

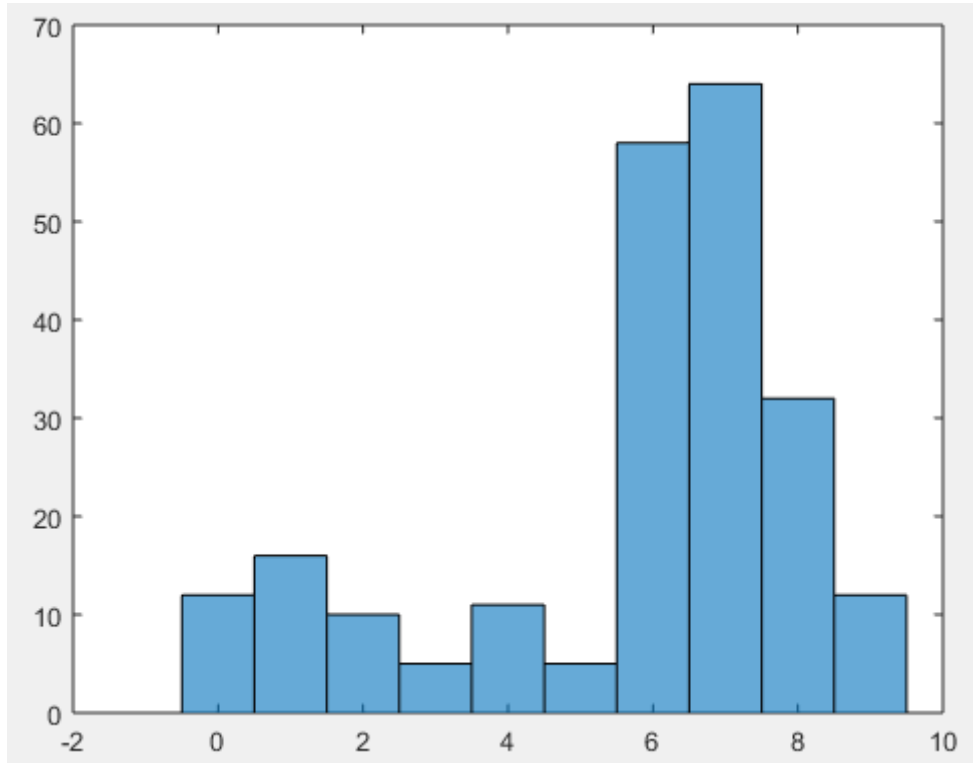


UNIVERSIDADE FEDERAL DE ITAJUBÁ – UNIFEI – CAMPUS ITABIRA  
CURSO DE ENGENHARIA DA COMPUTAÇÃO - ECO  
COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DIGITAL DE IMAGENS  
PROF. GIOVANI B. VITOR  
3ª LISTA DE EXERCÍCIOS

DAN QUENAZ MAGDYAN SILVA PIMENTEL - 26728

1. **Imagem monocromática:** sensores medem a reflexão de luz visível das superfícies dos objetos (amostragem), transformando essas medidas em números inteiros (quantização). **Imagem colorida:** cada pixel  $p$  terá associado um vetor  $\mathbf{I}(p) = (\mathbf{I1}(p), \mathbf{I2}(p), \mathbf{I3}(p))$  com medidas de reflexão de luz nos comprimentos de onda do RGB respectivamente. **Imagem de Satélite:** pode conter para cada pixel amostras da luz refletida na Terra em vários comprimentos de onda. **Ressonância magnética:** mede o tempo de relaxação dos *spins* dos átomos de hidrogênio em  $(x, y, z)$  após desmagnetização de um campo magnético forte. A quantização mapeia os valores reais dos pixels em números inteiros  $I(p)$  de 0 a  $2^b - 1$ , onde  $b$  é o número de bits necessários para representar cada pixel. Se  $f(x, y)$  representa uma função contínua da reflexão de luz no espaço  $R^2$ , a amostragem gera uma função discreta  $I(x, y)$ .
2. As imagens do mundo real são capturadas através de dispositivos como uma câmera. Utilizando técnicas de CG é possível extrair informações como tamanho real.
3. a)  $p = \frac{400^2}{4000^2} \Rightarrow p = 0.04 \Rightarrow p = 0,2 \times 0,2$   
b)  $2^b - 1 = 4095 \Rightarrow b = 12$   
c) Maior resolução, possui o dobro de pixels por metro. 8000x8000.

