

PROJECT #13

Prof^a: Marly G. F. Costa*Autor*: Pedro Victor Dos Santos Matias, 21601225

Introdução

Este trabalho apresenta uma aplicação de processamento digital de imagens em uma imagem de raio-x. Nesse projeto é feito uma filtragem no domínio da frequência para realçar os detalhes finos, como as bordas dos ossos. Dois processos são implementados em MATLAB, o primeiro com um filtro gaussiano e o segundo com um filtro passa-alta de ênfase.

A próximas seções apresentaram os procedimentos executados; os resultados obtidos com as imagens resultantes da filtragem; conclusão com as considerações sobre os resultados e uma apêndice com os codigos implementados.

Procedimentos Metodológicos

Esse projeto possui o seguintes procedimentos:

1. Aplique um filtro gaussiano passa-alta com $D_0 = 40$ na imagem abaixo e mostre o resultado.

Imagem original

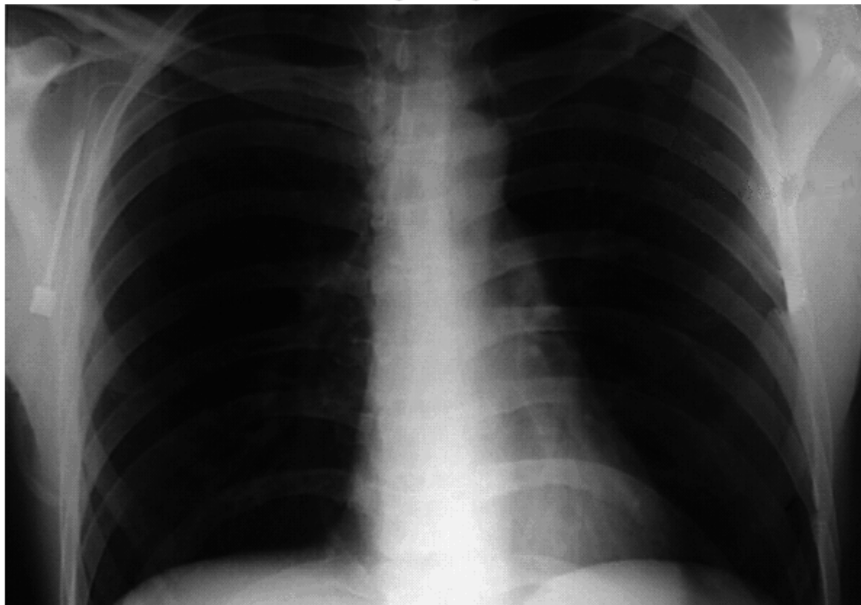


Figura 1: Imagem original

2. Aplique um filtro de ênfase, dado pela equação abaixo e varie os valores de k_1 e k_2 pede-se que:
 - (a) Mostre a imagem resultante do filtro de ênfase
 - (b) equalize o histograma da imagem e mostre a imagem resultante

$$g(x, y) = \mathfrak{F}^{-1}([k_1 + K_2 * H_{HP}(u, v)] F(u, v))$$

Em que:

k_1 - Controla o offset da origem;

k_2 - Controla a contribuição da alta frequência; $H_{HP}(u, v)$ - Filtro Gaussiano passa alta com $D_0 = 40$

k_1	k_2
0	1
0.2	0.8
0.5	0.5
0.8	0.2
1	0

Tabela 1: Valores de combinação para k_1 e k_2

O processamento da filtragem da imagem foi realizado a ferramenta MATLAB seguindo as etapas abaixo:

1. Realizar o padding e multiplicar a imagem por $(-1)^{x+y}$ para centralizar o espectro de frequência
2. Geração do filtro no domínio da frequência, $H(u, v)$
3. Calcular a transformada de fourier da imagem, $f(x, y) \rightarrow F(u, v)$
4. Produto de Hadamard (multiplicação de arrays por elemento a elemento) entre a imagem transformada $F(u, v)$ com a máscara $H(u, v)$.

$$G(u, v) = H(u, v) \circ F(u, v)$$

5. Calculo do da transformada inversa de $G^{-1}(u, v)$
6. Realizar um *cropping* para remover o padding e multiplicar o resultado $(-1)^{x+y}$ para recentralizar.

