

MAC0110 – Introdução à Computação

TURMA: MATEMÁTICA APLICADA E COMPUTACIONAL

Quarto Exercício-Programa

Entrega: até 2 de julho de 2014

Um pequeno editor de imagens

Um dos objetivos deste exercício-programa é realizar algumas transformações em imagens armazenadas em arquivos no formato PGM. Este formato, que é utilizado para imagens em níveis de cinza, será usado neste programa pela sua simplicidade. Cada ponto (pixel) que compõe uma imagem é representado por um inteiro não-negativo que corresponde a uma tonalidade de cinza. Os extremos dessa escala de tons de cinza são o número zero, que representa o "preto", e o número 255, que representa o "branco". Uma imagem é representada por uma matriz de números inteiros não-negativos que correspondem a um nível de cinza de cada ponto. Os pontos de uma imagem são indexados a partir do canto superior esquerdo, da esquerda para a direita, e de cima para baixo. Assim, os pontos numa imagem de 200 linhas e 300 colunas estão indexados da seguinte forma:

```
(0, 0)    (0, 1)    ... (0, 299)
(1, 0)    (1, 1)    ... (1, 299)
...       ...       ...
(199, 0)  (199, 1)  ... (199, 299)
```

É sobre a matriz (chamada de matriz de trabalho) que representa uma imagem, que o programa deve aplicar uma das transformações: rotação, rebater, negativo ou suavizar ruídos.

1 Funções de Transformação

Nesta seção descrevemos as funções de transformação de imagens que devem ser implementadas neste exercício-programa.

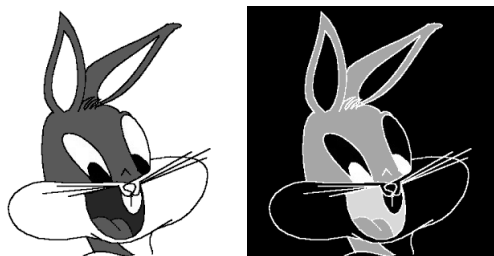
1.1 Funções de edição**1.1.1 Negativo**

Figura 1: Negativo: pernalonga-n.pgm

Fazer o negativo de uma imagem consiste em "inverter" os valores dos pixels; ou seja, os valores próximos de "branco" se transformam em próximos de "preto" e vice-versa. Se uma imagem está representada na matriz a , para obter o seu negativo basta substituir cada elemento $a[i][j]$ por $\text{maior_valor} - a[i][j]$, onde maior_valor é o maior inteiro na matriz a .

1.1.2 Rebatimento



Figura 2: Rebater na horizontal e na vertical: pernalonga-h.pgm e pernalonga-v.pgm

Seu programa deve rebater uma imagem na horizontal ou na vertical.

1.1.3 Rotação

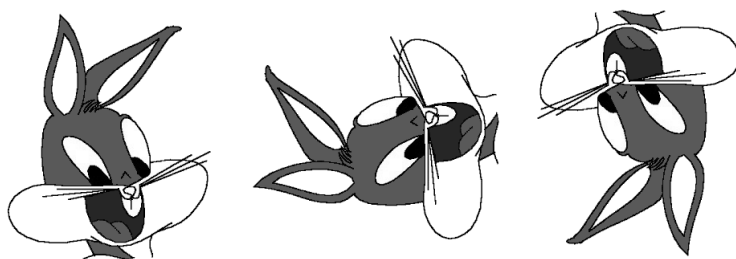


Figura 3: Rotações (duas): pernalonga-r.pgm e pernalonga-r-r.pgm

Seu programa deve fazer a rotação da imagem em 90 graus no sentido anti-horário. Tenha atenção especial no caso de imagens retangulares.

1.2 Funções de filtro

Filtros são tranformações com a finalidade de melhorar uma imagem, dar mais nitidez, etc..

Todas as funções de filtro seguem o mesmo princípio. Cada ponto (i, j) da imagem resultante da aplicação de um filtro é uma função dos valores de uma vizinhança desse ponto.

Seja A uma matriz de inteiros não-negativos com m linhas e n colunas, e seja p um número inteiro positivo ímpar. Dada uma coordenada (i, j) em A , a vizinhança de tamanho p em torno de (i, j) é a submatriz A_{ij} de A com p linhas, p colunas e centro em (i, j) .

