

# Ferramenta interativa para segmentação de imagens digitais

Bruno Klava

Dezembro de 2006

Universidade de São Paulo  
Instituto de Matemática e Estatística  
MAC 499 - Trabalho de Formatura Supervisionado  
Orientadora: Profa. Dra. Nina S. T. Hirata

## Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Fundamentos</b>	<b>4</b>
2.1	Imagens digitais . . . . .	4
2.2	Grafos . . . . .	5
2.2.1	Caminhos . . . . .	6
2.2.2	Componentes conexos . . . . .	6
2.3	Morfologia matemática . . . . .	7
2.3.1	Gradiente morfológico . . . . .	7
2.4	Segmentação . . . . .	7
2.4.1	Limiarização . . . . .	10
2.4.2	Watershed . . . . .	11
2.4.3	Watershed a partir de marcadores . . . . .	14
<b>3</b>	<b>Sistema proposto</b>	<b>15</b>
3.1	Menus . . . . .	18
3.1.1	Arquivo . . . . .	19
3.1.2	Cor . . . . .	20
3.1.3	Filtros . . . . .	20
3.1.4	Opções . . . . .	20
3.1.5	Ajuda . . . . .	21

3.2	Ferramentas . . . . .	21
3.2.1	Pincel/Borracha . . . . .	22
3.2.2	Cor dos marcadores . . . . .	22
3.2.3	Seletor de cores . . . . .	22
3.2.4	Diâmetro do pincel/borracha . . . . .	22
3.2.5	Segmentar . . . . .	22
3.2.6	Zoom . . . . .	23
3.3	Camadas . . . . .	23
3.3.1	Linhas de watershed . . . . .	23
3.3.2	Marcadores . . . . .	23
3.3.3	Imagem filtrada . . . . .	24
3.3.4	Imagem de entrada . . . . .	24
<b>4</b>	<b>Resultados</b>	<b>24</b>
<b>5</b>	<b>Conclusão</b>	<b>29</b>
<b>6</b>	<b>Parte subjetiva</b>	<b>30</b>
<b>A</b>	<b>Apêndice - Diagramas UML</b>	<b>32</b>
A.1	Pacote Estruturas . . . . .	32
A.2	Pacote Filtros . . . . .	33
A.3	Pacote Formatos . . . . .	33
A.4	Pacote GUI . . . . .	34
A.5	Pacote Util . . . . .	35

## Lista de Figuras

1	Imagem digital. . . . .	5
2	Representação matricial da figura 1. . . . .	5
3	Diferenças entre 4-conectividade e 8-conectividade. . . . .	6
4	Caminhos com diferentes conectividades. . . . .	6
5	Imagem com gradiente morfológico. . . . .	9
6	Segmentação por limiarização. . . . .	11
7	Escolha de limiar adequado. . . . .	12
8	Watershed como simulação de imersão. . . . .	13
9	Segmentação automática da figura 5(a). . . . .	14
10	Segmentação por marcadores automáticos. . . . .	17
11	Captura de tela da ferramenta em execução. . . . .	18
12	Caixa de ferramentas. . . . .	21
13	Janela de camadas. . . . .	23

14	Resultado obtido com a ferramenta (#5 (42049)). . . . .	25
15	Resultado obtido com a ferramenta (#60 (223061)). . . . .	26
16	Resultado obtido com a ferramenta (#58 (227092)). . . . .	27
17	Resultado obtido com a ferramenta (#92 (8746)). . . . .	28
18	Pacotes da ferramenta. . . . .	32
19	Pacote Estruturas. . . . .	32
20	Pacote Filtros. . . . .	33
21	Pacote Formatos. . . . .	33
22	Pacote GUI. . . . .	34
23	Pacote Util. . . . .	35

## Lista de Algoritmos

1	Rotulação de componentes conexos . . . . .	8
2	Limiarização . . . . .	10
3	Watershed a partir de marcadores . . . . .	16

## 1 Introdução

A segmentação é uma etapa importante em praticamente todos os problemas que envolvem análise de imagens digitais. A segmentação tem por objetivo particionar o domínio espacial da imagem, de forma a demarcar as regiões de interesse. Tais regiões geralmente correspondem aos objetos alvos da análise em questão. Uma segmentação imprecisa pode comprometer os resultados da análise.

A segmentação é um processamento difícil, uma vez que imagens variam muito e também porque muitas vezes não é fácil descrever formalmente o que desejamos segmentar.

Por estes motivos, os sistemas de análise de imagem geralmente procuram estratégias para facilitar o processo de segmentação. Em alguns tipos de exames médicos, como a tomografia computadorizada, por exemplo, é comum injetar algum tipo de solução de contraste, para ressaltar o tecido que será analisado em relação aos demais.

Como nem sempre é possível facilitar o processo de segmentação através de técnicas para melhoria de contraste, geralmente são desenvolvidos sistemas para segmentar imagens restritas a um domínio específico, como por exemplo, um sistema para segmentação de vasos sanguíneos em imagens de retina.

A transformação watershed reduz o problema de segmentação a um problema de encontrar marcadores para as regiões de interesse. Apesar de este

ser um problema em geral mais simples que aquele, achar os marcadores de forma automática não é uma tarefa trivial.

Para contornar a dificuldade em se especificar formalmente as regiões de interesse, este trabalho propõe o desenvolvimento de uma ferramenta interativa para a segmentação de imagens digitais em níveis de cinza. A ferramenta tem como objetivo principal facilitar a criação e edição manual de marcadores.

Seguindo esta introdução, na seção 2 apresentamos alguns conceitos básicos e descrevemos o algoritmo watershed. Na seção 3 descrevemos as funcionalidades da ferramenta implementada. Na seção 4 apresentamos alguns resultados de segmentação obtidos com o uso da ferramenta e na seção 5 apresentamos as conclusões e passos futuros deste trabalho. No apêndice são apresentados diagramas UML da ferramenta implementada.

## 2 Fundamentos

Nesta seção, são apresentados conceitos teóricos e terminologias necessárias para a compreensão do trabalho realizado.

### 2.1 Imagens digitais

Uma imagem pode ser definida como uma função bidimensional,  $f(x, y)$ , na qual  $x$  e  $y$  são coordenadas espaciais e o valor assumido por  $f$  no ponto  $(x, y)$  é chamado de intensidade da imagem no ponto  $(x, y)$ . Quando as coordenadas espaciais e a intensidade assumem valores finitos e discretos, chamamos a imagem de imagem digital.

Cada elemento da imagem digital é chamado de pixel (contração de *picture element*), e associa um ponto  $(x, y)$  do domínio espacial a uma intensidade.

Por exemplo, a figura 1 representa uma imagem digital de 5 pixels de largura por 5 pixels de altura. Uma forma equivalente de representar a imagem é através de uma matriz, onde cada elemento é o valor do nível de cinza do pixel de posição correspondente, como na figura 2.

Neste trabalho, são consideradas apenas imagens digitais em níveis de cinza, assumindo intensidades no intervalo  $[0, 255]$ , que corresponde a uma escala de cinzas entre o preto (0) e o branco (255).

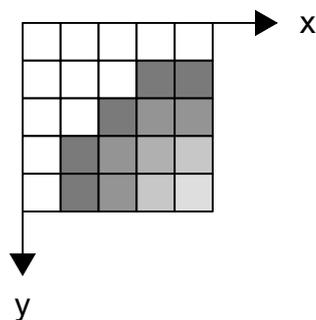


Figura 1: Imagem digital.

$$f = \begin{pmatrix} 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 122 & 122 \\ 255 & 255 & 122 & 147 & 147 \\ 255 & 122 & 147 & 176 & 199 \\ 255 & 122 & 147 & 199 & 222 \end{pmatrix}$$

Figura 2: Representação matricial da figura 1.

## 2.2 Grafos

Para qualquer conjunto  $V$ , denotamos por  $V^2$  o conjunto de todos os pares não ordenados de elementos de  $V$ . Assim, cada elemento de  $V^2$  terá a forma  $\{v, w\}$ , sendo  $v$  e  $w$  dois elementos distintos de  $V$ .

Um grafo (não direcionado)  $G(V, A)$  é uma estrutura formada por um conjunto  $V$  de vértices e um conjunto  $A$  de arestas, tal que  $A \subseteq V^2$ .

Uma imagem digital pode ser definida como um grafo, no qual cada pixel é um vértice, e as arestas ligam os pixels vizinhos, segundo alguma definição de vizinhança:

**4-conectividade** Os pixels vizinhos por 4-conectividade de um pixel  $p$  são os pixels posicionados ao redor de  $p$  na direção vertical ou horizontal, indicados em cinza na figura 3(a). Defina um grafo 4-conectado, como na figura 3(c).