Resultados

O filtro passa alta permite atenuar frequências baixas, realçando detalhes finos da imagem (componentes de alta frequência). Para ambas as questões há a necessidade de aplicar um filtro gaussiano passa-alta. O filtro pode ser escrito como:

$$H(u, v) = e^{\frac{-D^2(u, v)}{2\sigma^2}}.$$

Onde σ é uma medida de dispersão, em sistemas de comunicação denominada ruído Gaussiano. Substituindo essa medida pelo valor equivalente a frequência de corte D_o . obtemos:

$$H(u, v) = e^{\frac{-D^2(u, v)}{2 \cdot D_o^2}}.$$

O filtro é naturalmente um suavizador retirando qualquer frequência acima da frequência de corte, para realizar o oposto subtraímos o filtro de 1, logo nosso filtro gaussiano passa-alta fica:

$$H_{HP}(u, v) = 1 - e^{\frac{-D^2(u, v)}{2 \cdot D_o^2}}.$$

0.1 Aplicação do filtro gaussiano passa-alta com $D_o = 40$

Os resultados obtidos com aplicação do filtro na imagem original foi a imagem seguinte:

Nesse resultado ocorreu o realce das bordas do osso com o fundo, que são os detalhes de alta frequência. A imagem fica escura devido não ter o termo DC para clarear e é feito um redimensionamento dos níveis de cinza com o `imshow(g, [])`.

