Universidade Federal de Minas Gerais
Departamento de Ciência da Computação

DCC004 – Algoritmos e Estruturas de Dados II

Profs. Cristiano Arbex Valle e Gisele L. Pappa

Trabalho prático 1
Segmentação de imagens

Valor: 15 pontos
Entrega: 06/05/2018

1 Descrição

Câmeras portáteis ou de monitoramento já são parte da nossa vida cotidiana. A câmera do seu celular é utilizada para fotografar o conteúdo dos exercícios da sala de aula e as câmeras de monitoramento do CAD são capazes de mostrar seus passos nas última semanas. Algoritmos de análise automática de imagens são cada vez mais comuns, e podem ser utilizados para reconhecer uma pessoa específica em uma imagem ou contar o público de um show. Esses algoritmos de análise automática normalmente começam com uma mesma tarefa: segmentar a imagem de acordo com diferentes regiões.

Considerando que uma imagem digital é representada por uma matriz, que contém um valor de intensidade para cada célula (pixel), o objetivo da segmentação é particionar os pixels de uma imagem em grupos, geralmente relacionados a um objeto ou região da imagem, de forma a simplificar sua representação e facilitar sua análise. Os pixels contidos em uma mesma região compartilham alguma característica em comum, como a cor ou o valor de intensidade. Mais especificamente, o processo de segmentação atribui um rótulo a cada pixel da imagem, identificando a qual grupo ele pertence.

A Figura 1 mostra uma aplicação simples de contagem de moedas utilizando segmentação, onde as moedas foram separadas do fundo e os pixels referentes a cada moeda constituem um grupo (região) diferente (representados na figura da direita por cores diferentes).

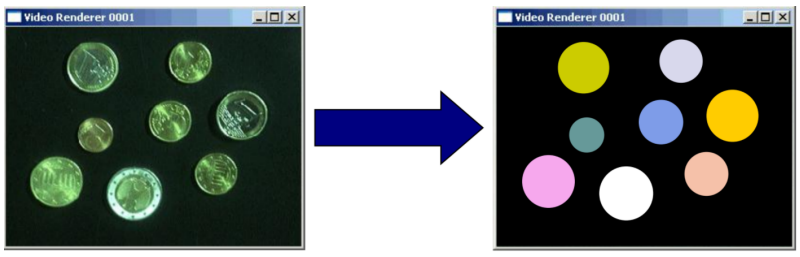

Figura 1: Contagem de moedas.

Neste TP, iremos implementar um algoritmo de segmentação denominado algoritmo de crescimento de regiões. Este algoritmo define um grupo a partir de um pixel aleatório inicial, denominado semente, e expande o grupo para os vizinhos do pixel semente de acordo com um critério baseado na intensidade do pixel. Para um dado pixel $I(u, v)$, sua vizinhança 4-conexa é dada pelos pixels $I(u, v + 1)$, $I(u + 1, v)$, $I(u, v - 1)$ e $I(u - 1, v)$. Se a diferença de intensidade entre dois pixels vizinhos for pequena, o grupo se propaga. Por exemplo, se $|I(u, v) - I(u, v + 1)| \leq T$, então (u, v) e $(u, v + 1)$ pertencem ao mesmo grupo (região), onde T é um valor de limiar predefinido.

2 Formato das Imagens

Utilizaremos neste TP os formatos de imagem PGM e PPM. O formato PGM (ASCII, P2) representa imagens em tons de cinza com 8 bits de profundidade (ou seja, os valores de cada pixel variam de 0 a 255). Imagens PPM, por sua vez, armazenam o valor dos pixels com 8 bits para o canal Vermelho, 8 bits para o canal Verde e 8 bits para o canal Azul.

Tanto imagens .pgm e .ppm são armazenadas como um arquivo texto:

- A primeira linha representa a codificação utilizada.
- A segunda indica o tamanho da imagem (resolução).
- A terceira linha indica quantos níveis de intensidade existem na imagem.
- Seguem então os valores dos pixels em suas respectivas posições na imagem.