Por exemplo, dada a seguinte matriz 5×5

$$\begin{bmatrix} 9 & 4 & 5 & 0 & 8 \\ 10 & 3 & 2 & 1 & 7 \\ 9 & 1 & 6 & 3 & 15 \\ 0 & 3 & 8 & 10 & 1 \\ 1 & 16 & 9 & 12 & 7 \end{bmatrix}$$

a vizinhança de tamanho 3 em torno de $(1, 1)$ é a submatriz

$$\begin{bmatrix} 9 & 4 & 5 \\ 10 & 3 & 2 \\ 9 & 1 & 6 \end{bmatrix}$$

Note que para algumas coordenadas (por exemplo, para as bordas da imagem), a vizinhança será menor. Por exemplo, a vizinhança de tamanho 3 em torno de $(0, 1)$ é a submatriz

$$\begin{bmatrix} 9 & 4 & 5 \\ 10 & 3 & 2 \end{bmatrix}$$

Neste programa vamos utilizar apenas o filtro da mediana que definimos a seguir.

1.2.1 O filtro da mediana



Figura 4: Imagem original e após filtro da mediana: barco-ruído-m.pgm

O filtro da mediana é uma transformação bastante comum para suavizar ruídos do tipo impulso em sinais e imagens digitais. Cada coordenada de uma matriz, após a aplicação do filtro da mediana, é a mediana dos números numa vizinhança dessa coordenada. No caso da matriz do exemplo anterior, os números na vizinhança de tamanho 3 em torno de $(1, 1)$ são 9, 4, 5, 10, 3, 2, 9, 1, 6. Logo, $mediana(1, 1) = 5$, pois nessa vizinhança existem quatro números menores do que 5 e quatro números maiores do que 5. Para os casos em que a quantidade de números numa vizinhança é par (por exemplo, na vizinhança de algumas coordenadas nas bordas de uma imagem), vamos utilizar o $(k/2 + 1)$ -ésimo elemento da sequência ordenada dos seus k vizinhos. Assim, na matriz do exemplo anterior, os números na vizinhança de tamanho 3 em torno de $(0, 0)$ são 3, 4, 9, 10. Logo, $mediana(0, 0) = 9$, que é o $(k/2 + 1)$ -ésimo elemento, onde $k = 4$. Note que o cálculo da mediana de uma coordenada não deve influenciar o cálculo da mediana de outra coordenada.

No caso da matriz do exemplo acima, o resultado da aplicação do filtro da mediana por uma vizinhança de tamanho 3 é a seguinte matriz:

$$\begin{bmatrix} 9 & 5 & 3 & 5 & 7 \\ 9 & 5 & 3 & 5 & 7 \\ 3 & 3 & 3 & 6 & 7 \\ 3 & 6 & 8 & 8 & 10 \\ 3 & 8 & 10 & 9 & 10 \end{bmatrix}$$

2 Formato dos arquivos

Neste exercício-programa vamos utilizar arquivos no formato PGM (*Portable Gray Map*), que são fáceis de decodificar. Os nomes de tais arquivos possuem a extensão “.pgm”. Segundo este

