo de segmentação por divisão de regiões. Em (a) é mostrado uma imagem formada por duas regiões, uma mais clara e outra mais escura. Em (b) é mostrada a representação da imagem em uma matriz de 8x8 *pixels* de resolução. O resultado das sucessivas divisões da imagem em regiões é mostrado em (c) e a *quadtree* correspondente às divisões é mostrada em (d). O nó raiz corresponde à imagem inteira, e os nós-folhas correspondem às regiões da matriz que não necessitaram de subdivisões adicionais.

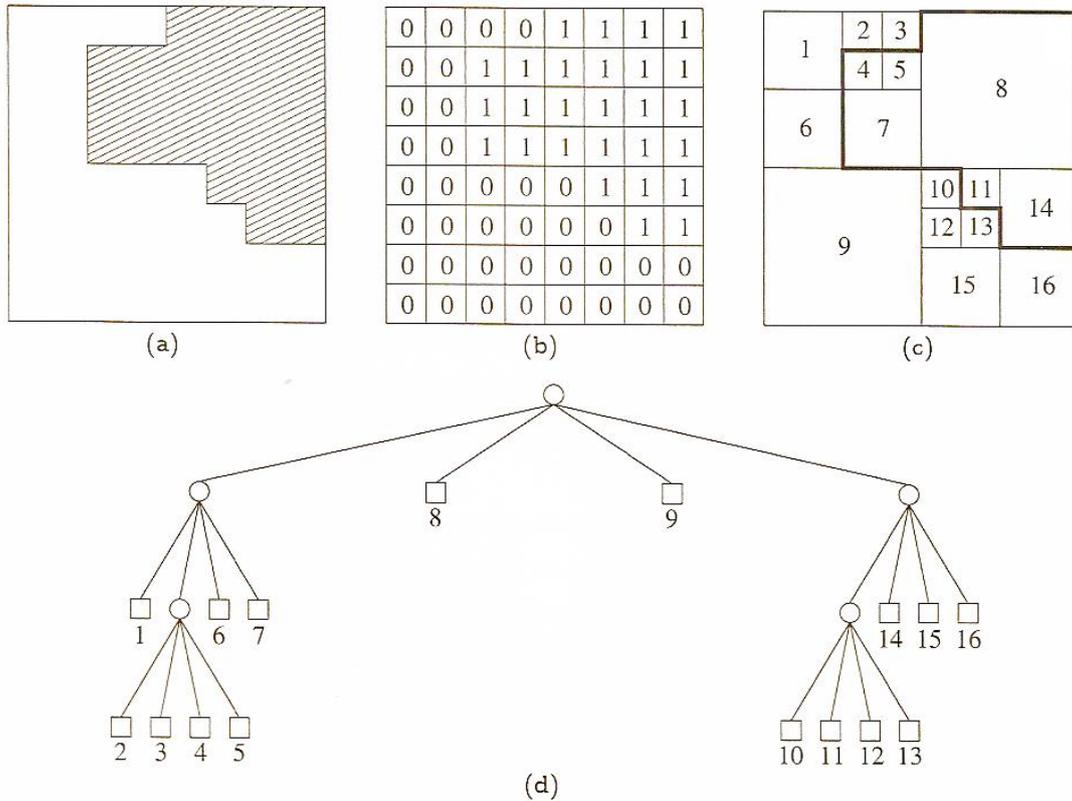


Figura 22 - Segmentação baseada na divisão de regiões. (a) Imagem original. (b) Sua representação matricial. (c) Divisões da imagem. (d) *Quadtree* correspondente (figura copiada de [1]).

3.3.3 Divisão e junção de regiões (*split-and-merge*)

A segmentação baseada na divisão e junção de regiões combina a vantagem das duas técnicas citadas anteriormente: crescimento de regiões e divisão de regiões. Esse método é útil em segmentação de imagens complexas.

Existem diversas formas de realizar o particionamento da imagem, uma possível alternativa é subdividir a imagem em várias regiões arbitrárias e disjuntas. Utilizar a junção de regiões se faz necessária, pois poderia haver regiões adjacentes apresentando propriedades similares, não satisfazendo a condição de segmentação (e) definida no início da Seção 3.3. Ou seja, duas regiões adjacentes R_i e R_j devem ser unidas se $P(R_i \cup R_j) = VERDADEIRO$. Uma abordagem muito utilizada para o particionamento da imagem é utilizar a decomposição em *quadtrees*, dividindo a imagem em blocos quadrados, como comentado na Seção 3.3.2. O algoritmo de segmentação baseado na divisão e junção de regiões pode ser descrito a seguir [2]:

1. Dividir em quatro quadrantes separados qualquer região R_i em que $P(R_i) = \text{FALSO}$.
2. Quando não for mais possível continuar dividindo, fundir quaisquer regiões adjacentes R_i e R_j tal que $P(R_i \cup R_j) = \text{VERDADEIRO}$.
3. Parar quando nenhuma fusão for mais possível.

3.3.4 Divisor de águas (*watershed*)

Na segmentação por divisor de águas, a imagem a ser segmentada pode ser entendida como uma superfície topográfica, no qual os níveis de intensidade dos *pixels* correspondem a valores de altitude dos pontos. Assim, o conjunto de *pixel* da imagem forma uma superfície contendo picos e vales.

A ideia básica do algoritmo pode ser entendida por meio de uma analogia. Suponha um orifício seja feito em cada mínimo regional da superfície, e que comece a surgir água desses orifícios a uma taxa uniforme. À medida que a água penetra, uma inundação é originada de baixo para cima nos vales dessas regiões, formando bacias hidrográficas (*watersheds*). Quando a água acumulada entre diversas bacias hidrográficas estão prestes a se juntar, uma linha de contenção é criada entre essas bacias.

O processo de inundação continua até que toda a superfície esteja a ponto de ser inundada, no qual haverá apenas as linhas de contenção delimitando cada bacia hidrográfica. Essas linhas definem os contornos dos objetos da imagem. A Figura 23 ilustra o processo de simulação da inundação para uma secção transversal da imagem.

