

Trabalho 4

Prof. Rodrigo Fernandes de Mello – mello@icmc.usp.br
Monitores: Igor Martinelli – igor.martinelli@usp.br
Victor Forbes – victor.forbes@usp.br,
Yule Vaz – yule.vaz@usp.br

1 Processamento de Imagens

Uma imagem preta e branca é, tipicamente, representada por uma matriz bidimensional cujos elementos apresentam, tipicamente, valores em um intervalo $[0, v_{max}]$ que representam intensidades de luz cuja unidade de localização é chamada pixel. Os valores associados à cada pixel podem ser definidos como uma função $f(x, y)$ em que x e y são, respectivamente, suas posições horizontal e vertical. Neste contexto, é possível manipular as informações de intensidades de luz e posição de uma imagem, de forma a realçar características de interesse, o que é denominado de processamento de imagens.

1.1 Imagem PGM

A designação de formato de imagem PBM (Portable Bitmap) engloba três formatos de imagem para imagens a preto e branco, em escala de tons cinzentos e a cores, todos eles sem compressão e que apresentam uma estrutura comum. Estes três tipos de formato de imagens são:

- PBM (Portable BitMap) - imagens a preto e branco (sem tons de cinzento)
- PGM (Portable GrayMap) - imagens em tons de cinzento
- PPM (Portable PixMap) - imagens a cores

A definição original destes formatos teve em vista permitir a transmissão de imagens por meio de correio eletrônico que até então não permitia a transmissão de ficheiros anexados, binários ou não. A definição do formato foi mais tarde modificada para permitir a representação binária dos conteúdos das imagens.

Os formatos de imagem PGM são constituídos pelos seguintes campos:

```
<magic number>
<Altura da imagem 'm'> <Largura da imagem 'n'>
<Valor máximo dos tons de cinza>
<valor do pixel (0,0)> ... <valor do pixel (0,n)>
.
.
.
<valor do pixel (m,0)> ... <valor do pixel (m,n)>
```

O Identificador do tipo de formato (designado por “magic number”), é determinado de acordo com Tipo, ASCII ou Binário

Tipo	ASCII	Binário
PBM	P1	P4
PGM	P2	P5
PPM	P3	P6

Em particular, no nosso caso, iremos trabalhar com imagens PGM do tipo ASCII, ou seja, “P2”. Assim, um exemplo de imagem PGM que o programa deve ser capaz de processar é:

```
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 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 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
```

Note que a segunda linha do arquivo apresenta as dimensões da imagem, e a terceira linha o valor máximo v_{max} de intensidade de luz que pode ser atribuída a cada pixel.

2 Descrição 1: Realce de Imagens

Uma das técnicas comumente utilizadas na área de processamento é o realce de imagens que pode melhorar tanto a qualidade de algum aspecto fotográfico, tal como iluminação, como também de algum aspecto que facilite futuras análises científicas. O realce é definido como a transformação ponto-a-ponto de uma imagem tal que $r(x, y) = g(f(x, y))$. Por exemplo, a identidade de uma imagem é uma operação trivial de realce tal que $r(x, y) = 1 * f(x, y)$. Já a transformação de negativo de uma imagem é definido como $r(x, y) = v_{max} - f(x, y)$.

Implemente um algoritmo que produza o negativo de uma imagem PGM. Ele deverá imprimir na tela os valores de intensidade de luz também no formato PGM.