4. a)



b) Pode-se observar uma imagem com um contraste alto, ou seja com maior números de tons claros.

c)C1:

[illegible]



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

[illegible]



5. a) Erosão B4

[illegible]

## b) Erosão B8

[illegible]

c)Dilatacao B4

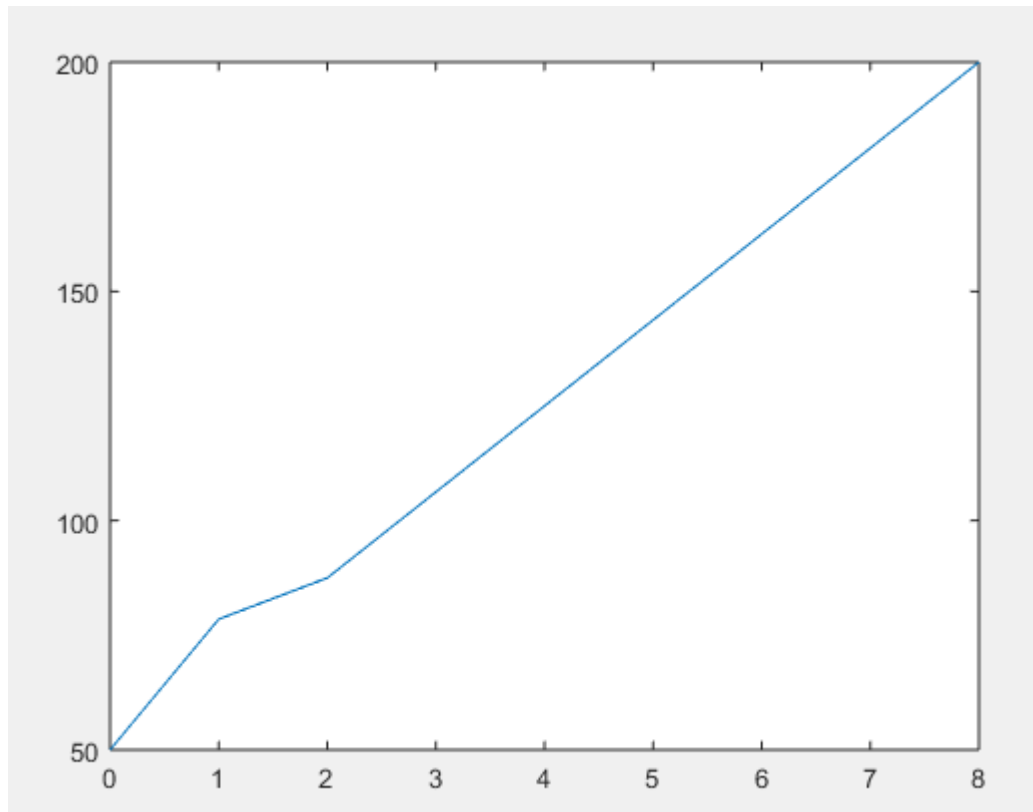
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1
1	1	1	2	2	2	2	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	2	2	2	2	2	1	1
1	2	2	2	2	2	2	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	2	2	2	2	1	1	1
1	2	1	1	1	2	2	2	2	2	1	1	1	1	1
2	2	2	1	1	2	2	2	2	2	1	1	1	1	1
2	2	2	1	1	2	2	2	2	2	1	1	1	1	1
1	2	2	1	1	2	2	2	2	2	1	1	1	1	1
1	2	2	2	1	2	2	2	2	2	1	1	1	1	1
2	2	2	2	2	1	2	2	2	1	1	1	1	2	1
2	2	2	2	2	2	1	1	1	1	1	1	2	2	2
2	2	2	2	2	2	2	1	1	1	1	1	1	2	2

d)Dilatacao B8

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	2	2	2	2	2	2	2	2	2	2	1	1
1	1	2	2	2	2	2	2	2	2	2	2	2	1	1
1	2	2	2	2	2	2	2	2	2	2	2	2	1	1
1	2	2	2	2	2	2	2	2	2	2	2	2	1	1
1	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2	2	2	1	1	2	2	2	2	2	1	1	1	1	1
2	2	2	1	1	2	2	2	2	2	1	1	1	1	1
2	2	2	1	1	2	2	2	2	2	1	1	1	1	1
2	2	2	2	1	2	2	2	2	2	1	1	1	1	1
1	2	2	2	2	2	2	2	2	2	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	1	1	2	2	2
2	2	2	2	2	2	2	1	1	1	1	1	2	2	2
2	2	2	2	2	2	2	1	1	1	1	1	2	2	2

e)

6.



Por interpolação:

0 – 50; 1 – 78.5; 2 – 87.5; 8 – 200.