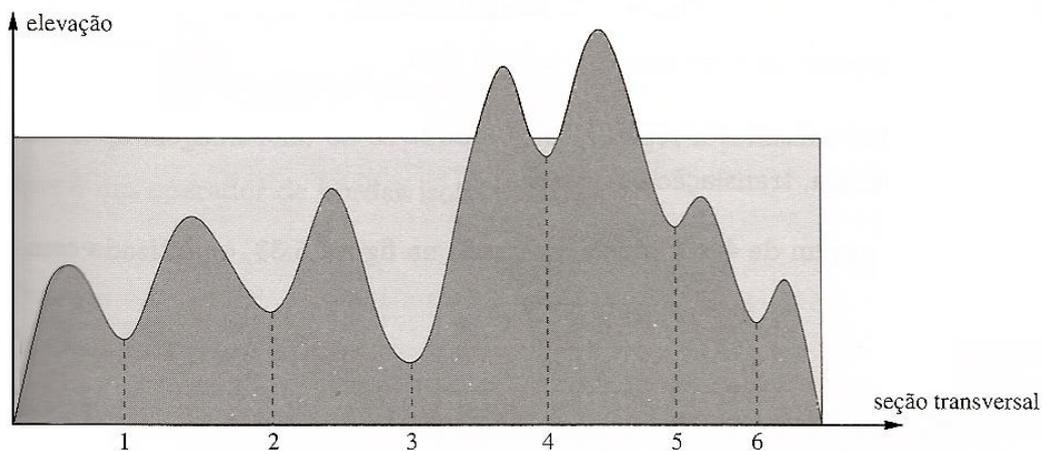


Figura 23 - Ilustração do processo de inundação da segmentação por divisor de águas (figura copiada de [1]).

A qualidade da segmentação por divisor de água pode ser afetada por diversos fatores. A inundação é sensível ao ruído, que pode ocorrer tanto na forma de valores máximos ou mínimos na imagem, prejudicando o resultado da segmentação. Irregularidades nas bordas das regiões podem permitir vazamentos de água no processo de inundação. Esses problemas resultam em uma segmentação com grande número de regiões (sobressegmentação) [1].

A Figura 24 a seguir mostra o resultado de uma segmentação baseada em Watershed para uma imagem em preto-e-branco e para uma imagem em tons de cinza. Podemos observar o problema de sobressegmentação presente na imagem em tons de cinza onde mais de 15 mil regiões foram detectadas.

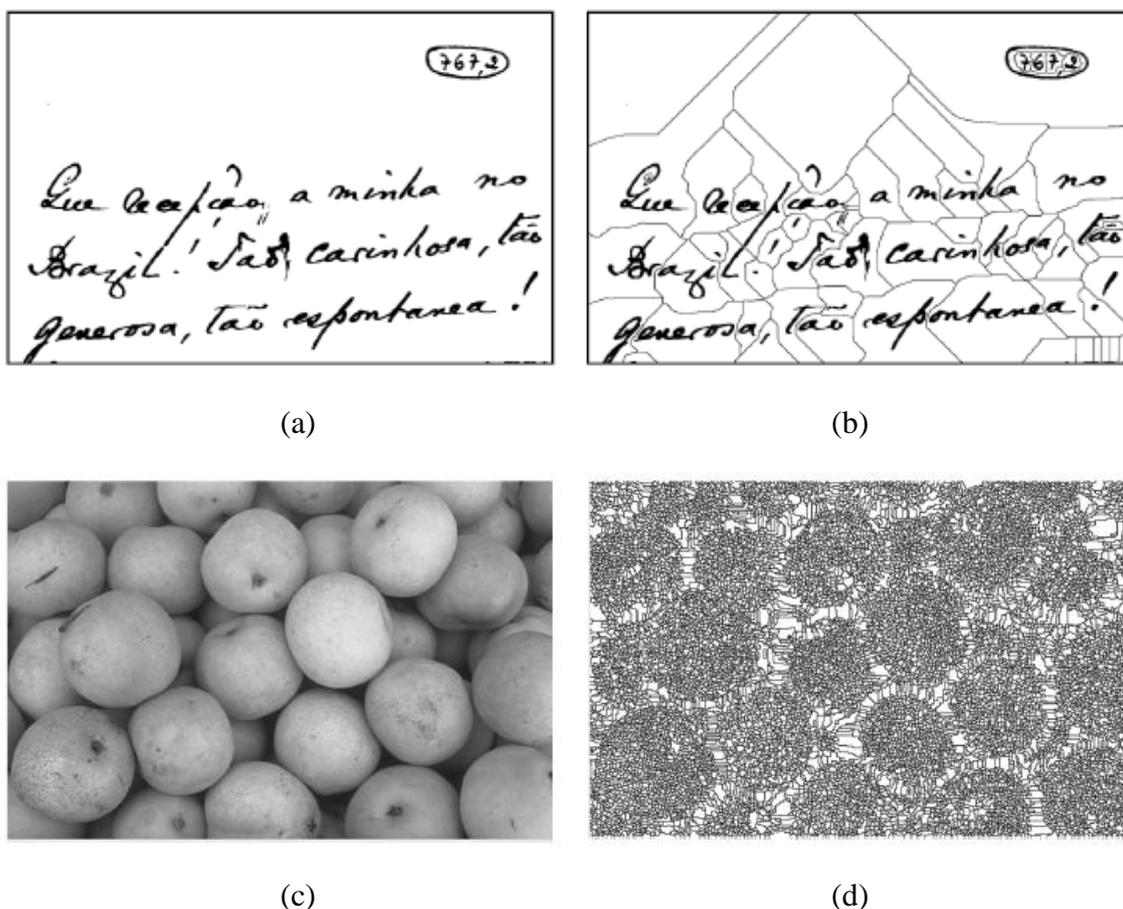


Figura 24 - Segmentação baseada em *Watershed*. (a) Imagem original em preto-e-branco. (b) Imagem segmentada. (c) Imagem original em tons de cinza. (d) Imagem sobressegmentada.

3.4 Segmentação de Imagens *Low-Depth-of-Field* Usando Filtros Morfológicos e Fusão de Regiões

O método de segmentação desenvolvido em [5], é uma abordagem de segmentação para imagens com baixa profundidade de campo (*low depth of field*). A profundidade de campo (*depth of field*) é a distância entre os objetos mais próximos e mais afastados em uma cena, que apresentam uma nitidez aceitável em uma imagem. Embora as lentes consigam focar precisamente em apenas um objeto por vez, a queda da nitidez, não ocorre de forma abrupta, mas em uma transição gradual [7]. A Figura 25 mostra uma imagem do tipo *low-depth-of-field* (*low-DOF*).

O método consiste em três etapas. Na primeira, a imagem *low-DOF* é transformada em um espaço de características apropriado, no qual a distribuição espacial dos componentes de alta frequência é representada. A escolha do espaço de características depende da aplicação.



Figura 25 - Imagem do tipo *low-depth-of-field*. Conforme os objetos se distanciam do foco, a nitidez da imagem é reduzida. O foco se encontra no plano da mão esquerda do pianista.

Nesse método, foi escolhida uma função estatística de alta ordem, onde momentos de quarta ordem foram calculados para todos os *pixels* da imagem. Assim, é obtido o mapa de características de estatística de alta ordem (*HOS map*). Na etapa seguinte, esse mapa é simplificado pela remoção de pequenos buracos escuros e manchas claras usando um filtro morfológico de reconstrução. Finalmente, o objeto de interesse é extraído, aplicando a fusão de regiões na imagem simplificada e uma limiarização.

