Busca de Regiões em Imagens

1 Considerações Iniciais

Neste trabalho, você deverá implementar em Linguagem C dois algoritmos **independentes** para resolver o problema a seguir.

- Algoritmo 1: Selecionar aleatoriamente, filtrar e salvar regiões (sub-imagens) de uma imagem fornecida pelo usuário.
- Algoritmo 2: Ler o diretório de imagens criado pelo Algoritmo 1 e localizar as sub-imagens em uma imagem fornecida pelo usuário.

2 Algoritmo 1

Seguem os requisitos:

1. Receber uma imagem I de dimensões n x m do usuário no formato pgm binária ou texto. Ver Seção 4.
2. Gerar uma quantidade t de sub-imagens a partir de pontos pseudo-aleatórios de I. As sub-imagens devem ter dimensão p x q especificadas pelo usuário. Todas as sub-imagens devem ter a mesma dimensão espacial, ou seja, p x q.
3. Cada sub-imagem deve ser processada pelo filtro da média com janela 3 x 3 e tratamento de bordas com zero. Detalhes em `http://www.ic.uff.br/~aconci/suavizacao.pdf` e `https://sandaminimadhushika96.medium.com/mean-filter-in-image-processing-66fe7bb733e4`.
4. As sub-imagens filtradas devem ser salvas em um diretório D especificado pelo usuário. As sub-imagens devem ser salvas em formato pgm (texto ou binário), ver Seção 4.

3 Algoritmo 2

Seguem os requisitos:

1. Permitir ao usuário informar o diretório das sub-imagens a serem buscadas e a imagem I, na qual será realizada a busca.
2. Dado uma sub-imagem, identificar a região correspondente em I com base em um critério de similaridade ou dissimilaridade. Sugestão: pesquise sobre erro médio quadrático (em inglês, *mean square error* (MSE)).
3. Reportar o tempo total para efetivar a busca de todas as sub-imagens existentes no diretório.
4. Salvar um arquivo de texto com as coordenadas em I onde a sub-imagem foi localizada. Considere a coordenada do primeiro pixel (superior esquerdo) da região corresponde à sub-imagem.
5. O arquivo de texto com as coordenadas dos blocos recuperados deve ter o seguinte formato:

   nome_da_subimagem_1, coord_x1, coord_y1
   nome_da_subimagem_2, coord_x2, coord_y2
   ...
   nome_da_subimagem_t, coord_x1, coord_y2

   , onde t corresponde à quantidade de sub-imagens salvas pelo o Algoritmo 1.

Nas seções seguintes deste documento estão descritas algumas informações necessárias para o correto desenvolvimento deste trabalho.

4 Formato PGM

Neste trabalho, você deve utilizar o formato PGM (*portable graymap*) para armazenar imagens em arquivos. Este formato tem duas variações, uma binária (o PGM "normal" ou *raw*) e outra textual (o PGM ASCII ou *plain*). Em ambos os casos, o arquivo deve conter um cabeçalho e a matriz correspondente à imagem. O exemplo a seguir mostra um arquivo PGM textual:

```
P2
5 4
16
9 4 5 0 8
10 3 2 1 7
9 1 6 3 15
1 16 9 12 7
```

A primeira linha do arquivo contém obrigatoriamente uma palavra-chave, que deve ser "P2" no caso de um arquivo PGM textual e "P5" no caso de um arquivo PGM binário. A segunda linha contém dois números inteiros que indicam o número de colunas e o número de linhas da matriz, respectivamente. A terceira linha contém um número inteiro positivo *maxval*, que deve ser igual ao maior elemento da matriz. Na definição do formato PGM, *maxval* não pode ser maior que 65535. Para fins deste trabalho, entretanto, *maxval* é no máximo 255. Os demais números do arquivo são os elementos de uma matriz de inteiros com os tons de cinza de cada ponto da imagem. Cada tom de cinza é um número entre 0 e *maxval*, com 0 indicando "preto" e *maxval* indicando "branco".