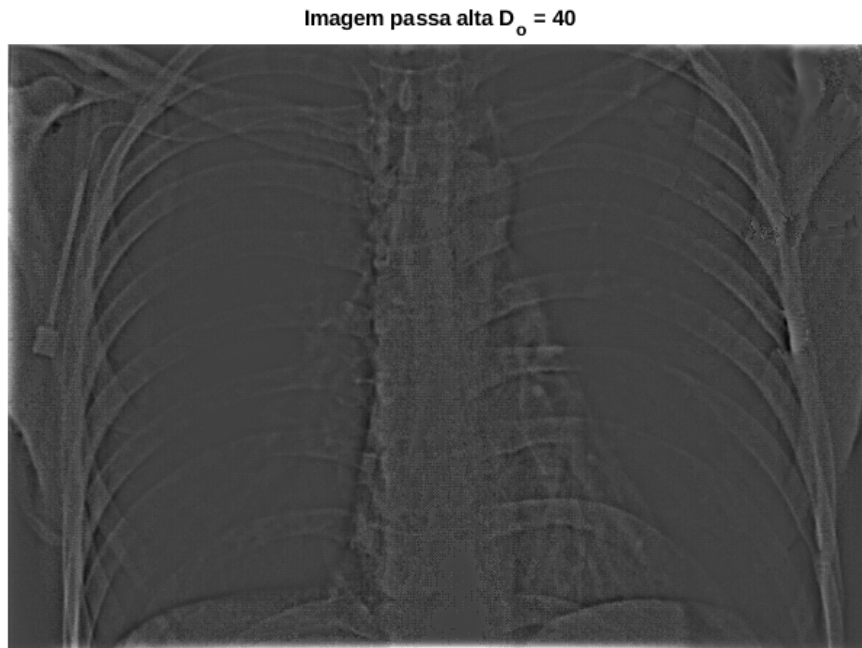
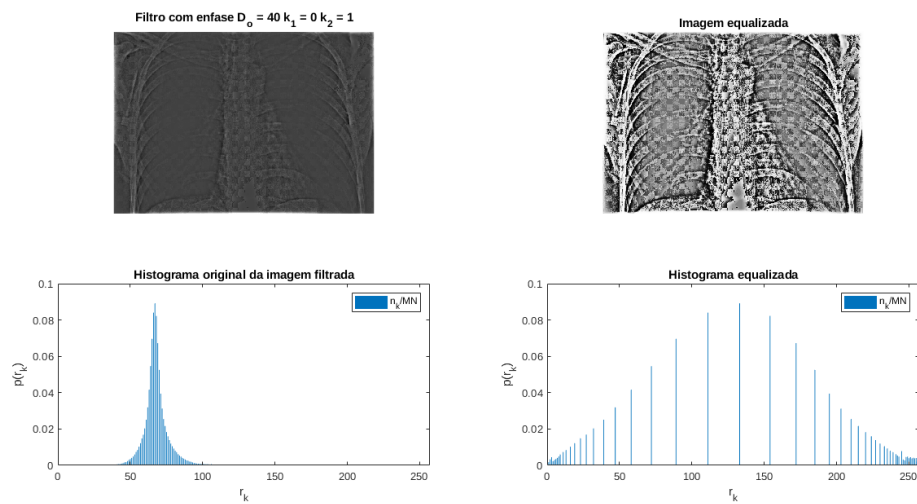
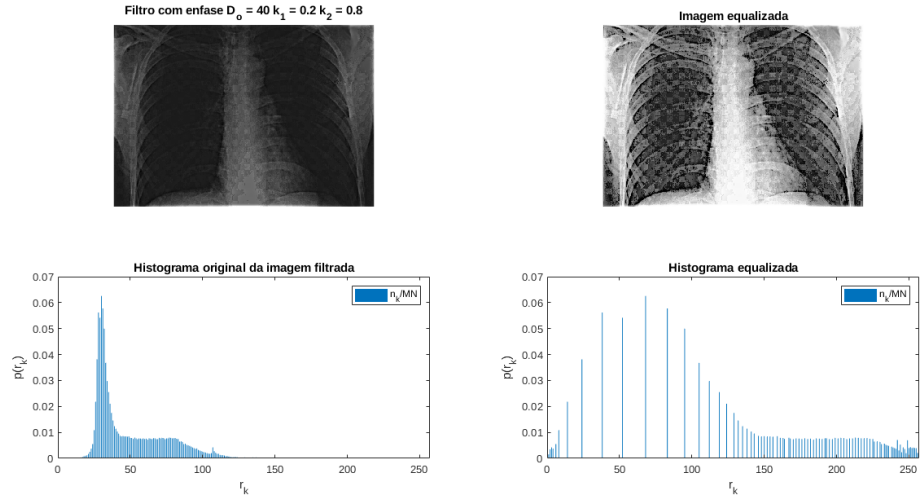
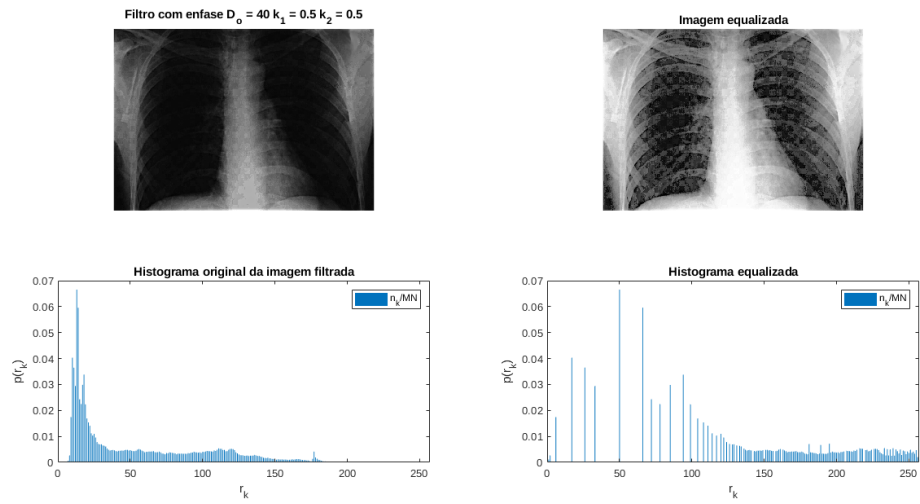
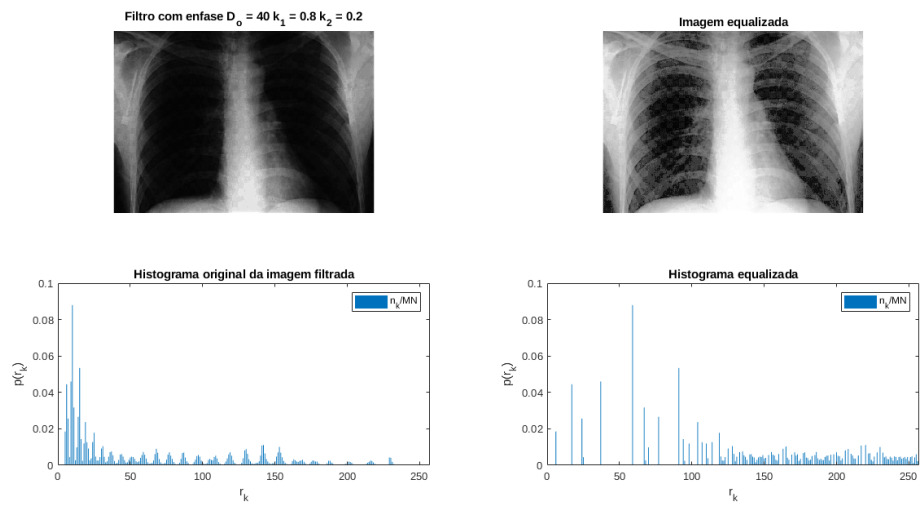