A Figura 26 mostra a sequência de etapas do algoritmo de [5]. Em (a) é mostrada uma imagem *low-DOF*. Em (b) é mostrado o mapa de características de estatística de alta ordem (*HOS map*). Em (c) é mostrada a imagem simplificada pela filtragem morfológica de reconstrução. Em (d) é mostrada a imagem resultante da junção de regiões, enquanto em (e) é mostrado o resultado final da segmentação.



Figura 26 - Algoritmo de segmentação de imagens (*low-DOF*) de Kim: (a) imagem original, (b) mapa de características, (c) filtragem morfológica, (d) junção de regiões e (e) resultado final da segmentação.

Segundo [5], o algoritmo proposto complementa a limitação de métodos que se baseiam somente nos detalhes finos dos objetos de interesse, pelo uso de filtros morfológicos. Esses filtros também permitem uma perfeita preservação de informação do contorno. Comparado com os métodos anteriores, o método proposto produz resultados de segmentação mais precisos, suportando processamento rápido.

4. Segmentação de Imagens Utilizando Elementos de Morfologia Matemática

Este capítulo apresenta o método de segmentação proposto neste trabalho, dividindo-se em duas partes. Na primeira, é descrito o algoritmo de detecção de bordas por meio de operações morfológicas, utilizando múltiplos elementos estruturantes [4]. Na segunda, é descrita as etapas seguintes do método de segmentação de imagens, baseado na detecção de bordas.

4.1 Detecção de Bordas Utilizando Múltiplos Elementos Estruturantes

Este método, descrito em [4], detecta as bordas de uma imagem através de operações morfológicas de erosão e dilatação. Como mencionado no final do Capítulo 2, bordas de uma imagem podem ser extraídas através das Equações 9 ou 10, com o uso de um elemento estruturante adequado.

Na maioria das operações morfológicas, a escolha do elemento estruturante depende do problema em questão. Seu tamanho e formato determinam o resultado final da detecção de bordas. Normalmente um único elemento estruturante simétrico é utilizado. Segundo [4], utilizar apenas um elemento estruturante dificulta a detecção de características complexas de bordas, pois só são sensíveis às bordas que possuem a mesma direção.

Diante dessas limitações, os autores em [4] propuseram um método de detecção de bordas baseado em operações morfológicas, que utilizasse múltiplos elementos estruturantes, para superar os problemas já mencionados. A ideia básica da morfologia por múltiplos elementos estruturantes é construir diferentes elementos numa mesma janela quadrada e que contenha a maioria das direções possíveis nessa janela.

Seja $F(x, y)$, uma imagem digital, o conjunto de elementos estruturantes de tamanho $(2N + 1) \times (2N + 1)$ pode ser denotado por:

$$B_i = \{F(x + m_0, y + n_0), \theta_i = i \times \alpha \mid -N \leq m_0, n_0 \leq N\} \quad (Eq. 21)$$

onde $i = 0, 1, \dots, 4N - 1$, $\alpha = 180^\circ/4N$ e θ_i é a direção do ângulo do elemento estruturante. Foram escolhidos a Equação 10 para a extração da borda e $N = 2$, resultando na criação de oito elementos estruturantes de tamanho de janela 5x5, com as seguintes direções dos ângulos: 0° , 22.5° , 45° , 67.5° , 90° , 112.5° , 135° e 157.5° . A Figura 27 mostra todos esses elementos estruturantes, onde as células sombreadas representam seus componentes.

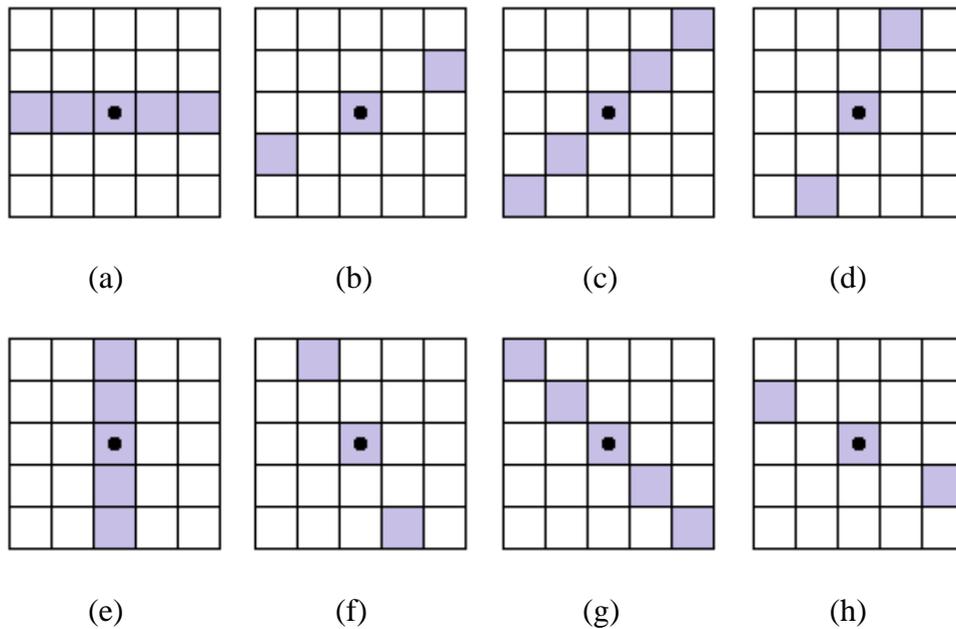


Figura 27 - Conjunto de oito diferentes elementos estruturantes em uma janela 5x5. (a) B_0 com $\theta_0 = 0^\circ$. (b) B_1 com $\theta_1 = 22.5^\circ$. (c) B_2 com $\theta_2 = 45^\circ$. (d) B_3 com $\theta_3 = 67.5^\circ$. (e) B_4 com $\theta_4 = 90^\circ$. (f) B_5 com $\theta_5 = 112.5^\circ$. (g) B_6 com $\theta_6 = 135^\circ$. (h) B_7 com $\theta_7 = 157.5^\circ$.

Em seguida, para cada elemento estruturante criado, é extraída a borda da imagem utilizando a Equação 9 ou a Equação 10. Como resultado, são obtidas oito detecções de borda. A fim de calcular a detecção de borda final, é feita uma média aritmética entre todas as oito bordas.

$$\beta(F) = \frac{1}{M} \sum_{i=0}^{M-1} \beta_i(F) \quad (Eq. 22)$$

onde M é o número de elementos estruturantes criados.

Finalmente a imagem com a borda resultante é binarizada, a fim de obtermos uma imagem em preto-e-branco. Para calcular o valor do *threshold*, foi utilizado o método de limiarização global de Otsu [1].

