

MAC122 – Princípios de Desenvolvimento de Algoritmos

Segundo Semestre de 2014

Profs. Paulo Miranda e Marco Gerosa

Primeiro Exercício-Programa (EP1) - Parte I

Editor de Imagens

Neste exercício-programa você vai desenvolver um programa de manipulação de imagens, codificadas no formato PGM. Esse formato será utilizado pela sua simplicidade. Nesta primeira parte do EP, cada ponto (pixel) que compõe a imagem é representado por um valor, que representa uma tonalidade de cinza. Os extremos da escala 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, que correspondem a um nível de cinza de cada ponto. É importante destacar que os pontos da imagem são numerados (indexados) a partir do canto superior esquerdo, da esquerda para a direita, de cima para baixo. Assim uma imagem de 200 linhas e 300 colunas possui a indexação ilustrada a seguir:

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

Após lido o arquivo, esta será a representação da imagem com a qual o programa trabalhará. Nessa matriz de trabalho serão aplicadas uma série de transformações, que serão o objeto deste programa. O programa possibilitará aplicar mais um de uma transformação na mesma imagem.

1 Funções de Transformação

Nesta seção, descrevemos as funções de transformação de imagens a serem implementadas. Há funções de edição de imagem, funções de filtro e algumas funções auxiliares.

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 em relação à média dos valores possíveis. Os valores próximos de branco se transformam em próximos de preto e vice-versa. Se a imagem está armazenada na matriz M , o negativo é calculado da seguinte forma: $255 - M[i][j]$

1.1.2 Corte

O corte de uma imagem significa definir uma área retangular dentro da imagem como sendo a própria imagem. Seu programa deve ler as coordenadas do canto superior esquerdo da área de corte (xSup, ySup) e do canto inferior direito (xInf, yInf) e considerar a nova imagem de trabalho como sendo a imagem contida no retângulo definido por estes pontos. As linhas e colunas onde estão os pontos também devem ser incluídas na nova imagem. Caso seja informada alguma coordenada inválida (fora da área definida), avise o usuário e não faça nada com a imagem. Após o corte, a nova matriz de trabalho deve ser a área cortada e as próximas operações devem ser aplicadas sobre esta área.

1.1.3 Rebatimento



Figura 2: Rebatimento horizontal e vertical: pernalonga-h.pgm, pernalonga-v.pgm

Seu programa deve fazer o espelhamento vertical ou horizontal da imagem.

1.1.4 Rotação



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

Seu programa deve fazer a rotação da imagem em 90 graus, sentido horário. Tenha atenção especial no caso das figuras retangulares.

1.2 Funções de filtro

Filtros são transformações com a finalidade de melhorar uma imagem, dar mais nitidez, etc. Todas as funções de filtro irão funcionar baseadas no mesmo princípio. Cada ponto (i, j) da imagem resultante será uma função dos valores de sua vizinhança:

Seja M uma matriz de inteiros positivos com m linhas e n colunas, e seja p um número inteiro ímpar positivo, dada uma coordenada (i, j) em M , a **vizinhança** de uma **janela quadrada** de tamanho p em torno de (i, j) é a submatriz M_{ij} de M com p linhas e p colunas e centro em (i, j) .

Por exemplo, dada a seguinte matriz 5×5

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

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

9	4	5
10	3	2
9	1	6

Nos casos das bordas da imagem, a vizinhança será menor. Por exemplo, a vizinhança para o ponto $(0, 1)$, com uma janela de tamanho 3 é:

9	4	5
10	3	2

Agora vejamos a definição dos filtros.

1.2.1 O filtro da mediana

Filtro da mediana é uma transformação bastante comum para suavizar ruídos do tipo impulso em sinais e imagens digitais. Cada ponto da matriz após a aplicação do filtro é a mediana dos valores da vizinhança do ponto. No caso do exemplo anterior, os números em torno de $(1, 1)$ são 9, 4, 5, 10, 3, 2, 9, 1, 6. Logo, $Mediana(1, 1) = 5$ pois há 4 números menores que 5 e 4 números maiores que 5. Para casos em que existe um número par de vizinhos (alguns pontos nas bordas da imagem) utilizaremos, para simplificar os cálculos, o $(n/2 + 1)$ -ésimo elemento. Assim para o caso do ponto no canto superior esquerdo da matriz, iremos calcular a mediana superior de 3, 4, 9, 10, que é 9. Note que o cálculo de um ponto não deve influenciar o cálculo da mediana de outro ponto.