O formato PGM também permite colocar comentários. Todo o texto que vai desde um caractere '#' até (e inclusive) o próximo fim de linha é um comentário e deve ser ignorado. Este é um exemplo de arquivo PGM textual com um comentário:

```
P2
# feep.pgm
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 0 3 3 3 0 0 7 7 7 7 0 0 11 11 11 0 0 15 15 15 0 0 0
0 0 0 3 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 0 0 3 0 0 0 7 7 7 0 0 0 11 0 0 0 0 0 15 15 15 0 0
0 0 0 3 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 0 3 3 3 0 0 7 0 0 0 0 0 11 11 11 0 0 15 15 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
```

O formato PGM binário tem cabeçalho análogo ao do PGM textual, usando a palavra chave "P5" em vez da "P2". O que muda é o modo como é armazenada a matriz de tons de cinza. No formato PGM textual, essa matriz é guardada como uma sequência de caracteres ASCII contendo as representações decimais das tonalidades de cinza. No formato PGM binário, a matriz é guardada como uma sequência de *bytes*, sendo o valor de cada *byte* (de 0 a 255) uma tonalidade de cinza. O número de *bytes* da sequência é exatamente igual ao número de elementos da matriz[1]. Este é um exemplo de arquivo PGM binário:

[1]Na verdade, essa descrição se aplica apenas a arquivos PGM com $maxval \leq 255$, que são considerados neste trabalho. No caso $256 \leq maxval \leq 65535$, a matriz é guardada como um sequência de pares de *bytes* e o número de *bytes* da sequência é o dobro do número de elementos da matriz.

```
P5
# feep.pgm
24 7
15
```

... (sequência de 24 x 7 bytes com as tonalidades de cinza)

Existem dois outros formatos de arquivo muito semelhantes ao PGM: o PBM (*portable bitmap*), para imagens monocromáticas (só preto e branco, sem tons de cinza), e o PPM (*portable pixmap*), para imagens coloridas. No primeiro, os elementos da matriz podem assumir apenas os valores 0 e 1. No segundo, os elementos da matriz são triplas de números inteiros positivos correspondentes às intensidades das cores vermelha, verde e azul nos pontos da imagem. Quem quiser saber mais detalhes sobre esses formatos, visite as seguintes páginas:

- http://en.wikipedia.org/wiki/Portable pixmap
- http://netpbm.sourceforge.net/doc/pbm.html
- http://netpbm.sourceforge.net/doc/pgm.html
- http://netpbm.sourceforge.net/doc/ppm.html

5 Organização dos códigos fontes e Avaliação

ATENÇÃO:

1. A nota máxima deste trabalho é 10.0. A distribuição dos pontos segue como definido a seguir:

   - **Implementação do Algoritmo 1:** Pontuação: 4.0

     Sendo:
     - Gravação das sub-imagens: 2.0 pontos.
     - Filtro da Média: 2.0 pontos.

   - **Implementação do Algoritmo 2:** Pontuação: 4.0

     Sendo:
     - Busca das sub-imagens: 3.0 pontos.
     - Arquivo de coordenadas: 1.0 ponto.

   - **Organização do código:** Pontuação: 2.0

   Critérios para cada tópico acima:

   - Domínio da solução adotada: 50%
   - Análise Técnica: 50%

2. Todos os arquivos fontes do seu programa devem conter um cabeçalho como o seguinte:

```
/*************************************************/
/* Aluno: Joao de Souza
/* Matricula: 12345
/* Avaliacao 04: Trabalho Final                  */
/* 04.505.23 - 2023.1 - Prof. Daniel Ferreira    */
/* Compilador:...(DevC++ ou gcc) versao ...      */
/*************************************************/
```

A organização do programa deve obedecer a construção de arquivos de cabeçalhos e permitir a compilação separada. Deve-se utilizar processo de *linkedição*. Procure dividir seus códigos em arquivos sempre que necessário. Enfim, procure construir bibliotecas genéricas que proporcione posterior reuso.

3. O trabalho deverá ser realizado em **equipes com no máximo 4 (quatro) participantes**. **Nota individual.** Sugiro, portanto, que marquem seminários entre a equipe para que todos possam atingir o mesmo nível de conhecimento.

4. Códigos fontes copiados (com ou sem eventuais disfarces) receberão nota **ZERO**.

5. Programas que não executam receberão nota **ZERO**.

6. Trabalhos atrasados serão penalizados. A regra é a seguinte: para cada dia de atraso o aluno perde **25%** da nota.

7. É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos dos programas (conforme estudado em aula). O não obediência a este item poderá resultar em perda de até 15% da nota obtida.

8. O professor poderá executar o programa tantas vezes quantas forem necessárias para testar os vários casos possíveis para as entradas.

9. Os códigos devem ser disponbilizados ao professor por meio de um link no GitHub. Deve ser construído um README com detalhes para a execução.

10. No dia da apresentação é **indispensável** o comparecimento de toda a equipe. Faltas devem ser devidamente justificadas para projeto de segunda-chamada.

11. Datas para as apresentações:

    - **Dia 1: 21 de Junho de 2023 a partir das 07h40.**

    - **Dia 2: 26 de Junho de 2023 a partir das 07h40**.

A ordem de apresentação será definida posteriormente por sorteio.