A Figura 28 mostra o resultado da detecção das bordas de uma imagem em tons de cinza obtido por esse algoritmo. Nesta imagem ficam claros alguns problemas desse método. A primeira é que esse algoritmo tende a engrossar a espessura das bordas geradas. A espessura das bordas está fortemente relacionada com o tamanho e formato dos elementos estruturantes. Quanto maior o tamanho da janela do elemento estruturante, maior a quantidade de *pixels* erodidos ou dilatados. Por outro lado, utilizar uma janela de tamanho menor tem como consequência um menor número de elementos estruturantes, um menor número de direções, resultando numa menor quantidade de *pixels* sendo dilatados ou erodidos. Outro problema é que este método é sensível a ruídos e a informações de textura presentes na imagem.



Figura 28 - (esquerda) Imagem original. (direita) Detecção das bordas geradas pelo algoritmo. As informações sobre as texturas dos objetos da imagem foram também consideradas.

4.2 Segmentação de Imagens *Low-Depth-of-Field*

Para obter melhores resultados na segmentação, é importante detectar corretamente as bordas da imagem. Na seção anterior foi visto que o algoritmo de detecção de bordas apresenta alguns problemas, com a sensibilidade ao ruído e a informações de textura. As bordas de uma imagem normalmente são compostas de um grande número de componentes conectados. O ruído, por sua vez, normalmente é caracterizado por pequenos conjuntos de *pixels* conectados.

Apesar dos problemas comentados, o algoritmo de detecção de bordas apresentado possui indícios de que se comporta melhor em imagens do tipo *Low-Depth-of-Field*. A característica principal é que nessas imagens o foco está na região de interesse, já o *background* ou regiões que não são de interesse normalmente estão desfocados. Essa característica tende a fazer com que ruídos presentes no fundo da imagem, ou regiões de textura que não são de interesse sejam atenuadas. Como consequência, as bordas são detectadas mais precisamente. A Figura 29 (a) mostra um exemplo desse tipo de imagem.

Como pode ser observado na Figura 29 (b), o método de detecção de bordas morfológico com múltiplos elementos estruturantes gera ruídos na imagem. Observe que o ruído normalmente pode ser caracterizado como pequenas regiões de *pixels* conectados. A fim de atenuar a quantidade informação não importante, é feita uma filtragem na imagem. O filtro faz uma varredura na imagem, separando em N regiões os

pixels definidos por uma vizinhança-8 conectada. Depois, é feita uma contagem nos *pixels* de cada região, ordenando-as de forma decrescente. As n maiores regiões conectadas, onde $n = 1, 2, \dots, N$, permanecem na imagem, removendo as regiões que possuem um baixo número de componentes conectados. A Figura 29 (c) mostra a imagem filtrada.

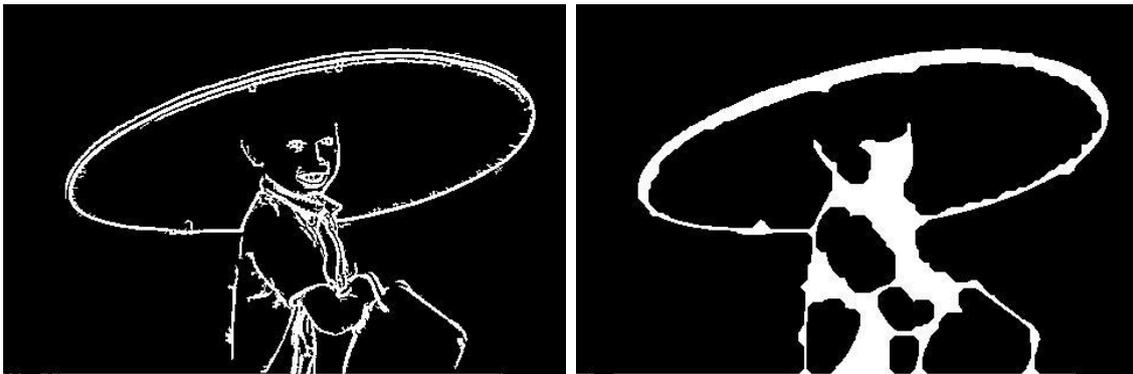
Outro problema na detecção de bordas, é que elas podem produzir bordas desconexas. Além disso, o processo de filtragem anterior pode remover algumas regiões de pequenas bordas desconexas. As bordas são importantes, pois definem o contorno do objeto de interesse. Diante desses problemas, a etapa seguinte do algoritmo é realizar a operação de morfológica de fechamento para conectar esses componentes desconexos. É utilizado um elemento estruturante em forma de disco, cujas dimensões devem ser definido para cada aplicação, pois essa escolha depende da imagem em questão. A Figura 29 (d) mostra a imagem após a operação de fechamento.

Mesmo após a operação de fechamento é possível que buracos ainda permaneçam na imagem. Esses buracos são preenchidos através da operação morfológica descrita na Equação 11, obtendo assim o *foreground* da imagem. A Figura 29 (e) mostra os buracos preenchidos. Os *pixels* brancos correspondem à região de interesse (*foreground*). Finalmente é feita a composição do *foreground* com a imagem original, como mostrado na Figura 29 (f).



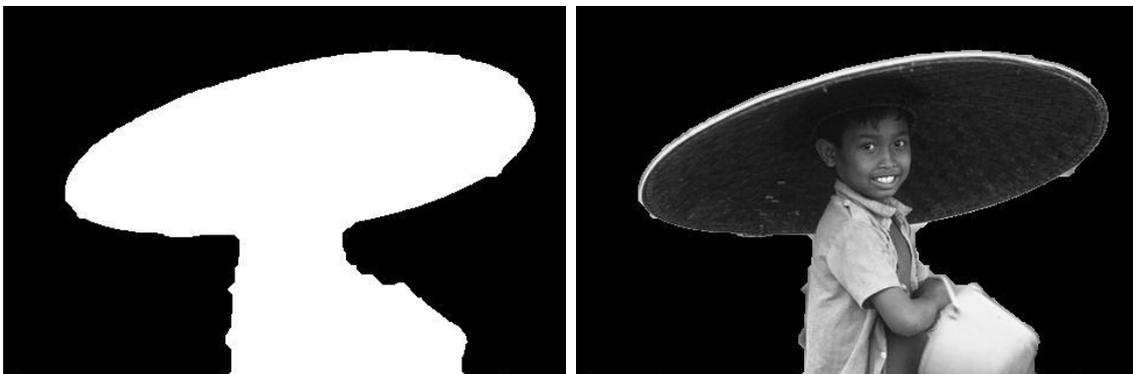
(a)

(b)



(c)

(d)



(e)

(f)

Figura 29 - Segmentação da imagem pelo algoritmo proposto em uma imagem *low depth of Field*: (a) Imagem original, (b) detecção da borda utilizando múltiplos elementos estruturantes, (c) remoção de ruídos em função da quantidade de *pixels* conectados, (d) fechamento de bordas desconexas, utilizando a operação morfológica de fechamento, (e) *foreground* da imagem obtido a partir da operação morfológica de preenchimento dos buracos, (f) Composição com a imagem original.