formato, o arquivo contém um cabeçalho e a matriz correspondente a uma imagem. Veja o exemplo a seguir.

```
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 uma palavra-chave “P2” que é obrigatória. A segunda linha contém dois números inteiros: o primeiro corresponde ao número de colunas e o segundo corresponde ao número de linhas da matriz. A terceira linha contém um número inteiro (*maior_valor*) que é o maior número da matriz. Para fins deste programa, *maior_valor* é no máximo 255. Os demais números do arquivo correspondem aos tons de cinza da imagem, que podem estar armazenados em forma de uma matriz de inteiros. Cada tom de cinza é um inteiro entre 0 e *maior_valor*, com 0 indicando “preto” e *maior_valor* indicando “branco”.

Gere arquivos pequenos para testar seu programa antes de trabalhar com as imagens reais.

Para visualizar as imagens no formato PGM, você pode usar algum programa, como por exemplo o *IrfanView*, para Windows, que pode ser obtido no sítio <http://www.irfanview.com/>.

O programa GIMP (GNU Image Manipulation Program), para Linux, é um transformador de imagens bastante completo, que implementa as transformações pedidas neste programa e inúmeras outras. Veja mais informações em <http://www.gimp.org/>.

Um outro programa que você pode utilizar, para Linux ou Windows, é o *ImageMagick*, que pode ser obtido no sítio <http://www.imagemagick.org/>.

A seguir dois links sobre o formato PGM.

- (1) (PGM faz parte da família PNM)
http://en.wikipedia.org/wiki/Netpbm_format
- (2) (PGM pode ser ASCII ou binário; nós estamos usando ASCII)
<http://people.sc.fsu.edu/~jburkardt/data/pgma/pgma.html>

3 O que o seu programa deve fazer

Inicialmente, seu programa deve solicitar ao usuário que digite o nome de um arquivo que armazena uma imagem no formato PGM. Se for possível abrir esse arquivo, o programa deve ler todos os dados desse arquivo e, em seguida, escrever na tela as informações da imagem (matriz) correspondente. A seguir, seu programa deve exibir na tela uma lista com as possíveis funções de transformação de imagens, com os respectivos códigos, e solicitar ao usuário que escolha o código de uma dessas funções. Seu programa deve aplicar a função de transformação na matriz de trabalho, e escrever na tela as informações da imagem (matriz) transformada. Depois, deve criar um arquivo no formato PGM com a imagem transformada. O nome desse arquivo deve ser o nome do arquivo com a imagem original (mas, sem “.pgm”), seguido por “-”, seguido pelo código da função de transformação aplicada na imagem, e seguido por “.pgm”.

Por exemplo, se o nome do arquivo com a imagem original é “xyz.pgm”, o arquivo a ser criado para a imagem transformada pela função de rotação deverá ter o nome “xyz-r.pgm”.

A seguir listamos as possíveis funções de transformação com os respectivos códigos.

- n - negativo. Seu programa deve calcular o negativo da imagem atual.
- h - rebater na horizontal. Seu programa deve rebater na horizontal a imagem atual.
- v - rebater na vertical. Seu programa deve rebater na vertical a imagem atual.
- r - rotação. Seu programa deve fazer a rotação de 90 graus da imagem atual, no sentido anti-horário.
- m - filtro da mediana. Seu programa deve aplicar o filtro da mediana na imagem atual, considerando uma vizinhança de tamanho 3.

4 Funções a serem implementadas

Utilize em seu programa as seguintes constantes:

```
#define MAXLIN      400    /* numero maximo de linhas para matrizes */
#define MAXCOL      400    /* numero maximo de colunas para matrizes */
#define MAXVIZ      9      /* numero maximo de vizinhos */
#define MAXNOMEARQ 81      /* tamanho maximo (80) para nome de arquivo */
```

O seu programa deve implementar, **obrigatoriamente**, as funções cujos protótipos estão descritos a seguir. Os protótipos e as definições dessas funções **não** devem ser alterados.

1. void le_arquivo_pgm(char inic_nome_arq[], int a[][MAXCOL], int *lin, int *col, int *maior);
2. void escreve_arquivo_pgm(char inic_nome_arq[], int a[][MAXCOL], int linhas, int colunas, int maior_valor);
3. void escreve_matriz_tela(int a[][MAXCOL], int linhas, int colunas, int maior_valor);
Escreve na tela as informações da imagem representada pela matriz a: número de linhas, número de colunas, maior valor e os elementos da matriz a no formato de matriz.
4. void negativo(int a[][MAXCOL], int linhas, int colunas, int *maior);
5. void rebate_horizontal(int a[][MAXCOL], int linhas, int colunas);
6. void rebate_vertical(int a[][MAXCOL], int linhas, int colunas);
7. void rotacao(int a[][MAXCOL], int *lin, int *col);
8. void filtro_mediana(int a[][MAXCOL], int linhas, int colunas, int *maior);
9. void copia_matriz(int a[][MAXCOL], int b[][MAXCOL], int linhas, int colunas);
Copia os elementos da matriz b para a matriz a.
10. int maior_valor_matriz(int a[][MAXCOL], int linhas, int colunas);
Devolve o maior inteiro da matriz a.
11. int determina_mediana(int a[][MAXCOL], int linhas, int colunas, int i, int j);
Determina e devolve a mediana de todos os inteiros da vizinhança de tamanho 3 em torno da coordenada (i, j) da matriz a. Tome cuidado com os vizinhos inexistentes.
Obs.: Para determinar a mediana, armazene todos os vizinhos num vetor; ordene os inteiros desse vetor, e obtenha a mediana.
12. void ordena(int v[], int n);
Rearranja os n inteiros do vetor v para que fiquem em ordem crescente (ou seja, $v[0] \leq v[1] \leq \dots \leq v[n-1]$).

Observação 1: Consideramos os valores de MAXLIN e MAXCOL como sendo iguais a 400. As imagens provenientes de dispositivos fotográficos modernos podem apresentar tamanhos bem superiores a estes. Mas, por questões de limitações de área de memória alocada para as matrizes dentro de funções na linguagem C, utilizaremos imagens limitadas a 400x400. Para utilizar matrizes maiores seriam necessários recursos mais avançados que não serão vistos nesta disciplina.

Observação 2: Caso uma execução do seu programa seja finalizada por violações na quantidade de memória alocada, verifique se seu programa está utilizando matrizes auxiliares desnecessárias que estejam excedendo o espaço disponível na memória. O programa se comporta bem com até três matrizes de 400x400 na memória, ao mesmo tempo.

5 Exemplo

A seguir apresentamos o que é exibido na tela de execução para cada uma das cinco execuções do programa, exemplificando as cinco possíveis funções de transformação, a partir da imagem armazenada no arquivo “exemplo.pgm”.

Digite o nome de um arquivo sem a extensao .pgm : exemplo

As informacoes da imagem no arquivo exemplo.pgm foram obtidas com sucesso.

Numero de linhas : 7
Numero de colunas : 5
Maior valor : 16

Matriz :
9 4 5 0 8
1 3 10 3 2
1 7 5 7 9
1 6 3 15 2
2 0 3 8 10
1 0 0 1 16
9 12 7 1 2

Lista das possiveis funcoes de transformacao e seus codigos:

n - negativo
h - rebater na horizontal
v - rebater na vertical
r - rotacao
m - filtro da mediana

Digite o codigo da funcao que deseja aplicar : n

Funcao NEGATIVO realizada com sucesso.

Numero de linhas : 7
Numero de colunas : 5
Maior valor : 16

Matriz :
7 12 11 16 8