**8-conectividade** Os pixels vizinhos por 8-conectividade de um pixel  $p$  são os pixels posicionados ao redor de  $p$ , tanto nas direções vertical e horizontal quanto nas diagonais, indicados em cinza na figura 3(b). Defina um grafo 8-conectado, como na figura 3(d).

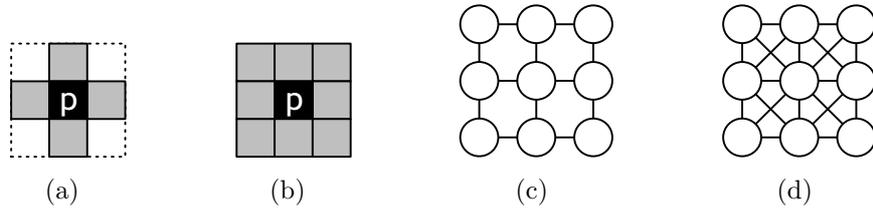


Figura 3: Diferenças entre 4-conectividade e 8-conectividade.

### 2.2.1 Caminhos

Um caminho entre dois vértice  $v_1$  e  $v_n$  em um grafo  $G(V, A)$  é uma seqüência  $v_1 v_2 \dots v_n$ , tal que  $\{v_i, v_{i+1}\} \in A$ , para  $1 \leq i < n$ ,  $v_i \in V$ , para  $1 \leq i \leq n$ , e  $v_i \neq v_j, \forall i, j$ , com  $i \neq j$ . Indicamos que existe um caminho entre  $v_1$  e  $v_n$  por  $v_1 \rightsquigarrow v_n$ .

O comprimento de um caminho é igual ao número de arestas que o compõem.

A definição de caminho depende diretamente do tipo de vizinhança adotada. Podemos notar este fato na figura 4, na qual  $abgh$  é um caminho 4-conectado, enquanto que  $ciej$  é um caminho 8-conectado.

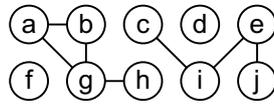


Figura 4: Caminhos com diferentes conectividades.

Uma observação que pode ser feita é que todo caminho 4-conectado é um caminho 8-conectado. A diferença entre os tipos de vizinhança fica explícita quando queremos obter o menor caminho entre dois vértices. Por exemplo, na figura 4, adotando a 8-conectividade, o menor caminho entre os vértices  $a$  e  $g$  tem comprimento 1, mas se utilizarmos a 4-conectividade, o menor caminho terá comprimento 2.

### 2.2.2 Componentes conexos

Um componente conexo em um grafo  $G(V, A)$  é formado por um conjunto de vértices  $C \subseteq V$  tal que, se  $p \in C$  e  $p \rightsquigarrow q$ , então  $q \in C, \forall q \in V$ .

Na figura 4, temos 4 componentes conexos:  $\{a, b, g, h\}$ ,  $\{c, e, i, j\}$ ,  $\{d\}$  e  $\{f\}$ .

No algoritmo 1 apresentamos como rotular os componentes conexos de uma imagem colorida, considerando como vizinhos os pixels conectados que têm a mesma cor. Os pixels do fundo, que têm a cor  $corFundo$ , são desconsi-

derados e recebem 0 como rótulo. Como saída do algoritmo, temos a imagem  $L$  com cada componente conexo rotulado com um número inteiro diferente.

## 2.3 Morfologia matemática

A morfologia matemática, baseada na teoria de reticulados<sup>1</sup>, estuda a decomposição de operadores em função de dois operadores elementares: a erosão e a dilatação.

Seja  $f$  uma imagem. A erosão de  $f$ , definida pela equação 1, é formada pelo valor mínimo dos níveis de cinza da imagem na vizinhança de  $x$ , para todo pixel  $x$  de  $f$ . A dilatação de  $f$ , definida pela equação 2, é formada pelo valor máximo dos níveis de cinza da imagem na vizinhança de  $x$ , para todo pixel  $x$  de  $f$ . Para os dois operadores, a vizinhança é definida por  $B$ , chamado de elemento estruturante<sup>2</sup>.

$$[\epsilon_B(f)](x) = \min\{f(z) : z \in B \text{ trasladado de } x\} \quad (1)$$

$$[\delta_B(f)](x) = \max\{f(z) : z \in B \text{ trasladado de } x\} \quad (2)$$

### 2.3.1 Gradiente morfológico

Para localizar as bordas dos objetos numa imagem digital, pode-se aplicar o gradiente morfológico, definido pela equação 3.

$$[\nabla_B(f)](x) = [\delta_B(f)](x) - [\epsilon_B(f)](x) \quad (3)$$

O gradiente morfológico possibilita achar as regiões da imagem em que há maior variação dos níveis de cinza. Como uma borda (limite entre dois objetos) é uma região onde os níveis de cinza variam mais, o gradiente responde mais forte (resultando em pixels com nível de cinza mais claros) nessas regiões do que no restante da imagem. Na figura 5, observamos o realce das bordas a partir do gradiente morfológico da imagem original, utilizando o quadrado 3x3 como elemento estruturante.

## 2.4 Segmentação

Uma das etapas mais importantes em praticamente qualquer sistema de análise de imagens é a segmentação, que consiste em definir na imagem

<sup>1</sup>Neste trabalho, é considerado o reticulado das imagens em níveis de cinza.

<sup>2</sup>As definições de erosão e dilatação apresentadas são válidas para elementos estruturantes planos e simétricos.

**Algoritmo 1:** Rotulação de componentes conexos

$L = \text{rotulacao}(f, \text{corFundo})$

$f$ : imagem de entrada (colorida)

$\text{corFundo}$ : cor dos pixels do fundo da imagem

$L$ : imagem de saída rotulada

$\text{fila}$ : fila simples (política *First In First Out*)

$\text{rotulo}$ : rótulo do componente conexo sendo rotulado

$\text{corAtual}$ : cor do componente conexo sendo rotulado

1. Inicialização

**para cada pixel  $p$  de  $f$  faça**

**se**  $f(p) = \text{corFundo}$  **então**

$L(p) \leftarrow 0$

**senão**

$L(p) \leftarrow -1$

$\text{rotulo} \leftarrow 0$

2. Rotulação

**para cada pixel  $p$  de  $f$  faça**

**se**  $L(p) = -1$  **então**

$\text{corAtual} \leftarrow f(p)$

$\text{rotulo} \leftarrow \text{rotulo} + 1$

$L(p) \leftarrow \text{rotulo}$

$\text{fila.insere}(p)$

**enquanto**  $\text{fila.vazia}()$  **faça**

$q \leftarrow \text{fila.remove}()$

**para cada vizinho  $v$  de  $q$  faça**

**se**  $L(v) = -1$  e  $f(v) = \text{corAtual}$  **então**

$L(v) \leftarrow \text{rotulo}$

$\text{fila.insere}(v)$



(a) Imagem original



(b) Gradiente morfológico

Figura 5: Imagem com gradiente morfológico.

linhas que delimitam a borda de cada objeto de interesse. O processo de segmentação subdivide uma imagem em suas partes constituintes. Tal subdivisão deve ser feita até o nível de detalhe desejado, que depende fortemente do conteúdo da imagem e da análise que será feita posteriormente.

Os algoritmos existentes para segmentação de imagens baseiam-se geralmente na descontinuidade ou similaridade dos níveis de cinza dos pixels. Dentre as diversas abordagens, podemos citar as baseadas em limiarização, crescimento de regiões, redes neurais, métodos estatísticos, transformada de Fourier, transformada de wavelet e na transformação watershed.

Os resultados de técnicas automáticas para a segmentação nem sempre são satisfatórios. Dependendo da situação, é muito mais fácil “corrigir” ou “fazer” uma segmentação à mão do que escolher e ajustar os parâmetros dos algoritmos. Tendo este problema em mente, para um sistema produzir melhores resultados, ele deve incorporar formas do usuário interagir durante o processo de segmentação.

### 2.4.1 Limiarização

Uma das abordagens mais intuitivas para a segmentação de imagens digitais é a limiarização, ou *thresholding*, especificada no algoritmo 2. A idéia por trás desta técnica é que os objetos de interesse têm níveis de cinza diferentes em relação ao fundo da imagem, logo, se um limiar adequado for escolhido, separamos facilmente os objetos do fundo. A escolha do limiar pode ser feita de forma automática, por exemplo, através da análise do histograma da imagem, que é um gráfico com a frequência dos níveis de cinza na imagem.

#### Algoritmo 2: Limiarização

```
 $f = \text{thresholding}(f, \lambda)$   
 $f$ : imagem de entrada  
 $\lambda$ : limiar de separação  
para cada pixel  $p$  de  $f$  faça  
    se  $f(p) \leq \lambda$  então  
         $f(p) \leftarrow 0$   
    senão  
         $f(p) \leftarrow 255$ 
```

Na figura 6, temos um exemplo simples de segmentação por limiarização, abordagem comum em sistemas para reconhecimento de caracteres.

