Universidade Federal de Minas Gerais | **Algoritmos e Estruturas de Dados II**
Instituto de Ciências Exatas | **1º Semestre de 2014**
Departamento de Ciência da Computação | **Prof. Luiz Chaimowicz e Raquel Prates**

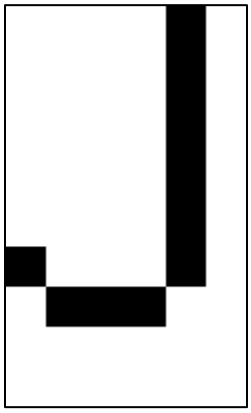
Exercício de Programação 2 – Detecção de Bordas

Valor: 5 pontos
Data de devolução: 01/04/2014

Um dos avanços da computação gráfica foi a possibilidade de pegar imagens e fotografias e criar versões digitalizadas destas. Existem diferentes formatos de armazenamento de imagens, e um desses é o *pbm*. O formato de imagem *pbm* é usado para imagens em preto e branco possui uma estrutura bem simples, representando pixels brancos com 0 e pixels pretos com 1. Os arquivos *pbm* sempre têm:

- Um identificador do formato de imagem;
- As dimensões de largura e altura da imagem;
- Os valores de cor para todos os pixels da imagem.

Abaixo está a imagem (ampliada) e sua descrição em *pbm*:



(a)

```
P1
6 10
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
1 0 0 0 1 0
0 1 1 1 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

(b)

Figura 1. Imagem da letra "J" ampliada vinte vezes (a) e sua representação no formato *pbm* (b)

O seu trabalho será aplicar um filtro de detecção de bordas em uma imagem de extensão *pbm*.

A detecção de bordas é uma técnica de processamento de imagens para encontrar áreas em uma imagem onde há mudanças abruptas no brilho. Um dos meios de fazer isso é aplicando um filtro de detecção de bordas.

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Figura 2. Filtro de detecção de bordas

Esse filtro é uma matriz de tamanho 3×3 e sua aplicação funciona da seguinte forma: considere uma imagem como uma matriz *M* de tamanho I×J. A aplicação do filtro na imagem original irá gerar uma nova matriz, *B*. Para cada posição da matriz (junto com seus 8 vizinhos), você irá multiplicar seus valores pelo filtro de detecção. Veja o exemplo:

1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0

×

-1	-1	-1
-1	8	-1
-1	-1	-1

=

1	1	0	0	0
1	3	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0

Figura 3.a. Aplicando o filtro de detecção de borda na posição (1,1) da matriz

1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0

×

-1	-1	-1
-1	8	-1
-1	-1	-1

=

1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	0	0	0

Figura 3.b. Aplicando o filtro de detecção de borda na posição (4,3) da matriz

De forma genérica, podemos dizer que:

$$B_{i,j} = 8 * M_{i,j} - (M_{i-1,j-1} + M_{i-1,j} + M_{i-1,j+1} + M_{i,j-1} + M_{i,j+1} + M_{i+1,j-1} + M_{i+1,j} + M_{i+1,j+1})$$

É estritamente necessário que o pixel $M_{i,j}$ possua 8 vizinhos. Portanto, nos casos onde $i = 0$, $j = 0$, $i = I-1$ e $j = J-1$ não é possível calcular o valor de $B_{i,j}$. Para esse trabalho você poderá considerar que $B_{i,j} = 0$ para esses casos especiais.

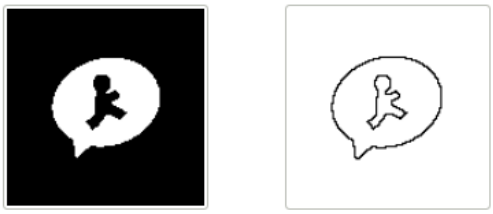


Figura 4. Exemplo de aplicação do filtro

Um exemplo de aplicação do filtro é visto acima. A imagem da esquerda é um ícone do serviço de chat AIM, a imagem da direita é o resultado da aplicação do filtro. Observe que somente as bordas da imagem original aparecem na imagem.

No seu trabalho você deverá ler um arquivo de imagem *pbm*, aplicar o filtro de detecção de bordas e ao fim imprimir a nova imagem em um outro arquivo *pbm*. Você deverá criar um tipo abstrato de dados *tipoImagem*, que armazenará a altura, a largura e a matriz de pixels da imagem e possuirá pelo menos 3 funções: *LeImagem*, *DetectaBorda* e *ImprimeImagem*, todas do tipo *void*. O seu main, será basicamente

a chamda dessas três funções. Fica a seu cargo quais são os parâmetros de entrada dessas funções. O formato de entrada e saída é o mesmo do Exercício 1:

./programa < entrada.pbm > saída.pbm

Lembre-se de levar em conta as seguintes considerações:

- Existem casos onde o resultado do filtro pode ser menor que 0 ou maior que 1. Considere que nesses casos o resultado será 0 ou 1.
- Lembre-se que você terá problemas para multiplicar a matriz de detecção de borda nas extremidades da imagem. Nesses casos, pode considerar que o cálculo nas bordas é igual a 0.
- A string P1 do início do arquivo é necessária somente para identificar a imagem como um arquivo do tipo *pbm*. Você não a usará para detectar as bordas, mas é necessário imprimi-la na sua função *ImprimeImagem*.
- Caso seu sistema operacional não suporte o formato *pbm*, você irá precisar baixar um aplicativo para abrir as imagens. Para o Windows, uma opção é o IrfanView.

O que deve ser entregue:

• **Onde:** o trabalho deverá ser entregue no PRATICO. Para isso, você deverá ter se cadastrado no Prático (http://aeds.dcc.ufmg.br/) e na turma (Turma N - Raquel Prates e Chaimowicz - AEDS2)

• **O que deve ser entregue:** Código fonte do programa em C (bem identado e comentado). Lembre-se que você não pode usar funções específicas de C++.

Comentários Gerais:

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 e identação serão observados.

3 O trabalho é individual.

4 Trabalhos copiados serão penalizados conforme anunciado.

5 Penalização por atraso: $(2^d - 1)$ pontos, onde d é o número de dias (úteis) de atraso. Note que após dois dias úteis, o trabalho não pode mais ser entregue.