```

15 13 6 13 14
15 9 11 9 7
15 10 13 1 14
14 16 13 8 6
15 16 16 15 0
7 4 9 15 14

```

O arquivo exemplo-n.pgm (com a imagem transformada) foi criado com sucesso.

...

Digite o codigo da funcao que deseja aplicar : h

Funcao REBATE NA HORIZONTAL realizada com sucesso.

Numero de linhas : 7
Numero de colunas : 5
Maior valor : 16

Matriz :

```

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

```

O arquivo exemplo-h.pgm (com a imagem transformada) foi criado com sucesso.

...

Digite o codigo da funcao que deseja aplicar : v

Funcao REBATE NA VERTICAL realizada com sucesso.

Numero de linhas : 7
Numero de colunas : 5
Maior valor : 16

Matriz :

```

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

```

O arquivo exemplo-v.pgm (com a imagem transformada) foi criado com sucesso.

...

Digite o código da função que deseja aplicar : r

Função ROTACAO realizada com sucesso.

Numero de linhas : 5

Numero de colunas : 7

Maior valor : 16

Matriz :

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

O arquivo exemplo-r.pgm (com a imagem transformada) foi criado com sucesso.

...

Digite o código da função que deseja aplicar : m

Função FILTRO DA MEDIANA realizada com sucesso.

Numero de linhas : 7

Numero de colunas : 5

Maior valor : 10

Matriz :

4	5	4	5	3
4	5	5	5	7
3	3	6	5	7
2	3	6	7	9
1	1	3	3	10
2	2	1	3	8
9	7	1	2	2

O arquivo exemplo-m.pgm (com a imagem transformada) foi criado com sucesso.

6 Outras informações

- Veja o arquivo com um esqueleto do programa, que você deve copiar, e escrever o seu programa a partir dele.
Nesse arquivo, as funções `le_arquivo_pgm` e `escreve_arquivo_pgm` já estão escritas.
- Veja também alguns arquivos com imagens no formato PGM, que você pode utilizar para testar o seu programa.
- Não utilize nenhuma outra função além das funções cujos protótipos foram descritos anteriormente.

- Escreva comentários em cada uma das funções explicando o que ela faz.
- Só utilize os recursos da linguagem C que foram vistos em aula e os que fazem parte deste enunciado.
- Para facilitar os testes a serem feitos durante a correção dos programas, deixe a saída idêntica (no conteúdo e na forma) às dos exemplos do enunciado.
- Não se esqueça de seguir as observações muito importantes que estão em **Informações sobre a entrega de EPs**.