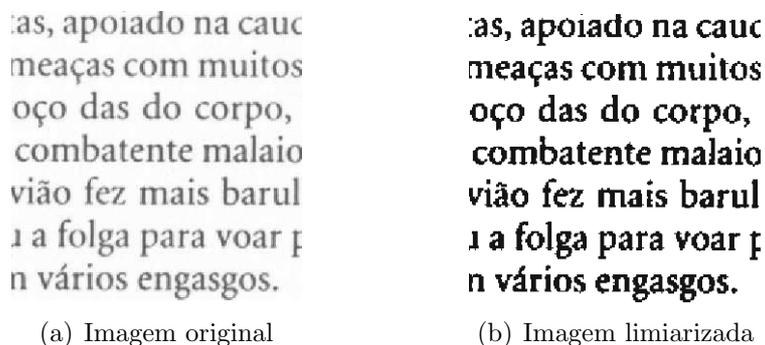


Figura 6: Segmentação por limiarização.

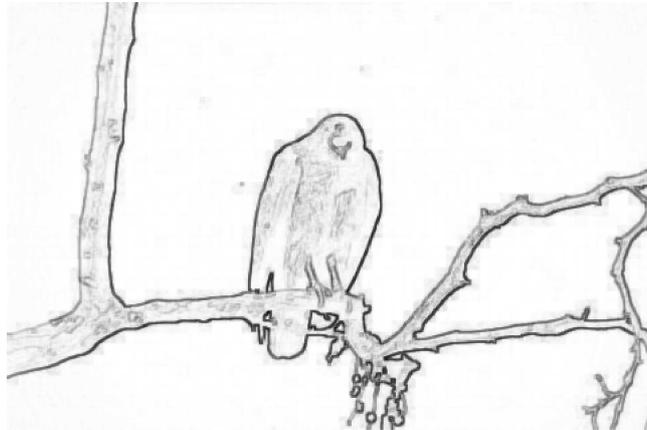
Em certos casos, é muito difícil encontrar um limiar adequado. Por exemplo, para segmentar a imagem da figura 5(a) a partir da limiarização de seu gradiente morfológico invertido (figura 7(a)), se escolhermos um limiar muito baixo, nem todas as bordas são mantidas, como na figura 7(b). Aumentando o limiar de forma a manter mais bordas, acabamos mantendo também detalhes da textura dos objetos, como na figura 7(c), o que pode prejudicar a análise posterior. Além de não conseguirmos especificar o nível de detalhes resultante da segmentação, as bordas obtidas são grosseiras, sendo necessário um pós-processamento para deixá-las com uma espessura adequada (1 pixel).

#### 2.4.2 Watershed

A transformação watershed é uma abordagem bastante utilizada em segmentação de imagens. Sua formulação mais intuitiva é baseada na simulação de imersão.

Considere a imagem de entrada em níveis de cinza como uma superfície topográfica. O objetivo é produzir linhas de divisão de águas (watersheds) nesta superfície. Para tal, um furo é feito em cada mínimo local  $M_k$  da superfície. A superfície é submersa a uma taxa constante, de modo que a água entre pelos mínimos locais. Quando frentes de água, vindas de diferentes mínimos locais, estão prestes a se encontrar, uma barreira é construída para evitar tal encontro. Em algum momento, o processo chega a um estado tal que somente os topos das barreiras estão visíveis acima do nível da água, correspondendo às linhas de watershed (em verde na figura 8). Dessa forma, a cada mínimo local  $M_k$  é associada uma represa  $CB_k$ .

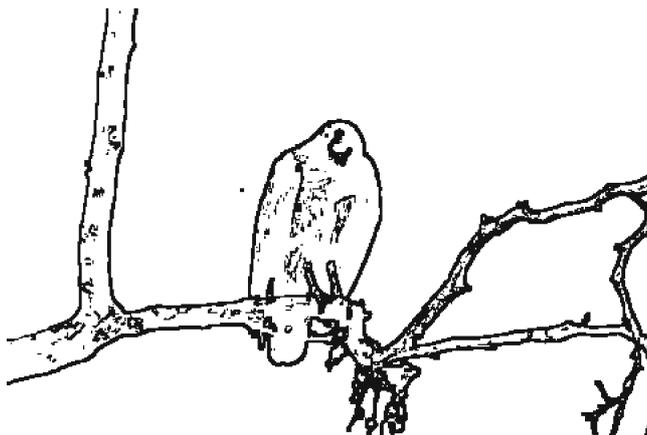
Uma outra forma de entender a transformação watershed é a partir da simulação de chuva, utilizando linhas de inclinação mais íngremes, ou *steepest slope lines*, como definido em [VS91]. A represa  $CB_k$ , associada ao mínimo  $M_k$ , é formada pelo conjunto de pixels  $p$ , tal que, quando uma gota de chuva



(a) Gradiente morfológico invertido da figura 5(a)



(b) Limiarização com  $\lambda = 140$



(c) Limiarização com  $\lambda = 210$

Figura 7: Escolha de limiar adequado.

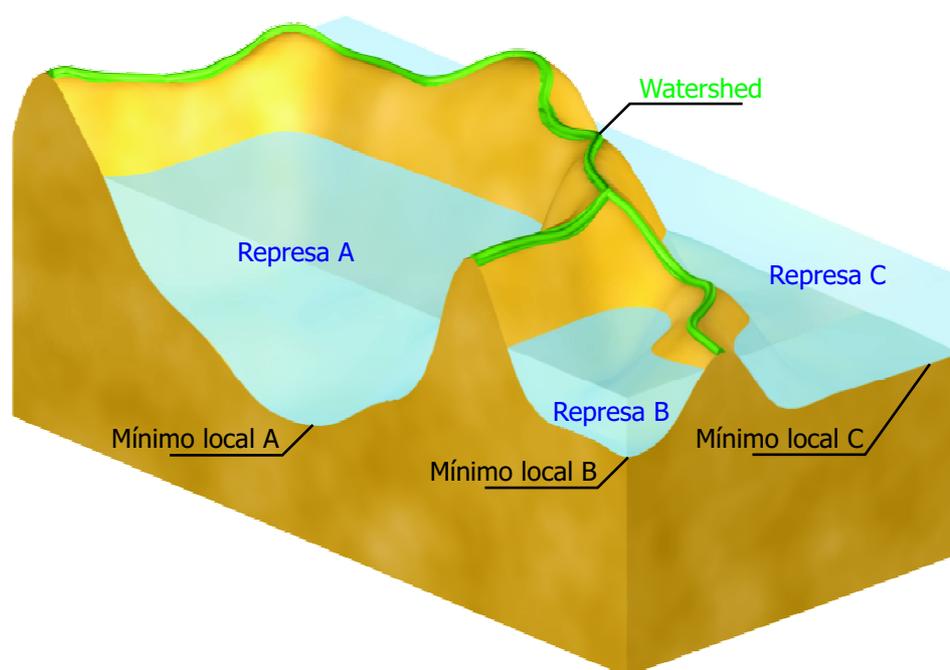


Figura 8: Transformação watershed vista como simulação de imersão.

cai sobre  $p$ , ela escoia pela linha de inclinação mais íngreme, até chegar a  $M_k$ . As linhas de watershed ficam definidas pelas linhas que separam diferentes represas.

Ao segmentar uma imagem a partir do watershed de seu gradiente morfológico, obtemos um resultado super segmentado, como observamos na figura 9, pois o gradiente morfológico geralmente apresenta muitos mínimos locais, devido aos ruídos e texturas dos objetos da imagem.

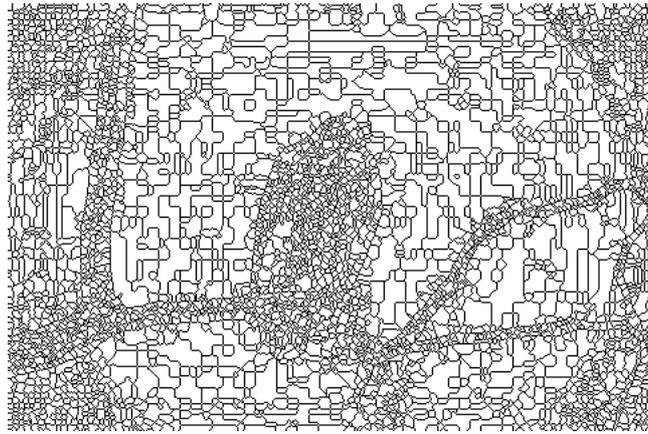


Figura 9: Segmentação automática da figura 5(a).