3 Descrição 2: Filtragem Espacial de Imagens

Outra técnica bastante utilizada para a melhoria na qualidade de visualização de uma imagem é a filtragem espacial. Essa técnica é comumente aplicada a partir da convolução entre a imagem e uma máscara, ou filtro espacial, w , definida por uma matriz $\mathbb{R}^{N \times N}$. O operador de convolução bidimensional é definido por:

$$g(x, y) = w(x, y) * f(x, y) = \sum_{u=-N/2}^{N/2} \sum_{v=-N/2}^{N/2} w(u, v) f(x - u, y - v) \quad (1)$$

sendo a origem – ponto $(0, 0)$ – de w o seu elemento central. A aplicação de tal operador sobre uma determinada imagem é ilustrada na Figura 1, em que o filtro é empregado em todo x e y da imagem. Note que a sua imagem deverá ser estendida quando a máscara for aplicada sobre suas laterais e seus cantos. Tal como ilustrado na Figura 2, atribua o valor zero a essa extensão, que não deverá ser impressa na saída do seu programa. A imagem da saída deverá ter as mesmas dimensões da de entrada, ou seja, a extensão é apenas utilizada para aplicar a convolução.

Cabe ainda ressaltar que os pixels podem assumir valores maiores que a intensidade de luz máxima permitida. Neste caso, o valor do pixel deverá ser normalizado para:

$$f_{new}(x, y) = v_{max} \frac{g(x, y) - g_{min}}{g_{max} - g_{min}} \quad (2)$$

tal que v_{max} é o maior valor, já pré-definido no arquivo `.pgm`, que as intensidades de luz podem assumir, $g(x, y)$ a aplicação da convolução entre a máscara w e a imagem f , e g_{max} e g_{min} são o máximo e mínimo valores de $g(x, y)$, respectivamente.

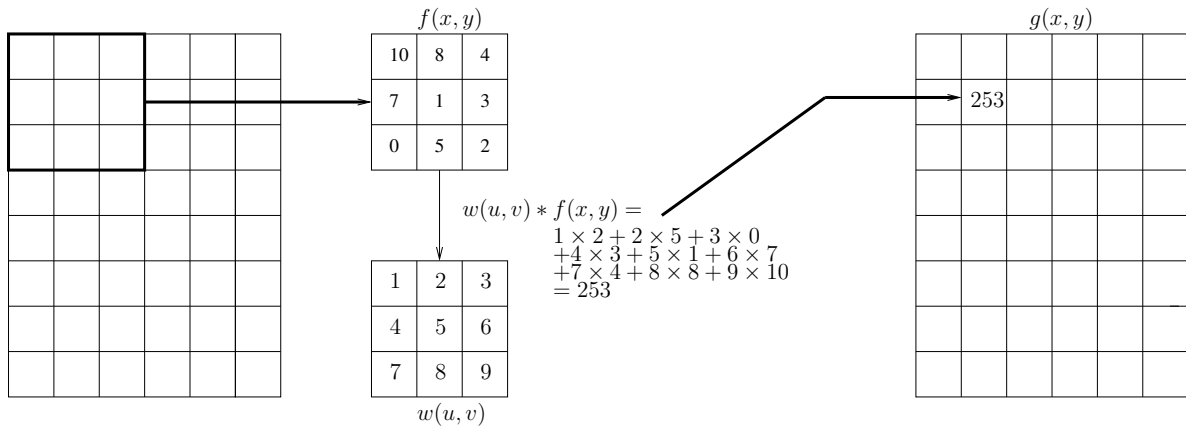


Figura 1: Aplicação de um filtro espacial sobre uma imagem por convolução.

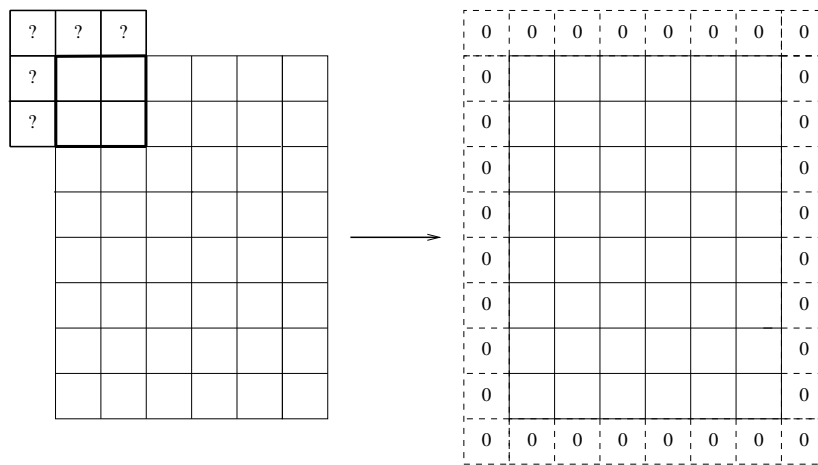


Figura 2: Extensão da imagem para aplicação da convolução.