As Figuras 2 e 3 mostram um exemplo de codificação .pgm e a imagem correspondente, enquanto as Figuras 4 e 5 mostram um exemplo de codificação .ppm. Mais detalhes sobre o formato em: `http://en.wikipedia.org/wiki/Netpbm_format`.

```
P2
24 7
15
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  3  3  3  3  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15 15 15 15  0
0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0 15  0
0  3  3  3  0  0  0  7  7  7  0  0  0 11 11 11  0  0  0 15 15 15 15  0
0  3  0  0  0  0  0  7  0  0  0  0  0 11  0  0  0  0  0 15  0  0  0  0
0  3  0  0  0  0  0  7  7  7  7  0  0 11 11 11 11  0  0 15  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Figura 2: Arquivo .pgm.

Figura 3: Visualização da imagem correspondente.

Caso seu visualizador de arquivos não suporte abrir arquivos .pgm e .ppm, você pode abrir as imagens usando este link: `http://paulcuth.me.uk/netpbm-viewer/`.

3 Objetivo

O objetivo desse trabalho é implementar um programa que lê uma imagem .pgm e segmenta suas regiões utilizando um algoritmo de crescimento de regiões. Avaliaremos seu programa em um conjunto de imagens que será disponibilizado para teste.

Seu programa deve:

- Ler a imagem .pgm indicada na chamada do programa e seu respectivo arquivo auxiliar.

```
P3
3 2
255
255   0   0     0 255   0     0   0 255
255 255   0   255 255 255     0   0   0
```

Figura 4: Arquivo .ppm.

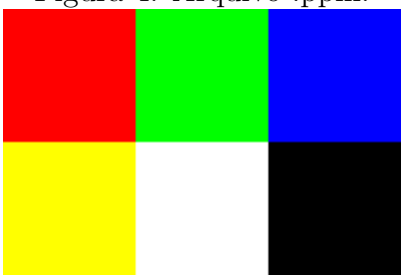

Figura 5: Visualização da imagem correspondente.

- Implementar o algoritmo de crescimento de regiões utilizando uma lista, fila ou pilha (o algoritmo NÃO deve ser recursivo).
- Atribuir a cor indicada no arquivo auxiliar às regiões.
- Salvar a saída em uma imagem no formato .ppm (ASCII, P3).

Acompanhando cada imagem, será fornecido um arquivo auxiliar contendo o número de sementes a ser utilizado para aquela imagem (Não utilizaremos uma semente aleatória, e sim a provida nesse arquivo auxiliar), o valor do limiar T, suas posições e a cor que identificará a respectiva região segmentada. O arquivo auxiliar terá o mesmo nome do arquivo da imagem, com extensão .txt, e terá o seguinte formato:

```
2
10
50,50, <255,0,0>
20,30, <0,0,255>
```

A primeira linha indica o número de sementes. A segunda linha indica o valor do limiar T. Cada uma das linhas seguintes contêm a posição x,y do pixel semente (no exemplo, a primeira semente está na posição 50,50) seguido do código <R,G,B> da cor que a região que contêm aquela semente deve receber. Por exemplo, no caso da semente 50,50, a região a qual ela pertence deve ser pintada de vermelho. Já a região da semente da posição 20,30 deve receber a cor azul.

4 Exemplos

A figura 6 mostra alguns exemplos de imagem e suas respectivas regiões segmentadas. Note que cada cor representa uma região diferente.

5 Indo além... (+2 pontos extra)

Modifique seu algoritmo para, ao invés de ler o arquivo auxiliar, ele detectar automaticamente o número de regiões utilizando sementes aleatórias.

Reimplemente o algoritmo de crescimento de regiões utilizando uma função recursiva. Reescreva apenas a função de segmentação.

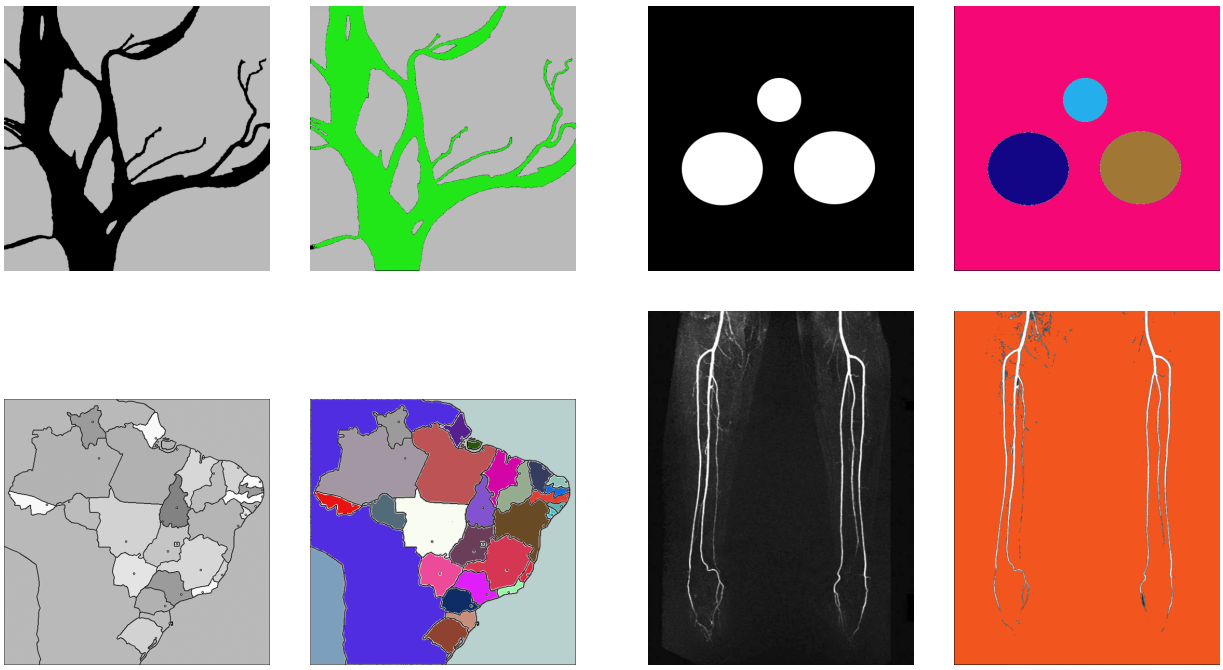

Figura 6: Exemplos de imagem e suas regiões segmentadas.

6 O que deverá ser entregue

Você deve submeter uma documentação de até 10 páginas contendo uma descrição da solução proposta, detalhes relevantes da implementação, além de uma análise de complexidade de tempo dos algoritmos envolvidos. A documentação deve conter:

1. Introdução: descrição sucinta do problema a ser resolvido e visão geral sobre o funcionamento do programa.

2. Implementação: descrição sobre a implementação do programa. Devem ser detalhados as estruturas de dados utilizadas (de preferência com diagramas ilustrativos) e o funcionamento das principais funções.

3. Estudo de complexidade: estudo da complexidade do tempo de execução dos procedimentos implementados.

4. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação. Faça também uma análise dos resultados para as diferentes imagens, dissertando se os resultados foram bons ou ruins e o porquê.

5. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet, se for o caso.

7 Submissão

Para submeter o trabalho, envie pelo Moodle um arquivo `.zip` ou `.rar` contendo apenas o código (arquivos `.c` e `.h`), além de um arquivo **pdf** com a documentação. **Não inclua o executável**. Ao receber seu código, nós compilaremos seu programa em um ambiente Linux com o seguinte comando:

```
gcc -Wall -std=c99 -lm *.c -o tp1
```

Em seguida, executaremos seu programa com o seguinte comando:

```
./tp1 nome_da_imagem
```

1. Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
2. Clareza, indentação e comentários no programa também serão avaliados.
3. O trabalho é individual.
4. Trabalhos copiados, comprados, doados, etc. serão penalizados severamente.
5. Penalização por atraso: $(2^d - 1)$ pontos, onde d é o número de dias de atraso.