Figura 2: Imagem filtrada

0.2 Aplicação do filtro passa alta de enfase

Nesse filtro há dois parâmetros novos o k_1 que faz esse ajuste de offset da origem e k_2 que controla a contribuição da alta frequência. Após a filtragem foi feita uma equalização da imagem. Os resultados para os valores da Tabela 1 foram:

Figura 3: Imagem filtrada $k_1 = 0$ e $k_2 = 1$

Figura 4: Imagem filtrada $k_1 = 0.2$ e $k_2 = 0.8$ Figura 5: Imagem filtrada $k_1 = 0.5$ e $k_2 = 0.5$ Figura 6: Imagem filtrada $k_1 = 0.8$ e $k_2 = 0.2$

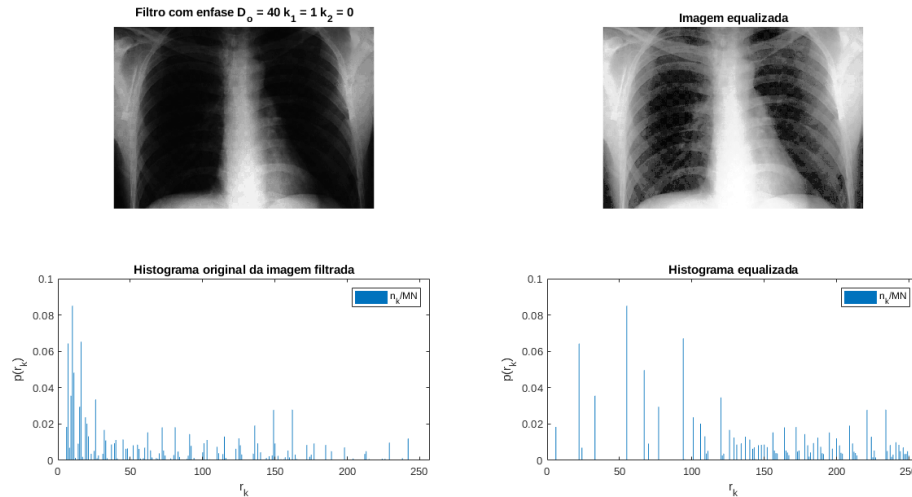


Figura 7: Imagem filtrada $k_1 = 1$ e $k_2 = 0$

Considerações Finais

Considerando os resultados obtidos e o que foi apresentado em sala de aula, temos primeiramente aplicação de um filtro gaussiano passa-alta que atenuava frequências abaixo de 40, significando que somente detalhes como bordas seriam realçadas. Isso ocorreu e devido não haver uma componente DC a imagem resultante ficou escura mesmo com o redimensionamento dos níveis de cinza.

Para a segunda questão sobre o filtro de enfase obtivemos resultados satisfatórios. Para os primeiros valores de k_1 e k_2 obtivemos o mesmo resultado de uma aplicação de apenas o filtro gaussiano, contudo aplicamos uma equalização de histograma para melhorar o contraste com uma redistribuição dos bins. No caso seguinte observa-se pelo histograma que mais valores estão sendo permitidos passarem após a filtragem e acontece um clareamento da imagem devido ao efeito de k_1 . Para o último caso é praticamente a mesma imagem original com o offset de 1 e uma equalização, melhorando o contraste do raio-x.