Implemente um algoritmo que leia uma máscara a partir de um arquivo e aplique a convolução entre essa máscara e uma imagem PGM. Ele deverá imprimir na tela os valores de intensidade de luz também no formato PGM.

3.1 Entradas e saídas

Serão entregues para o programa: 1) a operação a ser efetuada (negativo da imagem: 1, ou filtragem espacial: 2); 2) o nome da imagem; e, caso a operação seja a de filtragem espacial, 3) o nome do arquivo que contém a máscara. São apresentados nas Tabelas 1 e 2 as entradas e suas respectivas saídas. Além disso, para cada uma dessas entradas são ilustrados, nas Figuras 3 e 4, suas respectivas imagens. O arquivo que contém os filtros apresentam, na primeira linha, suas dimensões. Como as máscaras são quadradas, há apenas um valor. As linhas restantes definem os valores da matriz w , tal como na Tabela 3.



Figura 3: Imagem associada à entrada 1.in.

Entrada

1
img1.pgm

Saída

P2
24 7
15
15 15 15 15 15 15 15 15 15 15
15 15 15 15 15 15 15 15 15
15 15 15 15 15
15 12 12 12 12 15 15 8 8 8 8 15
15 4 4 4 4 15 15 0 0 0 0 15
15 12 15 15 15 15 15 8 15 15 15
15 15 4 15 15 15 15 15 0 15
15 0 15
15 12 12 12 15 15 15 8 8 8 15
15 15 4 4 4 15 15 15 0 0 0 0
15
15 12 15 15 15 15 15 8 15 15 15
15 15 4 15 15 15 15 15 0 15
15 15 15
15 12 15 15 15 15 15 8 8 8 8 15
15 4 4 4 4 15 15 0 15 15 15
15
15 15 15 15 15 15 15 15 15
15 15 15 15 15 15 15 15
15 15 15 15 15

Tabela 1: Entradas 1.in e sua respectiva saída 1.out

Entrada

2
img1.pgm
conv1.mat

Saída

P2
24 7
15
7 7 7 7 7 7 6 6 6 6 7 7 5 5 5
5 7 7 5 5 5 5 7
7 8 8 8 9 7 6 9 9 9 11 6 5 11
11 11 13 5 5 12 12 12 12 5
7 8 6 6 7 7 6 9 4 5 6 7 5 11 2
3 5 7 5 12 0 0 12 5
7 8 8 9 7 7 6 8 9 11 6 7 5 9 11
13 5 7 5 10 12 12 12 5
7 8 6 7 7 7 6 9 4 5 6 7 5 11 2
3 5 7 5 12 2 5 5 7
7 9 7 7 7 7 6 9 9 9 11 6 5 11
11 11 13 5 5 15 5 7 7 7
7 7 7 7 7 7 6 6 6 6 7 7 5 5 5
5 7 7 5 7 7 7 7

Tabela 2: Entradas 2.in e sua respectiva saída 2.out



Figura 4: Imagem associada à entrada 2.in.

```
3
0 -1 0
-1 4 -1
0 -1 0
```

Tabela 3: Exemplo do arquivo que contém o filtro para a operação de convolução.

3.2 Informações Adicionais

Serão levados em consideração na correção do trabalho:

- Modularização do código;
- Comentários;
- Alocação dinâmica de memória;
- Desalocação;
- Acesso indevido de memória;
- Inicialização de variáveis;
- Legibilidade do código (identação, códigos desnecessários, nomes de variáveis, etc.).