7. a)

8.0000	9.0000	8.5000	10.0000	8.5000
9.0000	7.2500	5.5000	9.2500	10.0000
8.0000	4.5000	2.0000	6.5000	9.0000
9.0000	7.2500	5.5000	9.2500	10.0000
8.0000	9.0000	8.5000	10.0000	8.5000

b)

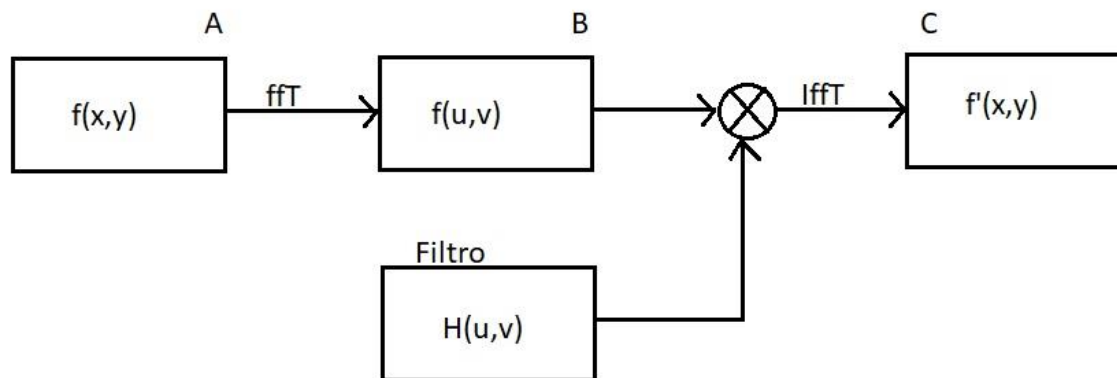
5.3333	7.1111	6.2222	8.0000	6.2222
7.1111	9.3333	6.6667	10.2222	8.0000
5.3333	5.7778	2.2222	7.5556	7.1111
7.1111	9.3333	6.6667	10.2222	8.0000
5.3333	7.1111	6.2222	8.0000	6.2222

c)

-32	-16	-8	-32	-40
16	44	56	44	16
0	0	0	0	0
-16	-44	-56	-44	-16
32	16	8	32	40



8.



a) Não haverá alterações na imagem resultante, devido ao fato de se realizar a normalização. Mesmo sendo uma fração ainda se mantém o mesmo.

b) Devido o fato de o filtro já ter sido aplicado, não se observa mudança alguma após a multiplicação. Entretanto se for aplicado o filtro diversas vezes no resultado da transformada, vai se observar então uma imagem com uma única cor.

- Código utilizado para resolução das questões 4 e 5.

```
matriz = [9,9,9,8,9,8,9,8,8,8,8,9,9,9,9;
          8,8,9,8,8,8,8,8,7,8,8,8,9,8,9;
          7,7,7,6,6,7,7,7,7,6,6,6,7,8,8;
          6,7,6,6,2,1,1,1,1,1,1,1,7,8,8;
          7,7,6,4,3,1,0,0,0,1,2,2,6,7,8;
          7,6,5,4,2,0,0,0,0,1,2,5,6,7,8;
          7,7,6,6,7,6,0,0,1,6,7,7,7,8,7;
          7,7,7,7,6,6,0,0,1,6,7,7,8,7,8;
          7,5,8,7,6,6,0,1,1,6,7,7,7,7,8;
          7,4,7,8,7,6,1,1,2,6,7,7,7,8,7;
          6,6,6,7,8,7,2,2,3,6,7,6,7,7,7;
          6,6,4,6,7,6,2,3,5,7,6,6,6,7,7;
          6,6,3,4,6,6,6,7,7,6,6,6,6,7,7;
          4,4,6,4,4,6,6,6,7,7,6,7,7,5,8;
          2,3,6,6,6,4,6,7,6,7,6,6,6,7,4];
```

```
matrizC1 = zeros(15);
matrizC2 = zeros(15);
matrizC3 = zeros(15);
matrizC4 = zeros(15);
```

```
matrizI1 = zeros(15);
matrizTretada = zeros(17);
```

```
kernelB4 = [0 1 0; 1 1 1; 0 1 0];
kernelB8 = [1 1 1; 1 1 1; 1 1 1];
```

```
zero=0;
um =0;
dois=0;
tres=0;
quatro=0;
cinco=0;
seis=0;
sete=0;
oito=0;
nove=0;
```

```
for i=1:size(matriz,1)
    for j=1:size(matriz,2)
        if(matriz(i,j)<=5)
            matrizI1(i,j)=1;
        end

        if(matriz(i,j)==0)
            zero=zero+1;
        elseif(matriz(i,j)==1)
```

```

        um=um+1;
        matrizC4(i,j)=1;
    elseif(matriz(i,j)==2)
        dois = dois+1;
        matrizC3(i,j) = 2;
    elseif(matriz(i,j)==3)
        tres = tres+1;
        matrizC3(i,j) = 2;
        matrizC4(i,j)=1;
    elseif(matriz(i,j)==4)
        quatro=quatro+1;
        matrizC2(i,j) = 4;
    elseif(matriz(i,j)==5)
        cinco=cinco+1;
        matrizC2(i,j) = 4;
        matrizC4(i,j)=1;
    elseif(matriz(i,j)==6)
        seis=seis+1;
        matrizC2(i,j) = 4;
        matrizC3(i,j) = 2;
    elseif(matriz(i,j)==7)
        sete=sete+1;
        matrizC2(i,j) = 4;
        matrizC3(i,j) = 2;
        matrizC4(i,j)=1;
    elseif(matriz(i,j)==8)
        oito=oito+1;
        matrizC1(i,j)=8;
    else
        nove=nove+1;
        matrizC1(i,j)=8;
        matrizC4(i,j)=1;
    end
end
end

matrizTretada(2:16, 2:16) = matrizI1;
disp(matrizTretada);
matrizAUX = zeros(3);
erosaoB4 = zeros(15);
erosaoB8 = zeros(15);
dilatacaoB4 = zeros(15);
dilatacaoB4 = zeros(15);

%Erosao B4
for i=2:16
    for j=2:16
        for x=1:size(kernelB4, 1)
            for y=1:size(kernelB4, 2)
                matrizAUX(x, y) = matrizTretada((i-x)+2, (j-
y)+2) - kernelB4(x, y);
            end
        end
    end
end