1.2.2 O filtro da média

No filtro da média, o valor de cada ponto deve ser a média dos valores da vizinhança. Utilize aritmética inteira para o cálculo da média.

1.2.3 Filtro da erosão

Neste filtro o valor de cada ponto em (i, j) será o valor mínimo de seus vizinhos em (k, l) . Ou seja

$$E(i, j) = \min_{(k, l) \in \text{vizinhança de } (i, j)} M(k, l)$$

1.2.4 Filtro de dilatação

Neste filtro o valor de cada ponto (i, j) será o valor máximo de seus vizinhos em (k, l) . Ou seja

$$D(i, j) = \max_{(k, l) \in \text{vizinhança de } (i, j)} M(k, l)$$

1.2.5 Filtros de borda

Filtros de borda procuram zerar posições semelhantes aos seus vizinhos e isolar aquelas diferentes. Vamos implementar três algoritmos diferentes para identificar as bordas.



Figura 4: Imagem original e após filtros de borda 1, 2 e 3: pernalonga-1.pgm, pernalonga-2.pgm, pernalonga-3.pgm

- borda1: utiliza a matriz resultante da subtração da matriz de dilatação pela matriz de erosão.

$$D(i, j) - E(i, j)$$

- borda2: utiliza a matriz resultante da subtração da matriz original pela matriz de erosão.

$$M(i, j) - E(i, j)$$

- borda3: utiliza a matriz resultante da subtração da matriz de dilatação pela matriz original.

$$D(i, j) - M(i, j)$$

Para que os resultados sejam melhor visualizados, torna-se necessário aplicar uma outra transformação, definida a seguir:

- limiarização. A limiarização de uma matriz M na altura $k > 0$ (inteiro) é dada por uma matriz binária L^k , definida como segue: $L^k(i, j) = \begin{cases} 255, & \text{se } M(i, j) \geq k \\ 0, & \text{caso contrário.} \end{cases}$

Você deve aplicar em cada filtro de borda, a limiarização para um k fornecido pelo usuário¹. Desta maneira teremos

- $B1(i, j) = L_k(D(i, j) - E(i, j))$
- $B2(i, j) = L_k(M(i, j) - E(i, j))$
- $B3(i, j) = L_k(D(i, j) - M(i, j))$

2 O que o seu programa deve fazer

Ao executar o programa, ele deve solicitar o prefixo do nome de um arquivo .pgm (sem a extensão .pgm). Caso seja fornecido um nome de arquivo inválido, o programa deve continuar a perguntar o nome do arquivo, até ser fornecido um nome válido e os dados terem sido carregados. Após a carga do arquivo, o programa deve apresentar um menu de opções e deve solicitar um comando:

- c - carga. Seu programa deve ler um novo arquivo. Deve ser solicitado o prefixo do nome de um arquivo .pgm (sem a extensão) e ler os dados nele contido. Caso seja fornecido um nome de arquivo inválido, o programa deve continuar a perguntar o nome do arquivo, até ser fornecido um nome válido e os dados terem sido carregados.
- t - exibição na tela. Seu programa deve mostrar na tela a matriz correspondente à imagem. Utilize o formato "%3d" para impressão dos números.
- n - negativo. Seu programa deve calcular o negativo da imagem.
- r - rotação. Seu programa deve fazer a rotação da imagem.
- v - rebatimento vertical. Seu programa deve fazer o rebatimento vertical da imagem.
- h - rebatimento horizontal. Seu programa deve fazer o rebatimento horizontal da imagem.
- x - corte. Seu programa deve ler os quatro valores das coordenadas e fazer o corte da imagem.