5. Experimentos

Este capítulo tem como objetivo apresentar e discutir os resultados dos experimentos realizados com o algoritmo de segmentação proposto, utilizando imagens *low depth of field*. É feita uma comparação entre o algoritmo proposto neste trabalho e o algoritmo proposto em [5]. Também é feita uma

5.1 Experimento 1

O objetivo do desse experimento é testar o comportamento do algoritmo de segmentação proposto por este trabalho. Para avaliá-lo foram coletadas imagens *low depth of field* na internet. Foram realizados experimentos com 25 imagens desse tipo. Nesse capítulo, porém, é mostrado apenas o resultado de sete imagens, no qual duas delas são comparadas com o algoritmo de [5]. O resultado obtido a partir de outras imagens pode ser observado na Seção de Anexos deste trabalho.

5.1.1 Metodologia

As bordas foram extraídas utilizando o operador morfológico de extração de fronteiras descrito pela Equação 10. Foi escolhido um valor de $N = 2$ para a construção dos elementos estruturantes, resultando assim em oito elementos estruturantes com janela de tamanho 5×5 .

O algoritmo proposto, como comentado no capítulo anterior, não é totalmente automatizado. Dois parâmetros precisam ser definidos, sendo fortemente correlacionados com a imagem em questão:

- Quantidade dos n maiores componentes conectados, durante a etapa de filtragem da imagem.
- Tamanho do raio elemento estruturante (disco), na operação de fechamento da imagem.

Os resultados de segmentação apresentados correspondem às melhores configurações obtidas, variando esses dois parâmetros. Os valores desses parâmetros, para cada imagem, são mostrados na Tabela 1.

Figura	Numero de comp. conectados	Raio do Elem. Estruturante (em <i>pixels</i>)
30	25	20
31	5	15
32	1	5
33	50	15
34	280	15
35	20	15
36	5	14

Tabela 1 - Valores utilizados nos parâmetros do algoritmo de segmentação proposto.

5.1.2 Resultados

As Figuras 30 a 36 mostram os resultados obtidos da segmentação utilizando o algoritmo proposto. Nas Figuras 35 e 36 é feita uma comparação entre os resultados obtidos entre o algoritmo proposto neste trabalho e o algoritmo de segmentação proposto por [5].

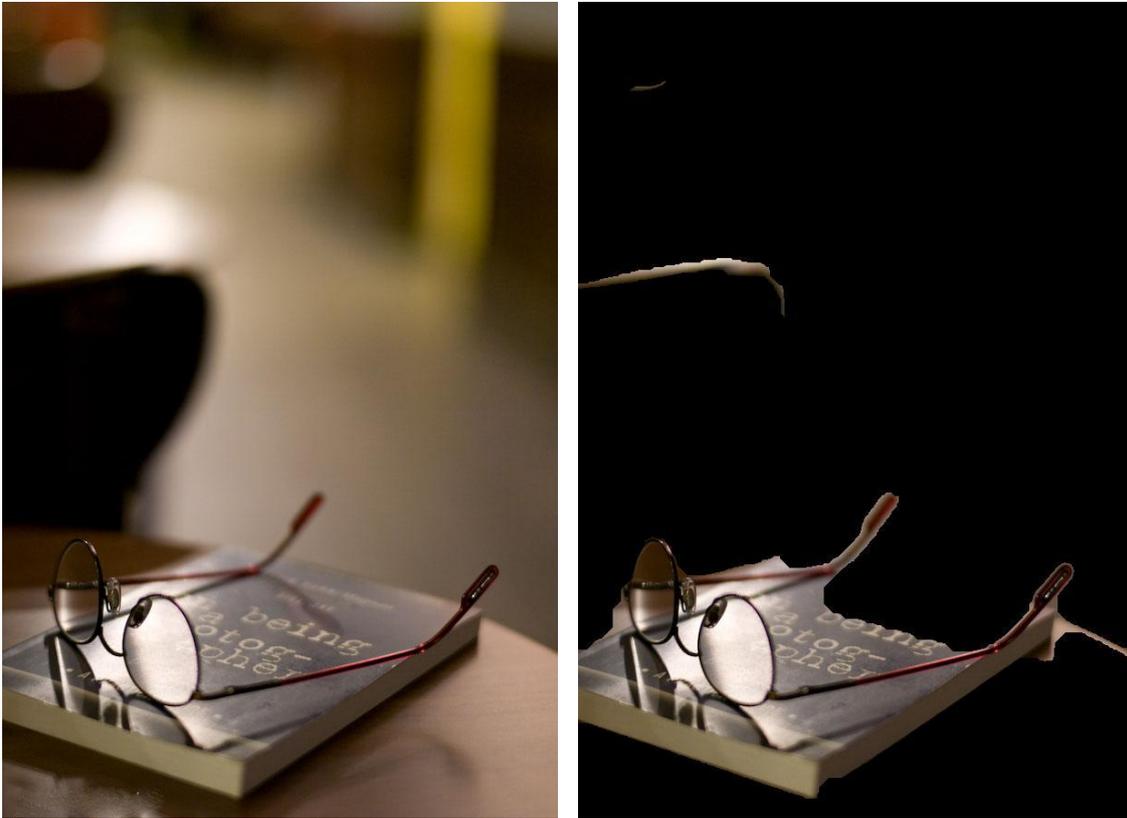


Figura 30 - Segmentação da imagem utilizando o algoritmo proposto em imagens *low depth of field*: (esquerda) imagem original e (direita) imagem segmentada.