```

```

        end
    end
    erosaoB4(i-1, j-1)= min(min(matrizAUX));
    matrizAUX = zeros(3);
end
end

%Erosao B8
for i=2:16
    for j=2:16
        for x=1:size(kernelB4, 1)
            for y=1:size(kernelB4, 2)
                matrizAUX(x, y) = matrizTretada((i-x)+2, (j-
y)+2) - kernelB8(x, y);
            end
        end
        erosaoB8(i-1, j-1)= min(min(matrizAUX));
        matrizAUX = zeros(3);
    end
end

%Dilatacao B4
for i=2:16
    for j=2:16
        for x=1:size(kernelB4, 1)
            for y=1:size(kernelB4, 2)
                matrizAUX(x, y) = matrizTretada((i-x)+2, (j-
y)+2) + kernelB4(x, y);
            end
        end
        dilatacaoB4(i-1, j-1)= max(max(matrizAUX));
        matrizAUX = zeros(3);
    end
end

%Dilatacao B8
for i=2:16
    for j=2:16
        for x=1:size(kernelB4, 1)
            for y=1:size(kernelB4, 2)
                matrizAUX(x, y) = matrizTretada((i-x)+2, (j-
y)+2) + kernelB8(x, y);
            end
        end
        dilatacaoB8(i-1, j-1)= max(max(matrizAUX));
        matrizAUX = zeros(3);
    end
end

matrizRes = matrizC1 + matrizC2;

```

```

%figure;
%histogram(matrizRes);
%figure;
%histogram(matriz);

```

- Código utilizado para resolução da questão 6.

```

%%%%Questão_6%%%%

clear all;

vet1 = [0 1 2 8];

% Por interpolação
% 0 - 50
% 1 - 78.5
% 2 - 87.5
% 8 - 200

vet2 = [50 78.5 87.5 200];

plot(vet1, vet2);

```

- Código utilizado para resolução da questão 7.

```

%%%QUESTÃO 07%%%
clear all;

Matriz = [16 16 16 16 16;
          16 0 0 8 16;
          16 0 4 0 16;
          16 0 0 8 16;
          16 16 16 16 16];

MatrizExpan = zeros(7);
MatrizExpan(2:6, 2:6) = Matriz;

kernel1 = [1/16 2/16 1/16;
            2/16 4/16 2/16;
            1/16 2/16 1/16];

kernel2 = [1/9 1/9 1/9;
            1/9 1/9 1/9;
            1/9 1/9 1/9];

```

```
kernel3 = [-1 -2 -1;  
           0 0 0;  
           1 2 1];
```

```
matrizA = zeros(5);  
matrizB = zeros(5);  
matrizC = zeros(5);
```

```
%Letra_a  
for i=2:6  
    for j=2:6  
        for x=1:3  
            for y=1:3  
                matrizA(i-1, j-1) = matrizA(i-1, j-1) +  
kernel1(x, y)*MatrizExpan(i-x+2, j-y+2);  
            end  
        end  
    end  
end
```

```
%Letra_b  
for i=2:6  
    for j=2:6  
        for x=1:3  
            for y=1:3  
                matrizB(i-1, j-1) = matrizB(i-1, j-1) +  
kernel2(x, y)*MatrizExpan(i-x+2, j-y+2);  
            end  
        end  
    end  
end
```

```
%Letra_c  
for i=2:6  
    for j=2:6  
        for x=1:3  
            for y=1:3  
                matrizC(i-1, j-1) = matrizC(i-1, j-1) +  
kernel3(x, y)*MatrizExpan(i-x+2, j-y+2);  
            end  
        end  
    end  
end
```