### 2.4.3 Watershed a partir de marcadores

A transformação watershed a partir de marcadores é uma abordagem para contornar o problema da super segmentação. Também pode ser descrita como uma simulação de imersão, mas ao invés de fazer furos nos mínimos locais, furamos os pixels correspondentes aos marcadores. Dessa maneira, o problema de segmentação fica reduzido ao problema de achar os marcadores apropriados.

O algoritmo 3, como descrito em [DL03], é baseado no caminho de menor custo entre cada pixel e o conjunto de marcadores, de forma que a cada marcador  $M_k$  (componente conexo do conjunto de marcadores) é associada uma represa  $CB_k$ , conforme a equação 4, na qual a imagem é modelada como um grafo.  $C^*(M, p)$  é o custo do caminho de custo mínimo da região  $M$  ao pixel  $p$ , que é definido pelo custo do caminho de custo mínimo de qualquer pixel de  $M$  a  $p$ , conforme a equação 5. O custo do caminho entre 2 pixels  $p_1$  e  $p_n$  é dado por um custo lexicográfico  $C(p)$ , definido pelas equações 6 a 9, no qual o primeiro componente  $C^1(p)$  é o nível de cinza máximo no caminho

e o segundo componente  $C^2(p)$  é o número de vezes que o custo do primeiro componente é o mesmo antes de chegar a  $p_n$ .

$$CB_k = \{p : C^*(M_k, p) \leq C^*(M_j, p), j \neq k\} \quad (4)$$

$$C^*(M, p) = \min\{C^*(m, p) : m \in M\} \quad (5)$$

$$C(p_1, \dots, p_n) = [C^1(p_n), C^2(p_n)] \quad (6)$$

$$C^1(p_1) = 0, \quad (7)$$

$$C^1(p_n) = \max\{C^1(p_n), f(p_2), \dots, f(p_n)\} \text{ para } n > 1, \quad (8)$$

$$C^2(p_n) = \max\{j : C^1(p_n) = C^1(p_{n-j}), j = 0, 1, \dots, n-1\} \quad (9)$$

Como entrada do algoritmo 3, temos duas imagens:  $f$  e  $L$ .  $f$  é a imagem em níveis de cinza que desejamos segmentar (geralmente, essa imagem é formada pelo gradiente da imagem que se deseja analisar).  $L$  é uma imagem contendo os  $n$  marcadores rotulados, ou seja, dado um marcador  $M_k$ , para  $1 \leq k \leq n$ , todos seus pixels são rotulados com  $k$ , e os pixels do fundo são rotulados com 0.

Inicialmente, todos os pixels de  $L$  são não-permanentes. Como saída do algoritmo, temos  $L$  com cada represa  $CB_k$  (associada ao marcador  $M_k$ ) rotulada com  $k$ . Além disso, a imagem binária<sup>3</sup>  $ws$  indica as linhas de watershed resultantes.

Uma forma de gerar automaticamente os marcadores é a partir da limiarização do gradiente morfológico da imagem. Dessa forma, reduzimos o problema da super segmentação, mas independentemente do limiar escolhido, é difícil obter um resultado satisfatório, como observamos na figura 10.

### 3 Sistema proposto

Este trabalho propõe o desenvolvimento de uma ferramenta interativa para a segmentação de imagens. A interatividade do usuário se concentra na manipulação de marcadores, que são utilizados na segmentação pelo algoritmo 3, descrito na página 16.

A ferramenta foi desenvolvida utilizando a plataforma Java (J2SE 5.0) da Sun Microsystems e o ambiente de desenvolvimento Eclipse. Por ser implementada em Java, a ferramenta pode ser executada em qualquer ambiente

---

<sup>3</sup>Em imagens binárias, 0 indica pixels brancos, enquanto 1 indica pixels pretos.

**Algoritmo 3:** Watershed a partir de marcadores

$L = cb(f, L)$

$f$ : imagem de entrada

$L$ : imagem rotulada (entrada e saída)

$fila$ : fila de prioridade (inicialmente vazia)

$ws$ : imagem binária (inicialmente toda preenchida com 0)

1. *Inicialização*

**para cada pixel  $p$  com  $L(p) \neq 0$  faça**

└  $fila.insere(p, 0)$

2. *Propagação*

**enquanto não  $fila.vazia()$  faça**

└  $p \leftarrow fila.remove()$

└ marcar  $p$  como permanente

└ **para cada vizinho  $q$  não-permanente de  $p$  faça**

└└ **se  $q$  não está rotulado então**

└└└  $L(q) \leftarrow L(p)$

└└└  $fila.insere(q, f(q))$

└└ **senão**

└└└ **se  $L(q) \neq L(p)$  e  $ws(q) = 0$  e  $ws(p) = 0$  então**

└└└└  $ws(q) \leftarrow 1$

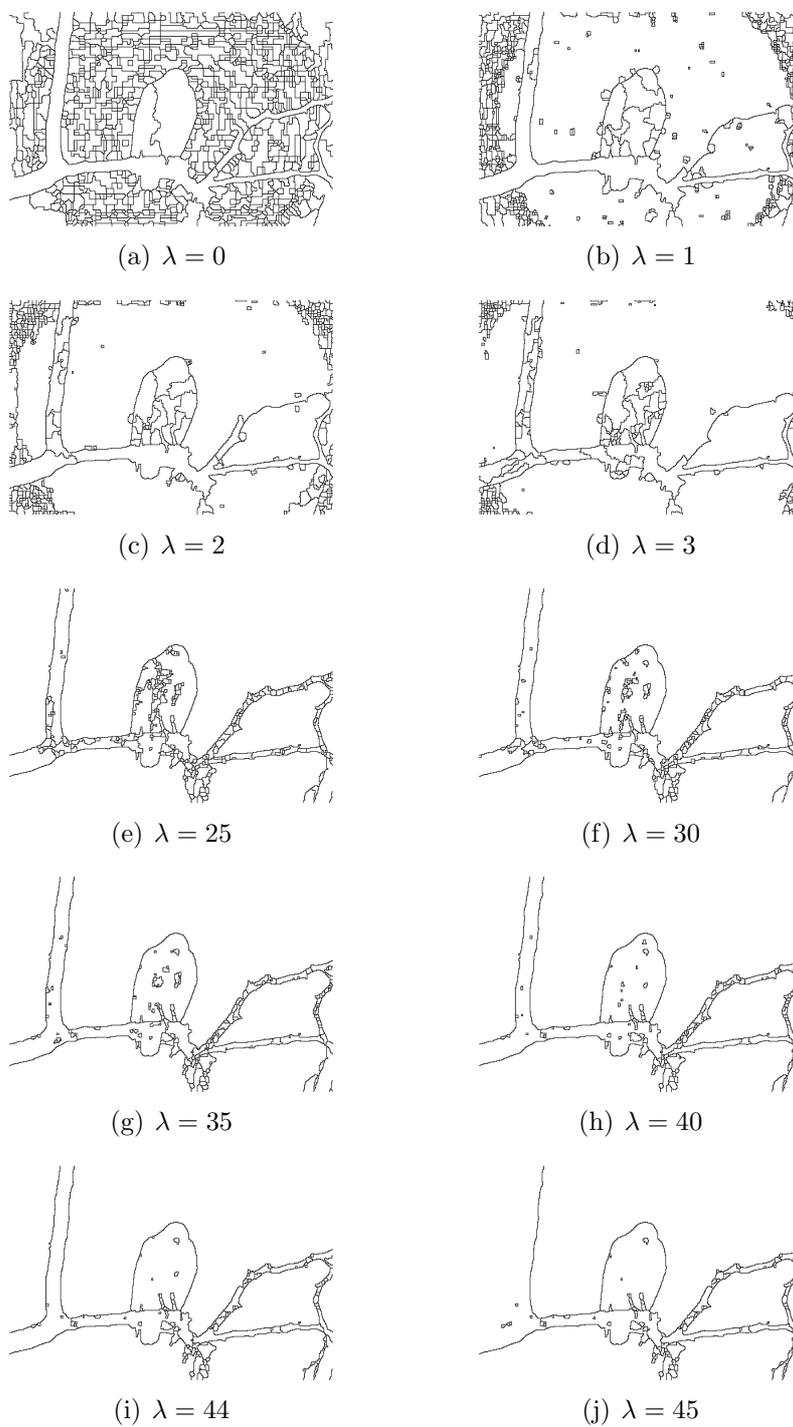


Figura 10: Segmentação a partir de marcadores gerados automaticamente.

que possua uma JVM<sup>4</sup> instalada. Na figura 11, é exibida uma captura de tela da ferramenta em execução.