¹O valor das posições da matriz resultante é definido em geral como 0 ou 1. Colocamos os valores 0 ou 255 para melhor visualização dos resultados.

- e - erosão. Seu programa deve aplicar o filtro de mínimo à imagem.
- d - dilatação. Seu programa deve aplicar o filtro de máximo à imagem.
- m - mediana. Seu programa deve aplicar o filtro da mediana.
- M - média. Seu programa deve aplicar o filtro da média.
- 1, 2 ou 3 - filtro de bordas. Seu programa deve aplicar o filtro de borda apropriado (veja a descrição de filtros), com a limiarização para k dado pelo usuário.
- g - gravação. Seu programa deve gravar em um arquivo a imagem que está sendo transformada. O nome do arquivo deve ser o nome do arquivo original, seguido dos comandos de transformação da imagem que foram executados na imagem. Por exemplo, se o arquivo chamado xyz.pgm for transformado pelos comandos r e 2, o arquivo gravado deve ser xyz-r2.pgm
- C - comparação. Seu programa deve ler o prefixo do nome de um arquivo de imagem, ler a imagem e comparar com a imagem atual, informando se são iguais ou diferentes. Esta opção é útil para comparar as imagens que seu programa gera com as imagens geradas pelo programa executável entregue.
- a - ajuda. Seu programa deve exibir novamente o menu de opções.
- s - sair. Seu programa deve ser encerrado.

OBS: Ao carregar um novo arquivo ou reverter para o original, a memória dos comandos de modificação utilizados deve ser zerada, de modo que ao salvar a imagem sejam registrados no nome do arquivo somente os comandos dados daquele momento para frente.

OBS2: Considere o tamanho máximo de janela como sendo 50 e o mínimo como sendo 3. Caso o usuário digite um valor inválido, avise-o e leia novamente o valor até que ele digite um valor válido.

3 Formato dos arquivos

Para este programa vamos utilizar arquivos em formato PGM, que são fáceis de decodificar. Segundo este formato, o arquivo contém um cabeçalho e a matriz correspondente à imagem. Veja exemplo a seguir.

```
P2
5 5
16
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
```

A primeira linha contém uma palavra-chave “P2” que identifica o tipo de arquivo. Será sempre “P2” no nosso caso. A segunda linha contém dois números que correspondem ao número de colunas (primeiro) e linhas (segundo). A terceira linha contém um número que é o maior número da matriz. As demais linhas do arquivo correspondem às linhas da matriz (imagem). Opcionalmente, o arquivo pode conter uma linha com comentários iniciada com o caractere ‘#’. Por questões de simplicidade, considere que as imagens que iremos trabalhar não possuem esta linha (se você pegar um arquivo pgm na internet ou gerado de um aplicativo, você pode editar o arquivo em um editor de texto simples e remover a linha de comentário).

Resumindo, um arquivo PGM possui a seguinte estrutura interna:

```
P2
Largura Altura
Valor maximo de intensidade
Valor pixel (0,0)
Valor pixel (1,0)
Valor pixel (2,0)
...
Valor pixel (x,y)
...
Valor pixel (Largura - 1, Altura - 1)
```

Na Figura 2 temos um exemplo de uma pequena imagem e do seu arquivo texto correspondente no formato PGM do tipo P2.

Gere arquivos pequenos para testar seu programa antes de trabalhar com as imagens reais. Para visualizar as imagens você pode usar algum programa, como o IrfanView. O programa gimp (para Linux) é um transformador de imagens bastante completo, que implementa as transformações pedidas neste EP e inúmeras outras. Com ele você pode,

