

Trabalho 3 – Segmentação de Imagens

SCC0221 - Introdução à Ciência de Computação I

Tema: Arquivos, Alocação Dinâmica e Recursão

Professores: Moacir Ponti e Rudinei Goularte

Monitores/PAE: André Luis M. Fakhoury, Gabriel Van Loon, Henrico Brum, Lucas T. da Silva e Tamires T. S. Barbieri

Depois do sucesso alcançado com novos filtros – que transformam o rosto do usuário para que fique igual ao de um bebê – lançados por uma empresa concorrente, a empresa em que você trabalha decidiu também lançar uma coleção de filtros para seu aplicativo com a intenção de competir e recuperar alguns usuários. Apesar de você ainda fazer parte do time de estagiários da empresa, será sua a responsabilidade de implementar um dos filtros, que deve deixar a imagem capturada pela câmera com aspecto de desenho.

Neste projeto, você será responsável por implementar uma técnica de segmentação para obter os pixels de bordas de uma imagem. A tarefa de segmentação de imagem consiste em selecionar, a partir de uma posição inicial de uma imagem, um conjunto de pixels que respeitam um determinado critério. A Figura 1 ilustra o resultado final do processo de obtenção de borda após converter uma imagem para a escala cinza.



Figura 1: Processo para segmentar e obter as bordas de uma imagem.

Para facilitar o seu trabalho, você já receberá as imagens convertidas para a escala cinza. A técnica que você irá implementar para segmentação de imagens consiste em um algoritmo de conquista de regiões que se inicia em uma posição da imagem definida pelo toque do usuário na tela de seu smartphone. A partir de uma determinada posição, a busca deverá incluir o pixel dessa posição na região segmentada e avaliar os 4 pixels adjacentes (**os vizinhos na diagonal não são considerados**), respeitando a ordem de visita: **cima, direita, baixo e esquerda**. A busca deve proceder para um vizinho caso a diferença entre os valores do pixel vizinho e da média dos valores dos pixels que já pertencem à região segmentada no momento for menor ou igual ao critério de conquista pré-estabelecido. Após finalizar a busca em um pixel, esta deve retroceder ao pixel anterior e seguir para o próximo vizinho que satisfaz o critério (na ordem definida), ou retroceder ao pixel que a chamou.

O algoritmo deve ser executado para cada consulta do usuário, que poderá formar uma nova região segmentada na imagem. Ao final, você possuirá a imagem segmentada em regiões e então deverá selecionar apenas os pixels que fazem parte da(s) borda(s) dessas. Um pixel pertencerá à(s) borda(s) da imagem se algum de seus 4 vizinhos – **desconsiderando**

diagonais – não pertencer à mesma região segmentada que ele.

Formato do arquivo de entrada

A primeira linha da entrada possui um conjunto de caracteres que representa o nome do arquivo (em formato PGM) da imagem a ser processada. Em seguida, existe uma linha que contém um inteiro K ($1 \leq K \leq 100$) indicando a quantidade de consultas que serão feitas pelo usuário para a imagem da entrada. Então, seguem K linhas com três valores inteiros: as coordenadas X e Y da posição em que o usuário tocou na imagem (onde o processo de busca deve iniciar) e o valor de critério C a ser considerado na segmentação ($0 \leq X < M$, $0 \leq Y < N$ e $0 \leq C \leq 255$).

Formato do arquivo PGM

A primeira linha de um arquivo PGM possui a string “P2” que identifica o tipo de arquivo e deve ser ignorada. A segunda linha do arquivo possui dois valores inteiros N e M que representam as dimensões da imagem (largura e altura, respectivamente). A terceira linha do arquivo possui apenas um valor inteiro e também deve ser ignorada. Após isso, o arquivo contém $N \times M$ inteiros $I_{i,j}$ ($0 \leq I_{i,j} \leq 255$, $0 \leq i < M$ e $0 \leq j < N$) e cada $I_{i,j}$ indica a intensidade do pixel (i, j) da imagem em tons de cinza. Repare que os N primeiros valores são referentes aos pixels da primeira linha da imagem. **A imagem deve ser armazenada na memória de maneira dinâmica!**

Mais informações sobre esse formato de arquivo podem ser encontradas em: <http://netpbm.sourceforge.net/doc/pgm.html>.

Formato da saída

A saída deve possuir, em cada linha, as coordenadas (*linha*, *coluna*) de cada pixel que pertence à(s) borda(s) da(s) regiões segmentadas da imagem formadas após concluir todas as consultas da entrada. Os valores das coordenadas devem ser apresentados na seguinte ordem:

- coordenadas com menor índice de linha primeiro;
- se houver empate no primeiro critério, considerar as coordenadas com menor índice de coluna primeiro.