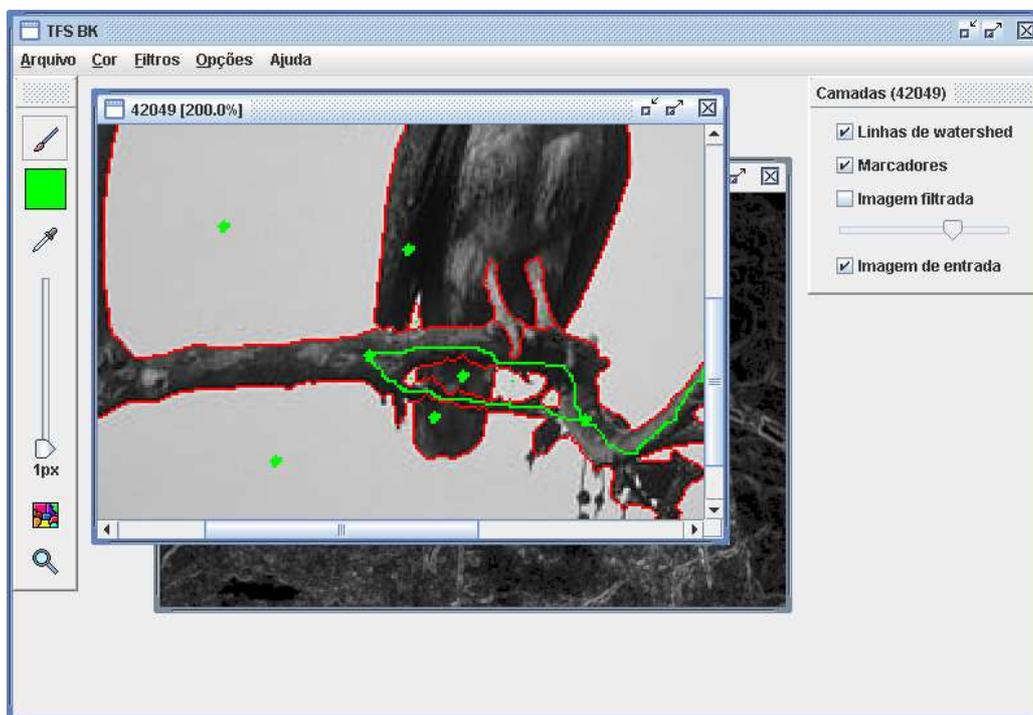


Figura 11: Captura de tela da ferramenta em execução (marcadores em verde e watershed em vermelho).

A seguir, são descritas as funcionalidades da ferramenta através das partes de sua interface gráfica: menus, ferramentas e camadas.

### 3.1 Menus

- Arquivo
  - Abrir arquivo de imagem
  - Salvar imagem segmentada
  - Abrir imagem de marcadores
  - Salvar imagem de marcadores
  - Sair
  
- Cor

---

<sup>4</sup>Java Virtual Machine

- Watershed
- Fundo
- Filtros
  - Nenhum
  - Gradiente morfológico
- Opções
  - 4-conectividade
  - 8-conectividade
  - Segmentar automaticamente
  - Aplicar gradiente morfológico automaticamente
  - Ocultar imagem filtrada automaticamente
  - Reposicionar janelas
- Ajuda
  - Formatos de imagem suportados

### 3.1.1 Arquivo

Através do menu *Arquivo*, o usuário pode abrir e salvar imagens.

A opção *Abrir arquivo de imagem* permite que o usuário navegue pelo sistema de arquivos para abrir o arquivo que contém a imagem que deseja segmentar. Já a opção *Salvar imagem segmentada*, permite salvar o resultado da segmentação tal como está exibido, conforme as cores utilizadas e a visibilidade das camadas (explicado nas seções 3.1.2 e 3.3, respectivamente).

Para facilitar o processo de edição de marcadores, o usuário pode salvá-los em um arquivo, através da opção *Salvar imagem de marcadores*. Num momento futuro, se for necessário algum refinamento na segmentação, basta abrir o arquivo salvo, através da opção *Abrir imagem de marcadores*, e continuar a manipular os marcadores.

Note que é possível trabalhar com mais de uma imagem por vez na ferramenta.

Detalhes sobre os formatos de arquivos suportados são dados na seção 3.1.5.

### 3.1.2 Cor

No menu *Cor*, o usuário pode configurar a cor das linhas de watershed e do fundo. Por padrão, as linhas de watershed são vermelhas (para contrastarem com a imagem em níveis de cinza) e o fundo é branco.

### 3.1.3 Filtros

Por padrão, a segmentação é feita sobre o gradiente morfológico da imagem de entrada. Para utilizar diretamente a imagem de entrada (caso já tenha a imagem do gradiente, por exemplo), sem realizar nenhum tipo de filtragem, basta utilizar a opção *Nenhum* do menu *Filtros*.

Foi definida uma interface para tornar possível a utilização de outros filtros em futuras expansões da ferramenta (mais detalhes nas seções 5 e A.2).

### 3.1.4 Opções

No menu *Opções*, o usuário pode configurar parâmetros e comportamentos da ferramenta.

As opções *4-conectividade* e *8-conectividade* definem qual tipo de vizinhança será utilizada no algoritmo 3 para definir o conjunto de pixels de cada represa. É importante notar que, se for utilizada a 4-conectividade, as linhas de watershed resultantes serão 8-conectadas, enquanto que se for escolhida a 8-conectividade, as linhas de watershed serão 4-conectadas. Por padrão, é utilizada a 4-conectividade.

A opção *Segmentar automaticamente* permite que o usuário visualize o resultado da segmentação a cada edição dos marcadores. Esta opção é selecionada por padrão.

A opção *Aplicar gradiente morfológico automaticamente* faz com que seja aplicado o filtro *Gradiente morfológico* quando uma imagem é aberta na ferramenta. Caso a opção não esteja selecionada, nenhum tipo de filtragem é realizada. Esta opção é selecionada por padrão.

A opção *Ocultar imagem filtrada automaticamente* faz com que a camada *Imagem filtrada* (seção 3.3.3) seja ocultada quando uma imagem é aberta na ferramenta. Esta opção é selecionada por padrão.

A opção *Reposicionar janelas* permite reposicionar a caixa de ferramentas (seção 3.2) e a janela de camadas (seção 3.3) nos cantos superior esquerdo e superior direito da janela da ferramenta, respectivamente. Além disso, posiciona em cascata todas as imagens abertas.

### 3.1.5 Ajuda

No menu *Ajuda*, é possível visualizar quais formatos de imagem suportados pela JVM em que a ferramenta está sendo executada são adequados para a leitura e escrita. Mais detalhes são encontrados na documentação do pacote `javax.imageio` em [Sun].

O formato WBMP foi excluído pois suporta somente imagens binárias.

O formato PNG foi escolhido como padrão para arquivos de marcadores pois é o único formato que suporta transparência e é livre para leitura e escrita (o formato GIF também suporta transparência, mas seu algoritmo de escrita é patenteado e, por isso, não é implementado na JVM).

## 3.2 Ferramentas



Figura 12: Caixa de ferramentas.

- Pincel/borracha
- Cor dos marcadores
- Seletor de cores
- Diâmetro do pincel/borracha
- Segmentar
- Zoom

### 3.2.1 Pincel/Borracha

A ferramenta *Pincel/borracha* é a principal ferramenta para a manipulação dos marcadores. A fusão do pincel e da borracha numa mesma ferramenta facilita a edição de marcadores, pois não é necessário ficar selecionando as duas ferramentas para alternar seu uso, aumentando a eficiência da edição dos marcadores. O comportamento do pincel/borracha depende do botão do mouse que é pressionado:

- Botão esquerdo: faz marcadores da cor selecionada
- Botão direito: apaga marcadores da cor selecionada
- Botão central (ou ALT + botão direito): apaga marcadores, independente das cores

Um clique duplo na ferramenta pincel/borracha apaga todos os marcadores existentes.

### 3.2.2 Cor dos marcadores

Exibe a cor corrente do pincel/borracha.

### 3.2.3 Seletor de cores

Permite selecionar a cor de algum marcador como a cor corrente do pincel/borracha.

### 3.2.4 Diâmetro do pincel/borracha

Permite alterar o diâmetro do pincel/borracha.

### 3.2.5 Segmentar

Realiza a rotulação dos marcadores atuais e a segmentação da imagem (filtrada ou de entrada, se não estiver sendo utilizado nenhum filtro). Seu uso só é útil quando a opção *Segmentar automaticamente* (seção 3.1.4) está desabilitada, o que deve ser feito quando estamos trabalhando com uma imagem muito grande ou complexa de segmentar, fazendo com que a rotulação<sup>5</sup> dos marcadores e a execução da segmentação a cada edição dos marcadores não interrompa o próprio processo de edição.

---

<sup>5</sup>Os marcadores são rotulados utilizando sempre a 8-conectividade, a fim de manter conexos os marcadores que são linhas de 1 pixel de espessura.