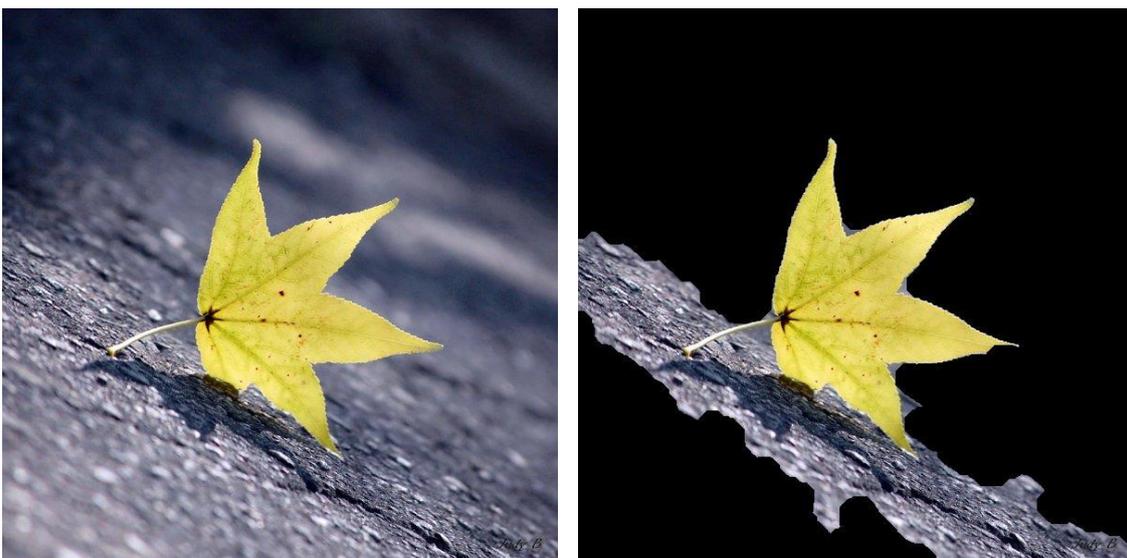


Figura 31- Segmentação da imagem utilizando o algoritmo proposto em imagens *low depth of field*: (esquerda) imagem original e (direita) imagem segmentada.



Figura 32 - Segmentação da imagem utilizando o algoritmo proposto em imagens *low depth of field*: (esquerda) imagem original e (direita) imagem segmentada.



Figura 33 - Segmentação da imagem utilizando o algoritmo proposto em imagens *low depth of field*: (esquerda) imagem original e (direita) imagem segmentada.



Figura 34 - Segmentação da imagem utilizando o algoritmo proposto em imagens *low depth of field*: (esquerda) imagem original e (direita) imagem segmentada.



(a)



(b)



(c)

Figura 35 - Comparação entre os métodos de segmentação de imagens: (a) imagem original, (b) resultado do algoritmo proposto (c) resultado do algoritmo de [5].



(a)



(b)



(c)

Figura 36 - Comparação entre os métodos de segmentação de imagens: (a) imagem original, (b) resultado do algoritmo proposto (c) resultado do algoritmo de [5].

5.1.3 Discussão

De acordo com os experimentos realizados, podemos observar que o algoritmo proposto consegue segmentar regiões de interesse em uma imagem *low depth of field*.

Na Figura 30, a região de interesse se encontrava no plano dos óculos e do livro. Podemos ainda observar uma falha na segmentação: foi segmentada uma pequena parte do braço da poltrona que está desfocada. Isso aconteceu na etapa da detecção de bordas, no qual o algoritmo é sensível às mudanças bruscas nos níveis de intensidade. Como essa a região detectada era maior que algumas regiões de interesse, ela não foi removida na etapa de filtragem.

A Figura 31 mostra um bom resultado, segmentando apenas a região focada na imagem. Perceba que a região de textura que não está focada não é segmentada como uma região de interesse, devido à suavização da mesma.

Na Figura 32 é mostrado outro bom resultado da segmentação. A peça branca do xadrez foi segmentada isoladamente, mesmo com a presença no meio da peça preta no em um fundo claro. A etapa de filtragem dos componentes conexos foi fundamental para conseguir esse resultado. Note que de acordo com a Tabela 1, apenas um componente conexo foi escolhido, ou seja, a peça branca, que possuía uma maior quantidade de componentes conectados.

Na Figura 33 é mostrado outro resultado da segmentação. Apesar de selecionar corretamente a região de interesse, perceba que as regiões internas que compõem as rodas e o triângulo da bicicleta foram segmentadas. Isso ocorre devido ao algoritmo utilizar a operação de preenchimento de buracos, que neste caso, tratou essas regiões como buracos.

Na Figura 34, foram detectadas muitas regiões contendo uma forte informação de textura na etapa de detecção de bordas. Essas regiões formam grandes conjuntos de componentes conexos, que fazem parte da região de interesse. Por isso, como se pode notar na Tabela 1, foram escolhidas 280 regiões conexas. Perceba que devido a essa complexidade na imagem, com uma grande quantidade de regiões de textura e uma grande mudança no nível de intensidade no *background* da imagem, a segmentação dessa imagem não obteve um bom resultado.

Na Figura 35 é feita uma comparação entre o algoritmo proposto neste trabalho e o algoritmo proposto por [5] em imagens *low depth of field*. Utilizando essa imagem, o algoritmo de [5] se mostrou superior detectando precisamente a região de interesse. O outro mostrou os problemas que já comentamos anteriormente: como ele está baseado na informação de borda, mudanças bruscas nos níveis de intensidade são detectadas. Podemos notar isso, na região das janelas do prédio que foram detectadas. A filtragem não foi capaz de remover esses componentes da imagem.

Na Figura 36 é feita outra comparação entre esses dois algoritmos, porém obtendo um resultado inverso: o algoritmo proposto se mostrou superior ao de [5]. Enquanto o primeiro detectou precisamente o contorno do menino, e posteriormente fez a segmentação, o segundo possui falhas: não preenchendo algumas regiões que fazem parte da região de interesse. Observe que as regiões internas do chapéu, do ombro e da bolsa do menino, não foram segmentadas.

Os resultados mostrados na seção anterior confirmam que as etapas de detecção de bordas e o da filtragem são as mais importantes no método de segmentação proposto. Detectores de bordas que extraíam pouca informação de textura e sendo pouco sensível ao ruído, ajudam na qualidade da segmentação. É importante também que a região de interesse possua uma maior quantidade de componentes conectados que as demais regiões e que elas não estejam conectadas à textura ou ruídos, facilitando a qualidade da filtragem da imagem.

Diante de outros experimentos realizados, pode-se verificar que o principal problema do algoritmo proposto por [5] é que em muitas situações ele deixa buracos na região de interesse. O resultado de outros experimentos com esse algoritmo pode ser observados na Seção Anexos deste trabalho.

5.2 Experimento 2

Neste experimento, foi gerada uma imagem padrão ouro da imagem do menino apresentada na Figura 36 (a). Essa imagem é apresentada na Figura 37. Foi selecionada uma imagem de *background* com poucos elementos de textura e suavizada com um filtro gaussiano com diferentes parâmetros a fim de simular a perda de foco de uma imagem *low depth of field*.

Uma composição foi feita entre a imagem padrão ouro e esses *backgrounds* suavizados através de um filtro passa-baixa Gaussiano com máscaras 3x3, 5x5, 7x7, 9x9 e 15x15 (quanto maior as dimensões da máscara, maior a abertura do filtro e maior o embaçamento da imagem), gerando as imagens mostradas na Figura 38. Em seguida, cada uma dessas imagens foi segmentada pelo algoritmo proposto. Os resultados da segmentação podem ser vistos na Figura 39.



Figura 37 – Imagem padrão ouro usada para composição e análise da robustez do algoritmo de segmentação proposto



Figura 38 – Composições geradas entre o padrão ouro e um *background*: (a) sem embaçamento, e com embaçamento gerado usando um filtro Gaussiano (b) 3x3, (c) 5x5, (d) 7x7, (e) 9x9 e (f) 15x15.