Exemplo de entrada 1

```
teste1.pgm
3
0 0 5
5 4 2
4 4 10
```

Arquivo teste1.pgm

```
P2
5 6
255
10 13 18 12 11
11 10 17 10 19
16 17 19 11 12
18 18 17 16 17
13 12 19 13 15
16 10 11 19 14
```

Saída para o exemplo de entrada 1

```
(0, 1)
(0, 2)
(1, 1)
(1, 2)
(2, 0)
(2, 1)
(2, 2)
(2, 3)
(3, 0)
(3, 1)
(3, 2)
(3, 3)
(3, 4)
(4, 2)
(4, 3)
(4, 4)
(5, 3)
(5, 4)
```

Comentário

A primeira consulta possui critério 5 e inicia na posição $(0, 0)$, que não pertence a nenhuma outra região e, então, é adicionada à primeira região segmentada. Não há vizinho acima, logo avalia-se o vizinho a direita $(0, 1)$. Como a diferença entre o valor do pixel $(0, 1)$ e a média dos valores dos pixels que pertencem à região segmentada é menor que o critério ($|13 - media(10)| \leq 5$), então a busca segue para o pixel $(0, 1)$. Adiciona-se ele na região segmentada e avalia-se os vizinhos. Novamente, não há vizinho acima, o vizinho a direita não satisfaz o critério ($|18 - media(10, 13)| \not\leq 5$), mas o vizinho abaixo $(1, 1)$ satisfaz ($|10 - media(10, 13)| \leq 5$) e a busca segue para ele. Então, adiciona-se o pixel $(1, 1)$ na região segmentada e avalia-se os vizinhos. O vizinho acima já pertence à uma

região, a direita não satisfaz o critério, abaixo também não satisfaz e a esquerda satisfaz ($|11 - \text{media}(10, 13, 10)| \leq 5$), logo a busca segue para o pixel (1, 0). Este é adicionado à região e analisa-se seus vizinhos. Os vizinhos acima e a direita deste já pertencem a uma região, mas o abaixo (2, 0) não pertence e satisfaz o critério ($|16 - \text{media}(10, 13, 10, 11)| \leq 5$), então a busca segue para ele. Adiciona-se à região e analisa-se os vizinhos: acima já pertence a uma região e a direita (2, 1) satisfaz o critério ($|17 - \text{media}(10, 13, 10, 11, 16)| \leq 5$), a busca segue para este. O pixel (2, 1) é adicionado na região e os vizinhos são analisados: acima já pertence a uma região, a direita não satisfaz o critério, abaixo também não e a esquerda já pertence a uma região. A busca no pixel (2, 1) é finalizada e retrocede para o pixel (2, 0). Restava analisar o pixel abaixo que não satisfaz o critério ($|18 - \text{media}(10, 13, 10, 11, 16, 17)| \not\leq 5$) e a esquerda que não existe. Então, a busca no pixel (2, 0) é finalizada e retrocede para o pixel (1, 0). Analisa-se o pixel a esquerda que não existe, finaliza a busca neste pixel e retrocede para o pixel (1, 1). Este pixel já havia analisado todos seus vizinhos, então apenas retrocede a busca para o pixel (0, 1). Faltava analisar o pixel a esquerda deste, que já pertence a uma região, então a busca é finalizada e retrocede para o pixel (0, 0). Deste pixel, ainda era preciso analisar o pixel abaixo, que já pertence a uma região, e a esquerda, que não existe. Logo, a busca é finalizada e toda a primeira consulta também.

Na segunda consulta, adiciona-se os pixels na segunda região segmentada na seguinte ordem: (5, 4), (4, 4), (4, 3) e (3, 3). Como o pixel inicial da terceira consulta já pertence à segunda região segmentada, a busca nem começa.

Ao final, os pixels ficarão divididos da seguinte forma:

```
1 1 X X X
1 1 X X X
1 1 X X X
X X X 2 X
X X X 2 2
X X X X 2
```

onde os pixels com 1 foram conquistados na primeira consulta, com 2 foram conquistados na segunda e com X não foram conquistados. É trivial agora obter os pixels que pertencem às bordas das regiões formadas.

Exemplo de entrada 2

```
teste2.pgm
2
0 0 20
4 3 15
```

Arquivo teste2.pgm

```
P2
6 6
255
140 121 194 185 181 195
```

```
135 124 148 126 197 184
169 151 131 191 183 111
187 160 154 127 176 109
152 139 143 162 167 124
143 126 188 176 155 126
```

Saída para o exemplo de entrada 2

```
(0, 1)
(0, 2)
(0, 3)
(1, 0)
(1, 2)
(1, 3)
(1, 4)
(2, 0)
(2, 1)
(2, 2)
(2, 3)
(2, 4)
(2, 5)
(3, 1)
(3, 2)
(3, 3)
(3, 4)
(3, 5)
(4, 2)
(4, 3)
(4, 4)
(4, 5)
(5, 2)
(5, 3)
(5, 4)
```

Comentário

Na primeira consulta, os pixels são adicionados na primeira região segmentada na seguinte ordem: (0, 0), (0, 1), (1, 1), (1, 2), (1, 3), (2, 2), (2, 1) e (1, 0). Na segunda consulta, os pixels são adicionados na segunda região segmentada na seguinte ordem: (4, 3), (4, 4), (3, 4), (2, 4) e (5, 3).

Ao final, os pixels ficarão divididos da seguinte forma:

```
1 1 X X X X
1 1 1 1 X X
X 1 1 X 2 X
X X X X 2 X
X X X 2 2 X
X X X 2 X X
```

Instruções

O código fonte (apenas arquivo .c) deve ser entregue pelo <http://run.codes>.

IMPORTANTE: Não serão aceitos trabalhos que utilizem estruturas de repetição para realizar a etapa de processamento (busca) do problema. Também não serão aceitos trabalhos que utilizem alocação estática de memória para arranjos e matrizes.

O trabalho será avaliado levando em consideração:

1. realização dos objetivos;
2. representação correta da entrada e saída de dados;
3. uso de comentários e estrutura do código (ex.: indentação, legibilidade, organização);
4. número de acertos no sistema run.codes.