### 3.2.6 Zoom

Permite alterar a escala de visualização da imagem. Seu comportamento depende do botão do mouse que é pressionado:

- Botão esquerdo: multiplica a escala atual de visualização por 2
- Botão direito: divide a escala atual de visualização por 2

Um clique duplo na ferramenta zoom altera a escala de visualização para 100%.

## 3.3 Camadas

A janela de camadas, exibida na figura 13, permite controlar a visibilidade das imagens utilizadas no processo de segmentação. É nessa janela que o usuário configura quais imagens serão sobrepostas para formar a imagem de saída desejada. Para alterar a visibilidade de uma camada, basta marcar ou desmarcar a caixa de seleção correspondente.

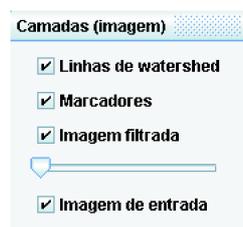


Figura 13: Janela de camadas.

### 3.3.1 Linhas de watershed

Exibe a imagem das linhas de watershed resultantes da segmentação, com a cor configurada no menu *Cor* (seção 3.1.2).

### 3.3.2 Marcadores

Exibe a imagem dos marcadores. Somente a imagem desta camada é alterada pelo pincel/borracha.

### 3.3.3 Imagem filtrada

Exibe a imagem filtrada, sobre a qual o algoritmo é aplicado. Caso nenhum filtro estiver selecionado, essa camada é desabilitada e o algoritmo é aplicado sobre a imagem de entrada.

Esta camada possui um controle a mais: uma barra de brilho, criada para facilitar a visualização da imagem do gradiente morfológico, que geralmente é muito escura. A transformação utilizada para alterar a escala de cinzas corresponde a aplicação da equação 10 para o nível de cinza  $n$  de cada pixel da imagem filtrada, de forma que os níveis de cinza sempre estão no intervalo  $[0, 255]$ . A barra de brilho altera o valor de  $\alpha$  no intervalo  $]0, 1]$ . Quando  $\alpha = 1$ , a imagem resultante é idêntica a original, e diminuindo o valor de  $\alpha$ , a imagem é clareada. Na equação 10,  $\text{round}(x)$  devolve o número inteiro mais próximo de  $x$ .

$$N(n, \alpha) = \text{round}(n^\alpha \cdot 255^{1-\alpha}), \quad n \in [0, 255], \quad \alpha \in ]0, 1] \quad (10)$$

### 3.3.4 Imagem de entrada

Exibe a imagem em níveis de cinza que se deseja segmentar.

Somente se as camadas *Imagem de entrada* e *Imagem filtrada* não estiverem visíveis é que é possível ver o fundo, com a cor definida no menu *Cor* (seção 3.1.2).

## 4 Resultados

Nesta seção, são mostrados resultados obtidos com a utilização da ferramenta. As imagens utilizadas foram retiradas do “The Berkeley Segmentation Dataset and Benchmark” [Com], que fornece uma base de imagens para comparação entre técnicas automáticas de segmentação.

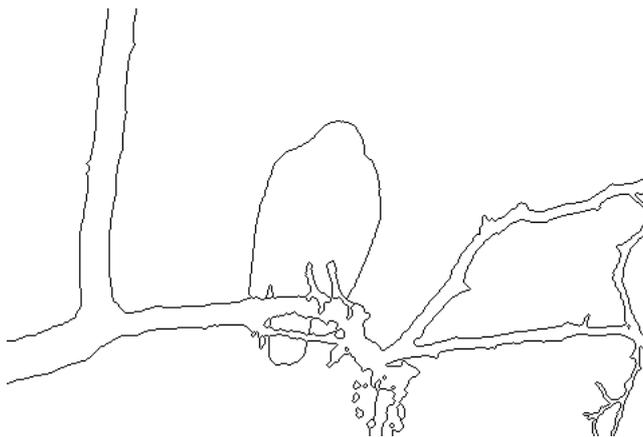
Para o leitor interessado em visualizar resultados de técnicas automáticas de segmentação para as imagens desta seção, as figuras 14, 15, 16 e 17 deste trabalho correspondem, respectivamente, às imagens #5 (42049), #60 (223061), #58 (227092) e #92 (8746) do conjunto de imagens de teste de [Com].



(a) Imagem original



(b) Marcadores sobrepostos à imagem original

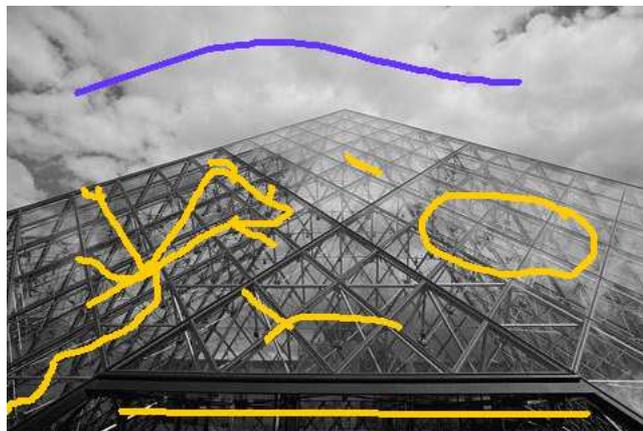


(c) Watershed

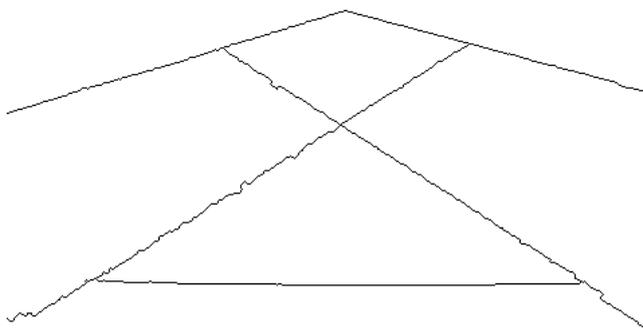
Figura 14: Resultado obtido com a ferramenta (#5 (42049)).



(a) Imagem original



(b) Marcadores sobrepostos à imagem original

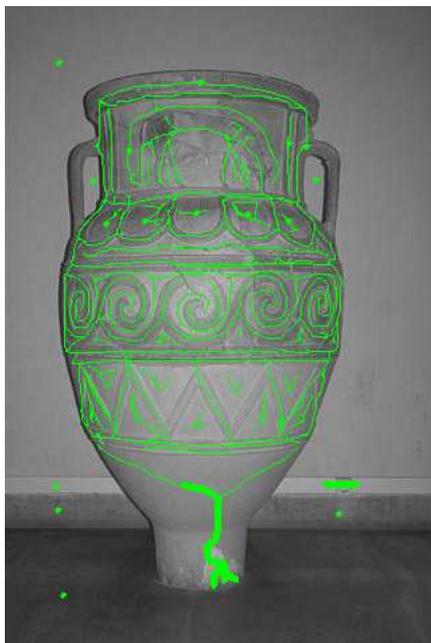


(c) Watershed

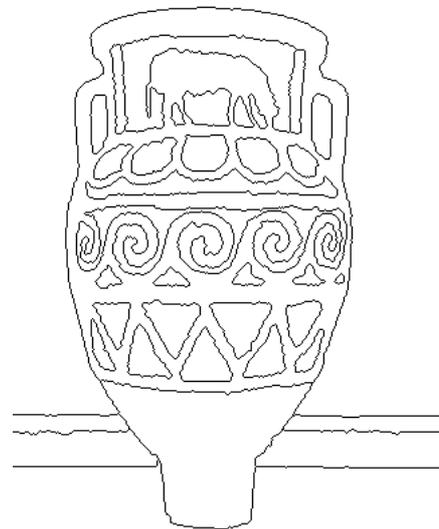
Figura 15: Resultado obtido com a ferramenta (#60 (223061)).



(a) Imagem original



(b) Marcadores sobrepostos à imagem original

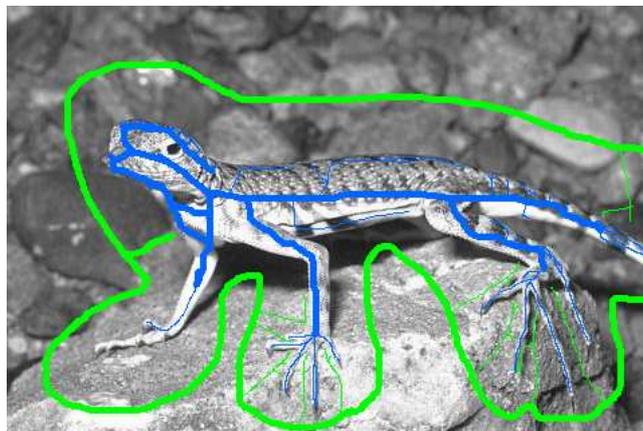


(c) Watershed

Figura 16: Resultado obtido com a ferramenta (#58 (227092)).