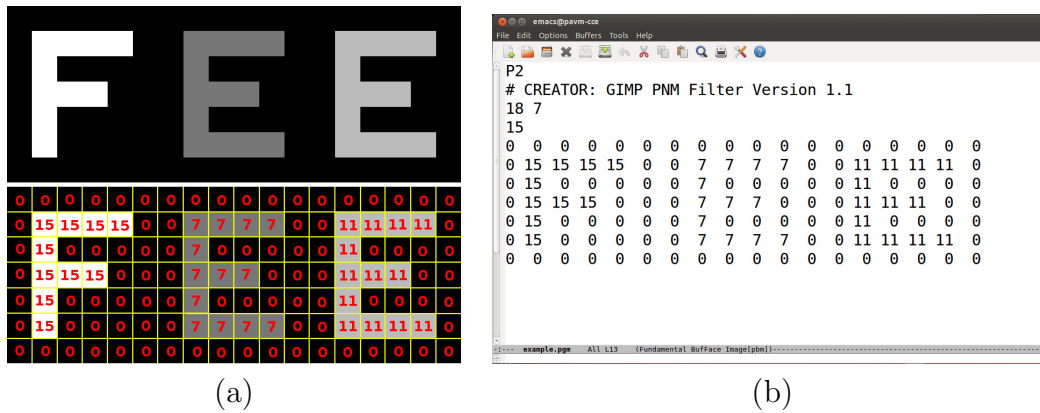


Figura 5: (a) Exemplo de uma imagem 18 x 7, e (b) seu arquivo correspondente no formato PGM sendo visualizado em um editor de textos (*emacs*).

entre muitas outras coisas, converter para o formato PGM imagens que estão em outros formatos. A seguir dois links sobre o formato PGM.

- (1) (PGM faz parte da família PNM)
en.wikipedia.org/wiki/Netpbm_format
- (2) (PGM pode ser ASCII ou raw (binário); nós estamos usando ASCII)
people.sc.fsu.edu/~burkardt/data/pgma/pgma.html
- (3) (Site do IrfanView)
www.irfanview.com

4 Estrutura de seu programa

Seu programa deve **obrigatoriamente** implementar as funções com os protótipos descritos abaixo:

```
int** AlocaMatriz(int m, int n);

void LiberaMatriz(int** M, int m, int n);

int** AbreImagem(char filename[], int* m, int* n);

void GravaImagem(char filename[], int** M, int m, int n);

void negativo(int** M, int m, int n);
```

```

void rebatimentoVertical(int** M, int m, int n);

void rebatimentoHorizontal(int** M, int m, int n);

int** rotacao(int **M, int* m, int* n);

int** corte(int** M, int* m, int* n,
            int xsup, int ysup, int xinf, int yinf);

void filtroMediana(int** M, int m, int n, int larguraJanela);

void filtroMedia(int** M, int m, int n, int larguraJanela);

void filtroErosao(int** M, int m, int n, int larguraJanela);

void filtroDilatacao(int** M, int m, int n, int larguraJanela);

void limiarizacao(int** M, int m, int n, int k);

void filtroBorda1(int** M, int m, int n, int larguraJanela, int k);

void filtroBorda2(int** M, int m, int n, int larguraJanela, int k);

void filtroBorda3(int** M, int m, int n, int larguraJanela, int k);

```

OBSERVAÇÃO 1: Todas as matrizes utilizadas para guardar os dados das imagens devem ser alocadas dinamicamente.

OBSERVAÇÃO 2: Caso seu programa seja finalizado por violações de tamanho de memória, verifique se você não está utilizando matrizes auxiliares desnecessárias que possam estar excedendo o espaço disponível na memória. O programa se comporta bem com até três matrizes de 400x400 na memória ao mesmo tempo.

OBSERVAÇÃO 3: Várias das operações exigem que você faça uma cópia da matriz, pois as novas posições dependem do valor das vizinhas antigas.

Seu programa não deve ser limitado às funções neste enunciado. Divida o seu programa em várias funções razoavelmente curtas, cada uma responsável por uma tarefa não muito grande. Procure sempre criar funções em vez de repetir código. Seu programa ficará menor e mais fácil de desenvolver, de depurar, de corrigir e de manter. A legibilidade do programa também será avaliada.

5 Exemplo

Segue abaixo um exemplo de execução do programa. Você pode simular outros comportamentos a partir do executável de referência entregue junto com o enunciado. Teste também com as figuras fornecidas.