Com a imagem padrão ouro como referência, pudemos fazer uma análise quantitativa do experimento. Calculamos os valores de verdadeiro positivo (VP), verdadeiro negativo (VN), falso positivo (FP) e falso negativo (FN), considerando como positivo os tons de *foreground* e negativo os tons de *background*. Assim, por exemplo, pontos com cor diferente de preto na imagem *ground truth* e também tons diferentes de preto na imagem sendo analisada geram valores de verdadeiro positivo. Os resultados para cada uma das imagens de composição podem ser vistos na Tabela 2.

Figura	Filtro Gaussiano	VP	VN	FP	FN
38 (a)	-	0,9994	0,7904	0,3042	0,0004
38 (b)	3x3	0,9996	0,8182	0,264	0,0003
38 (c)	5x5	0,9996	0,9152	0,1231	0,0003
38 (d)	7x7	0,9996	0,9047	0,1384	0,0003
38 (e)	9x9	0,9996	0,9496	0,0731	0,0003
38 (f)	15x15	0,9999	0,9845	0,0226	0,0004

Tabela 2 - Avaliação quantitativa da segmentação aplicada sobre composições com diferentes níveis de embaçamento do *background*. Valores apresentados normalizados.

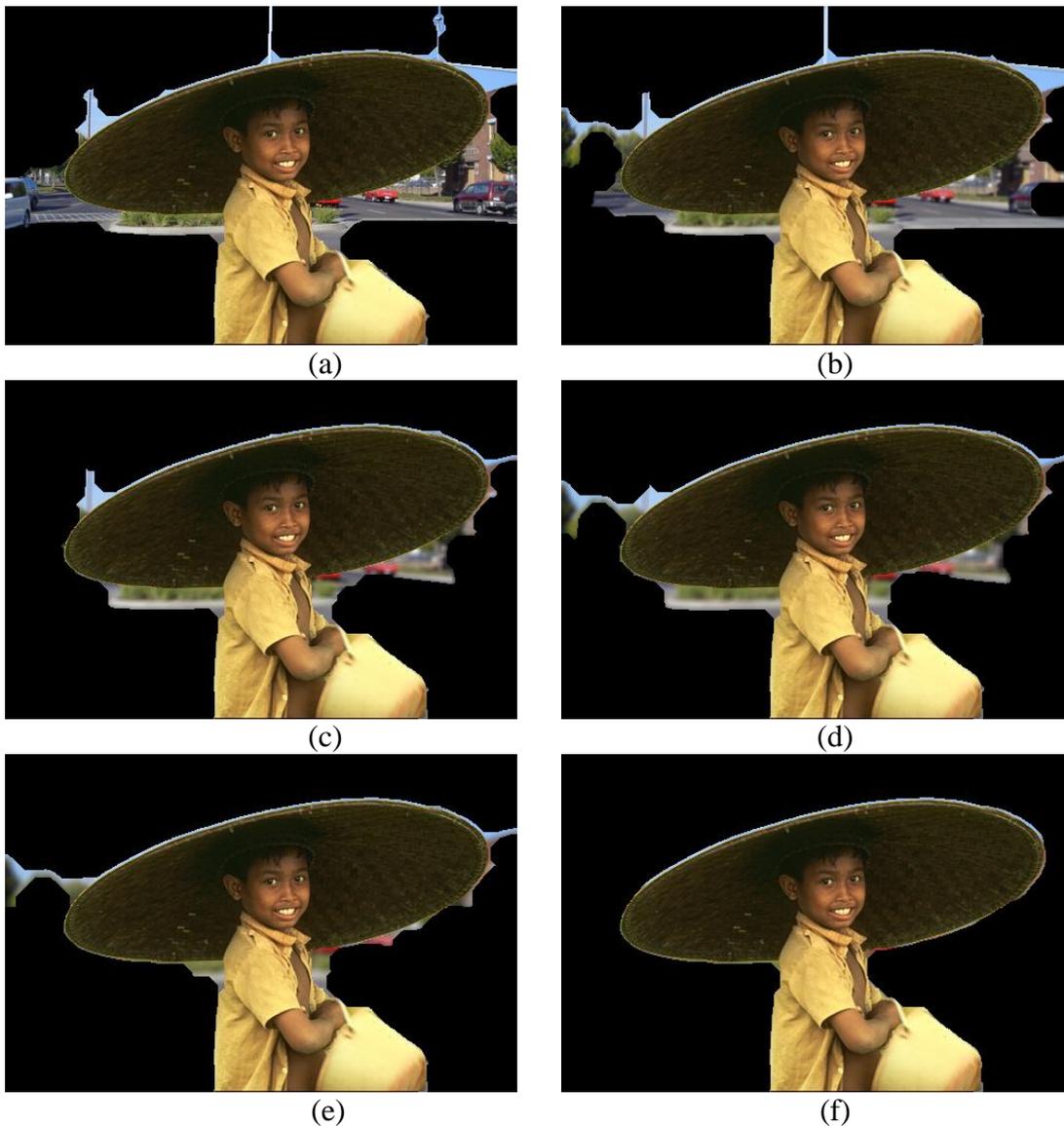


Figura 39 – Segmentação das imagens da Figura 38: segmentação da composição da imagem padrão ouro e (a) imagem de fundo sem embaçamento, e imagem de fundo com embaçamento gerado por um filtro Gaussiano (b) 3x3, (c) 5x5, (d) 7x7, (e) 9x9 e (f) 15x15.

Da Tabela 2, podemos concluir que todos os algoritmos segmentaram o menino de forma quase perfeita (valores de VP próximos a 1), mas houve erros na detecção do *background* (VN) que foram diminuindo à medida que o embaçamento aumenta. Isso é perceptível na visualização das imagens segmentadas. O melhor caso foi com o maior embaçamento conseguido com o filtro 15x15. Por consequência, os erros de detecção do *foreground* foram diminuindo da imagem sem composição até a imagem com maior suavização do *background*. Os erros de detecção do *background* (FN) foram próximos para todas as imagens.

6. Conclusão

Este trabalho apresentou um estudo sobre alguns métodos de segmentação de imagens e propôs um algoritmo de segmentação baseado em detecção de bordas e utilizando operações morfológicas. Foi apresentado também uma oportunidade de aplicação do algoritmo proposto por [4], na utilização do mesmo em imagens *low depth of field*.

A etapa de detecção de bordas é fundamental para a segmentação. É a partir dela que podemos extrair os objetos de interesse para a segmentação. Foi mostrado que o algoritmo de [4] não se comportava bem em imagens que possuem uma textura evidente, detectando-as indesejavelmente. Outro problema comentado foi a sensibilidade do mesmo em relação ao ruído.