(a) Imagem original



(b) Marcadores sobrepostos à imagem original



(c) Watershed

Figura 17: Resultado obtido com a ferramenta (#92 (8746)).

## 5 Conclusão

A ferramenta facilita a obtenção da segmentação das áreas desejadas em imagens de tipos diversos, o que seria impraticável utilizando ferramentas sem intervenção humana, que normalmente só apresentam resultados satisfatórios quando são utilizadas para imagens restritas a um domínio específico, em situações bem controladas.

As opções disponíveis na ferramenta tornam o processo de segmentação num processo interativo. Dessa forma, através da edição dos marcadores, o usuário tem um controle fino sobre o nível de detalhes que deseja obter.

Por exemplo, a opção *Segmentar automaticamente*, permite que o usuário visualize o efeito que cada marcador têm sobre o resultado final da segmentação, facilitando a escolha de marcadores adequados. Para imagens pequenas, a rotulação dos marcadores e o cálculo do watershed é muito rápido, de forma que o usuário pode visualizar quase que instantaneamente o resultado obtido com os marcadores.

Outra opção interessante da ferramenta é o uso de marcadores de diferentes cores que, em segmentações mais complexas, permite separar claramente um marcador de outro, que seriam rotulados como sendo o mesmo, caso fosse levado em conta somente a vizinhança dos seus pixels. Além disso, esta opção tem um caráter didático interessante, pois permite diferenciar o fundo da imagem e os objetos analisados entre si.

Em trabalhos futuros, a ferramenta irá incorporar técnicas de segmentação hierárquica, de forma a gerar automaticamente um conjunto inicial de marcadores, minimizando o esforço do usuário para obter o resultado desejado. Também é previsto incorporar técnicas para a segmentação de imagens coloridas e imagens de vídeo.

Para facilitar as futuras expansões da ferramenta, foram definidas interfaces para interação entre os objetos de sua estrutura. Por exemplo, foi definida uma interface para criação de filtros, pois serão necessários outros tipos de filtros para tratar imagens coloridas. Mesmo para imagens em níveis de cinza, outros filtros poderão ser incluídos, como filtros conexos, por exemplo, que têm a propriedade de eliminar algumas bordas (mas nunca modificar a posição ou introduzir novas bordas).

## 6 Parte subjetiva

Um dos principais motivos que me levou a escolher uma graduação em computação foi o interesse na manipulação de imagens por computador. Por isso, a realização deste projeto de formatura foi muito gratificante para mim, pois tive a oportunidade de aprofundar meus estudos na área de processamento de imagens.

O maior desafio encontrado foi conseguir gerenciar o tempo entre o projeto, o estágio e as tarefas de outras disciplinas da graduação. Mesmo com esse problema, considero que o projeto foi um sucesso, afinal o objetivo foi alcançado, pois foi desenvolvida uma ferramenta com resultados satisfatórios, com as características desejadas. Além disso, foi uma ótima oportunidade para aplicar diversos conhecimentos adquiridos durante a graduação.

Dentre as disciplinas do BCC, considero como mais importantes as relacionadas a desenvolvimento de algoritmos:

- MAC0122 (Princípios de Desenvolvimento de Algoritmos)
- MAC0323 (Estruturas de Dados)
- MAC0328 (Algoritmos em Grafos)

as relacionadas ao desenvolvimento de sistemas:

- MAC0211 (Laboratório de Programação I)
- MAC0242 (Laboratório de Programação II)
- MAC0441 (Programação Orientada a Objetos)
- MAC0332 (Engenharia de Software)
- MAC0413 (Tópicos de Programação Orientada a Objetos)

e as relacionadas às áreas de processamento de imagem e computação gráfica:

- MAC0417 (Visão e Processamento de Imagens)
- MAC0420 (Introdução a Computação Gráfica)
- MAC0447 (Análise e Reconhecimento de Formas: Teoria e Prática)

No desenvolvimento do projeto, tive a oportunidade de aplicar muitos conceitos estudados nas disciplinas citadas. Foi interessante observar, na prática, como a eficiência de um sistema depende fortemente da escolha de estruturas de dados e algoritmos adequados, além de como a organização inteligente da estrutura de um sistema facilita a inclusão de novas funcionalidades.

Agradeço à professora Nina, que foi muito atenciosa durante todo o projeto. A interação com ela foi de extrema importância, tanto pela sugestão da idéia para o projeto, quanto pelas suas valiosas colaborações com idéias para melhoria das funcionalidades da ferramenta implementada e da escrita desta monografia.

Como observado na conclusão deste texto, existem diversas funcionalidades a serem acrescentadas na ferramenta implementada. Pretendo fazer meu mestrado na área de processamento de imagens, estudando mais profundamente os trabalhos realizados na área, com o objetivo inicial de incorporar as funcionalidades já citadas.

## A Apêndice - Diagramas UML

Nesta seção, são apresentados diagramas UML da ferramenta desenvolvida.

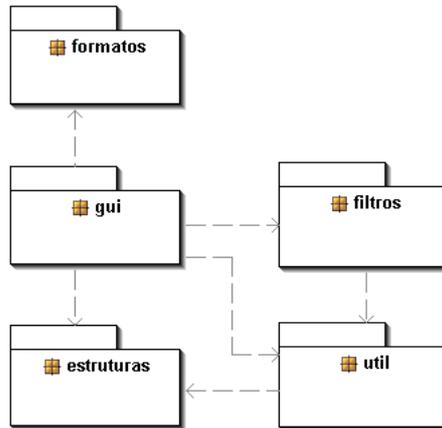


Figura 18: Pacotes da ferramenta.

### A.1 Pacote Estruturas

Formado pelas classes das estruturas de dados utilizadas na ferramenta.

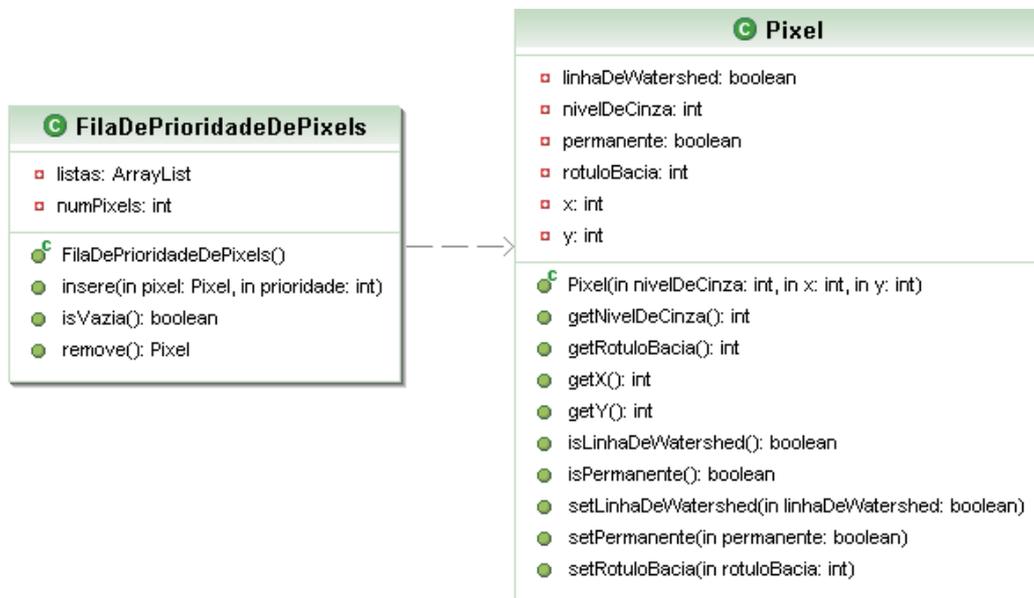


Figura 19: Pacote Estruturas.

## A.2 Pacote Filtros

Formado pelas classes e interface responsáveis pela filtragem de imagens.

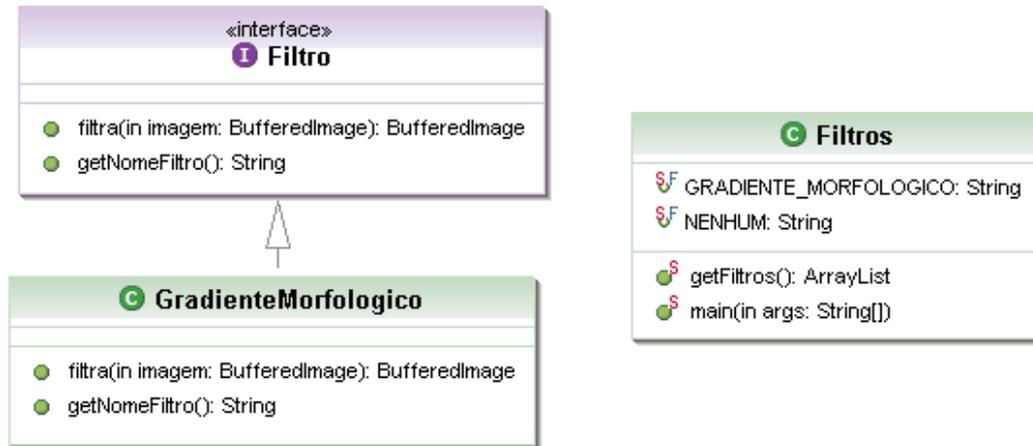


Figura 20: Pacote Filtros.