Digite o nome do arquivo de entrada: entrada

Arquivo entrada.pgm carregado com sucesso.

c - carga
t - exibicao na tela
n - negativo
r - rotacao
v - espelhamento vertical
h - espelhamento horizontal
x - corte
e - filtro da erosao
d - filtro da dilatacao
m - filtro da mediana
M - filtro da media
1 - filtro de bordas 1
2 - filtro de bordas 2
3 - filtro de bordas 3
g - gravacao
C - comparacao
a - ajuda
s - sair

Digite um comando: t

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

Digite um comando: r

Operacao realizada com sucesso.

Digite um comando: t

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

Digite um comando: h

Operacao realizada com sucesso.

Digite um comando: t

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

Digite um comando: v
Operacao realizada com sucesso.

Digite um comando: t

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

Digite um comando: e
Digite o tamanho da janela: 3
Operacao realizada com sucesso.

Digite um comando: t

0	0	2	2	1	1	1
0	0	2	2	0	0	0
0	0	3	0	0	0	0
1	1	1	0	0	0	0
1	1	1	0	0	0	0

Digite um comando: c
Digite o nome do arquivo de entrada: entrada
Arquivo entrada.pgm carregado com sucesso.

Digite um comando: d
Digite o tamanho da janela: 3
Operacao realizada com sucesso.

Digite um comando: t

9	10	10	10	8
9	10	10	10	9
7	10	15	15	15
7	7	15	15	15
6	6	15	16	16
12	12	12	16	16
12	12	12	16	16

Digite um comando: c
Digite o nome do arquivo de entrada: entrada
Arquivo entrada.pgm carregado com sucesso.

Digite um comando: m
Digite o tamanho da janela: 3
Operacao realizada com sucesso.
Digite um comando: t

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

Digite um comando: c
 Digite o nome do arquivo de entrada: entrada
 Arquivo entrada.pgm carregado com sucesso.

Digite um comando: M
 Digite o tamanho da janela: 3
 Operacao realizada com sucesso.
 Digite um comando: t

4	5	4	4	3
4	5	4	5	4
3	4	6	6	6
2	3	6	6	8
1	1	4	6	8
4	3	3	5	6
5	4	3	4	5

Digite um comando: x
 Informe x superior: 1
 Informe y superior: 3
 Informe x inferior: 5
 Informe y inferior: 5
 Limites invalidos

Digite um comando: x
 Informe x superior: 1
 Informe y superior: 3
 Informe x inferior: 4
 Informe y inferior: 4
 Operacao realizada com sucesso.
 Digite um comando: t

5	4
6	6
6	8
6	8

Digite um comando: n
 Operacao realizada com sucesso.
 Digite um comando: t

250	251
249	249
249	247

249 247

Digite um comando: c
Digite o nome do arquivo de entrada: entrada
Arquivo entrada.pgm carregado com sucesso.

Digite um comando: 1
Digite o tamanho da janela: 3
Informe o valor de k: 1
Operacao realizada com sucesso.
Digite um comando: t
255 255 255 255 255
255 255 255 255 255
255 255 255 255 255
255 255 255 255 255
255 255 255 255 255
255 255 255 255 255
255 255 255 255 255

Digite um comando: c
Digite o nome do arquivo de entrada: entrada
Arquivo entrada.pgm carregado com sucesso.

Digite um comando: 2
Digite o tamanho da janela: 3
Informe o valor de k: 1
Operacao realizada com sucesso.
Digite um comando: t
255 255 255 0 255
0 255 255 255 255
0 255 255 255 255
255 255 255 255 0
255 0 255 255 255
255 0 0 255 255
255 255 255 255 255

Digite um comando: c
Digite o nome do arquivo de entrada: entrada
Arquivo entrada.pgm carregado com sucesso.

Digite um comando: 3
Digite o tamanho da janela: 3
Informe o valor de k: 1
Operacao realizada com sucesso.
Digite um comando: t
0 255 255 255 0
255 255 0 255 255
255 255 255 255 255
255 255 255 0 255

255	255	255	255	255
255	255	255	255	0
255	0	255	255	255