Os problemas citados acima são minimizados em imagens *low depth of field*. Sua característica de focar apenas nos objetos de interesse, no qual as demais regiões não estão nítidas, ajudam ao algoritmo [4] alcançar maior precisão na detecção das bordas.

Diante disso, o algoritmo proposto neste trabalho utiliza a detecção de bordas de [4] como primeira etapa. Como segunda etapa é realizada uma filtragem na imagem, removendo regiões de *pixels* com poucos componentes conexos. Em seguida, é realizada a operação morfológica de fechamento. Posteriormente, possíveis buracos remanescentes são fechados. Finalmente é feita uma composição da imagem original com as imagens resultante dessas operações.

O algoritmo de segmentação possui algumas limitações. Como ele é baseado em bordas, variações bruscas nos níveis de intensidade são detectadas facilmente, o que pode gerar ruído, outro problema está na detecção de regiões de textura. Um importante passo pra minimizar esses problemas é a etapa de filtragem, podendo remover componentes indesejados. Outro problema identificado, é que o algoritmo pode preencher um buraco pertencente a uma região que não é de interesse, que não foi filtrada, conforme discutido na Seção 5.3.

Os experimentos mostram que o algoritmo proposto neste trabalho consegue segmentar regiões de interesse em uma imagem *low depth of field*. O método proposto, por sua vez, não é automático. É necessário definir dois parâmetros que estão fortemente correlacionados com a imagem em questão. Isso é uma desvantagem desse método, pois os algoritmos devem ser os mais automáticos possíveis. Por outro lado, conseguimos bons resultados de segmentação para o tipo de imagem abordada, conforme pode ser visto no Capítulo de Experimentos ou na Seção de Anexos.

Este trabalho ainda comparou os resultados obtidos do algoritmo proposto com o algoritmo proposto por [5] para segmentação de imagens *low depth of field*. Diante dos experimentos realizados, o algoritmo de [5] se mostra superior, em algumas situações, ao apresentado neste trabalho. Entretanto, em outras ocasiões, o método de [5] deixa buracos em regiões de interesse, no qual o algoritmo apresentado neste trabalho consegue fazer uma segmentação mais precisa. No segundo experimento mostramos que o erro da detecção do *background* da imagem é reduzido à medida que a desfocagem aumenta.

6.1 Trabalhos Futuros

Uma possível melhoria para os já comentados problemas do algoritmo de detecção de bordas de [4] seria utilizar sucessivas filtragens de mediana, após a detecção de borda, para remover os ruídos ou atenuar regiões de textura da imagem. Uma melhoria nesta etapa seria de grande importância para comportamento do restante do algoritmo.

Conforme comentado na Seção 4.2 e na Conclusão, uma desvantagem do método de segmentação proposto neste trabalho é que ele não é automático, ficando a critério do usuário o ajuste de dois parâmetros. Uma melhoria seria eliminar esta escolha do usuário, fazendo a definição dos parâmetros de forma automática.

Referências

- [1] PEDRINI, Hélio; SCHWARTZ, William Robson. *Análise de Imagens Digitais* (1ª edição). São Paulo, Thomson Learning, 2008.
- [2] GONZALEZ, Rafael C; WOODS, Richard E. *Processamento Digital de Imagens* (3ª edição). São Paulo, Pearson Prentice Hall, 2010.
- [3] PARKER J. R. “Algorithms for Image Processing and Computer Vision”, Editor John Wiley and Sons, 1996.
- [4] ZHAO, Yuqian; GUI, Weihua; CHEN, Zhencheng. “Edge detection based on multi-structure elements morphology” *Proceedings of the 6th World Congress on Intelligent Control and Automation*, June 21 - 23, 2006, Dalian, China, vol 2, pp 9795-9798.
- [5] KIM, Changick. "Segmenting low depth-of-field image using morphological filters and region merging" *IEEE Tr. on Image Processing*, vol. 14, issue 10, pp. 1503-1511, October 2005.
- [6] MATHWORKS. Morphology fundamentals: dilation and erosion. Disponível em: <<http://www.mathworks.com/help/images/morphology-fundamentals-dilation-and-erosion.html>>. Acesso em: 16 abr. 2013.
- [7] CAMBRIDGE IN COLOUR. Understanding depth of field in photography. Disponível em: <<http://www.cambridgeincolour.com/tutorials/depth-of-field.htm>>. Acesso em: 19 abr. 2013.

Anexos

Esta seção apresenta o resultado de outros experimentos realizados com o algoritmo de segmentação proposto, que não foram adicionados no Capítulo 5. Também é mostrada nessa seção comparações com o algoritmo de segmentação de [5].

Os melhores resultados de segmentação obtidos são apresentados a seguir. A Tabela 2 apresenta as configurações utilizadas nesses resultados.

Figura	Numero de comp. conectados	Raio do Elem. Estruturante (em <i>pixels</i>)
40	1	5
41	10	25
42	3	15
43	40	25

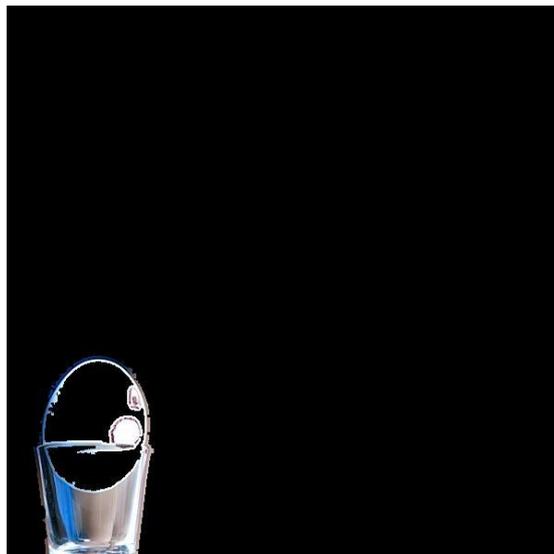
Tabela 3 - Valores utilizados nos parâmetros do algoritmo de segmentação proposto.

Resultados dos experimentos



(a)

(b)



(c)

Figura 40 - Comparação entre os métodos de segmentação de imagens: (a) imagem original, (b) resultado do algoritmo proposto (c) resultado do algoritmo de [5].



(a)



(b)



(c)

Figura 41 - Comparação entre os métodos de segmentação de imagens: (a) imagem original, (b) resultado do algoritmo proposto (c) resultado do algoritmo de [5].



(a)



(b)



(c)

Figura 42 - Comparação entre os métodos de segmentação de imagens: (a) imagem original, (b) resultado do algoritmo proposto (c) resultado do algoritmo de [5].



(a)



(b)



(c)

Figura 43 - Comparação entre os métodos de segmentação de imagens: (a) imagem original, (b) resultado do algoritmo proposto (c) resultado do algoritmo de [5].