## A.3 Pacote Formatos

Formado pelas classes que gerenciam os fomatos de arquivos suportados pela ferramenta.

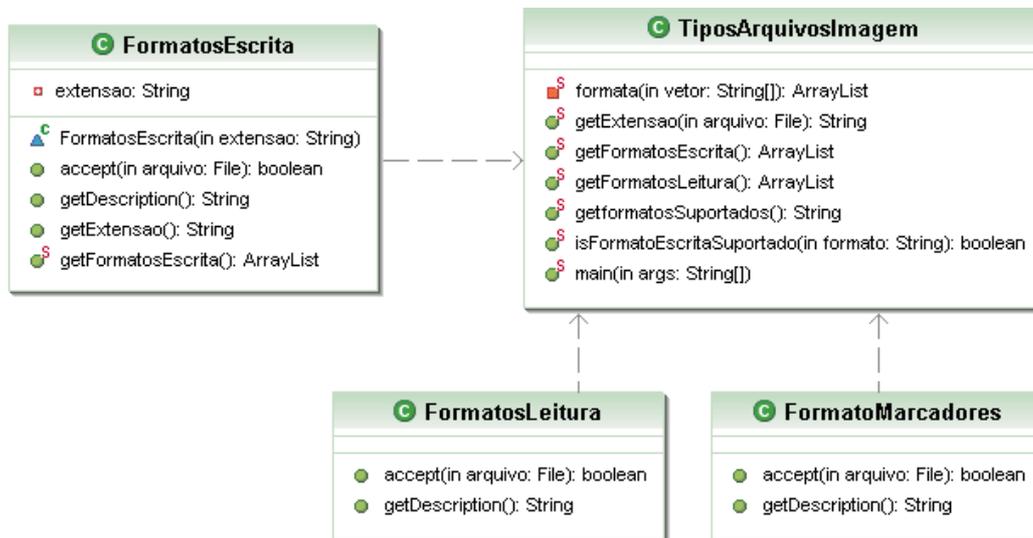


Figura 21: Pacote Formatos.

### A.4 Pacote GUI

Formado pelas classes e interface que representam a interface gráfica com o usuário.

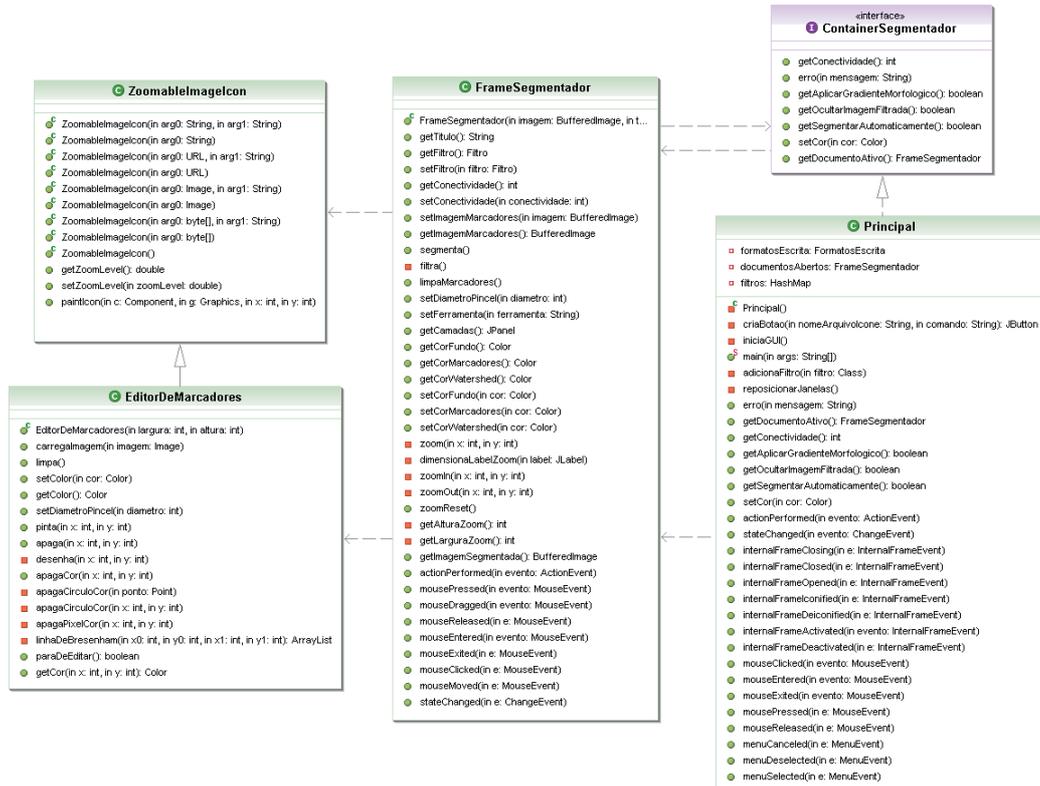


Figura 22: Pacote GUI.

### A.5 Pacote Util

Formado pelas classes utilitárias do sistema, contendo os algoritmos de watershed e rotulação de marcadores, entre outros.

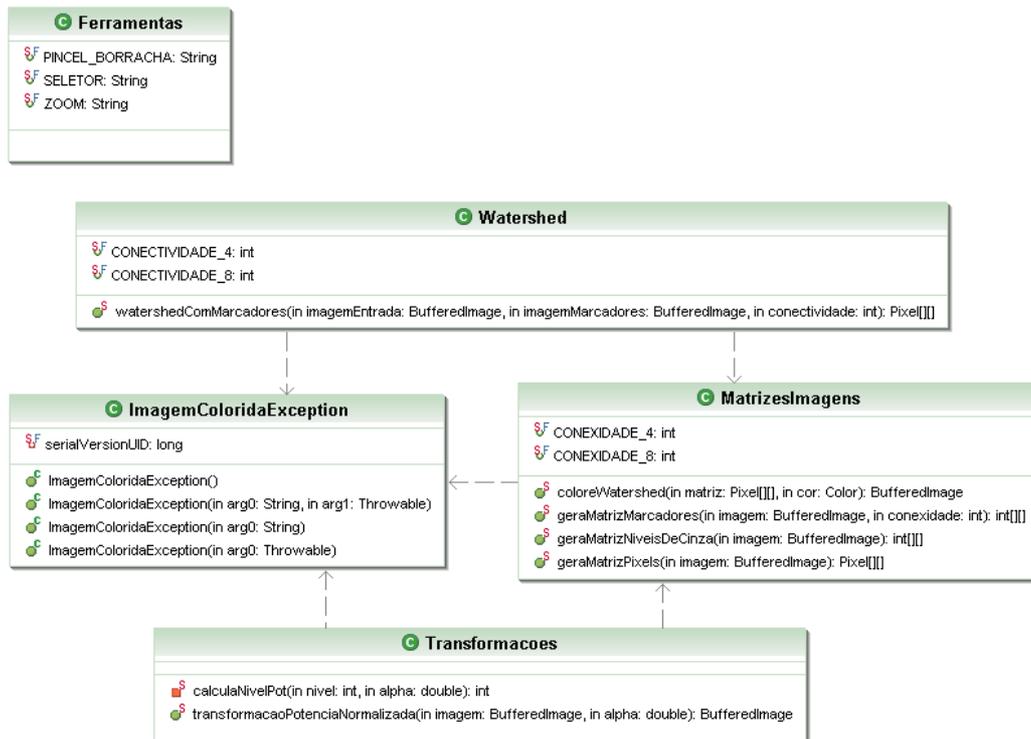


Figura 23: Pacote Util.

## Referências

- [Com] Computer Vision Group of Berkeley University of California. *The Berkeley Segmentation Dataset and Benchmark*. <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>.
- [DL03] E. R. Dougherty and R. A. Lotufo. *Hands-on Morphological Image Processing*. SPIE Press, 2003.
- [Sun] Sun Microsystems, Inc. *Java™ 2 Platform Standard Edition 5.0 API Specification, Package javax.imageio*. <http://java.sun.com/j2se/1.5.0/docs/api/javax/imageio/package-summary.html>.
- [VS91] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, june 1991.