Referências

- [1] GONZALES, Rafael C.; WOODS, Richard E. Digital image processing. 4. ed. Upper Sadler River, N.J.: Prentice Hall, c2017
- [2] GONZALES, Rafael C.; WOODS, Richard E. Digital image processing. 3. ed. Upper Sadler River, N.J.: Prentice Hall, c2008. 954 p. ISBN 978-0-13-168728-8
- [3] Análise de imagens digitais: princípios, algoritmos e aplicações - 2008 / 2008 PEDRINI, Hélio; SCHWARTZ, William Robson. Análise de imagens digitais: princípios, algoritmos e aplicações. 2008. São Paulo, SP: Thomson Learning, 2008. 508 p.
- [4] Introduction to Digital Image Processing with Matlab – Alasdair McAndrew, Thomson course technology, 2004
- [5] Image Processing – Principles and applications – Tinku Acharya, Ajoy K. Ray, Wiley Interscience, 2005

Apêndice

0.3 Script de Teste

```

1  % FTLO79_DIP_PROJECT_13
2  % Pedro V D S Matias (pvsu@icomp.ufam.edu.br), 15-06-2019 14:29
3  %-----
4  close all,clear all clc;
5  %%
6  f = imread('Fig0459(a)(orig_chest_xray).tif');
7  %a imagem original
8  figure,
9  imshow(f,[]);title('Imagem original');
10 %%
11 Do = 40;
12 g = filtro_gaussiano_alto(f,Do);
13
14 figure,
15 imshow(g,[]);title(['Imagem passa alta D_o = ',num2str(Do)]);
16
17 % Filtro de enfase
18 %% 0 e 1
19 g1 = filtro_enfase(f,Do,0,1);
20 %% 0.2 e 0.8
21 g2 = filtro_enfase(f,Do,0.2,0.8);
22 %% 0.5 e 0.5
23 g3 = filtro_enfase(f,Do,0.5,0.5);
24 %% 0.8 e 0.2
25 g4 = filtro_enfase(f,Do,0.8,0.2);
26 %% 1 e 0
27 g5 = filtro_enfase(f,Do,1,0);
28
29
30 %%
31 g11 = histEqual4e(g1);
32 p1 = imagehist4e(g1,'n');
33 p11 = imagehist4e(g11,'n');
34 figure;
35 subplot(2,2,1); imshow(g1,[],'Border','tight'); title('Filtro com enfase D_o = 40 k_1 = 0 k_2 = 1');
36 subplot(2,2,2); imshow(g11,[],'Border','tight'); title('Imagem equalizada');
37 subplot(2,2,3); bar(p1,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN');title('Histograma original');
38 subplot(2,2,4); bar(p11,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN'); title('Histograma equalizada');
39
40 %%
41 g22 = histEqual4e(g2);
42 p2 = imagehist4e(g2,'n');
43 p22 = imagehist4e(g22,'n');
44 figure;
45 subplot(2,2,1); imshow(g2,[],'Border','tight'); title('Filtro com enfase D_o = 40 k_1 = 0.2 k_2 = 0.8');
46 subplot(2,2,2); imshow(g22,[],'Border','tight'); title('Imagem equalizada');
47 subplot(2,2,3); bar(p2,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN');title('Histograma original');
48 subplot(2,2,4); bar(p22,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN'); title('Histograma equalizada');
49
50 %%
51
52 g33 = histEqual4e(g3);
53 p3 = imagehist4e(g3,'n');
54 p33 = imagehist4e(g33,'n');
55 figure;
56 subplot(2,2,1); imshow(g3,[],'Border','tight'); title('Filtro com enfase D_o = 40 k_1 = 0.5 k_2 = 0.5');
57 subplot(2,2,2); imshow(g33,[],'Border','tight'); title('Imagem equalizada');

```

```

58 subplot(2,2,3);bar(p3,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN');title('Histograma orig
59 subplot(2,2,4);bar(p33,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN'); title('Histograma eq
60
61 %%
62
63 g44 = histEqual4e(g4);
64 p4 = imagehist4e(g4,'n');
65 p44 = imagehist4e(g44,'n');
66 figure;
67 subplot(2,2,1); imshow(g4,[],'Border','tight'); title('Filtro com enfase D_o = 40 k_1 = 0.8 k_2 =
68 subplot(2,2,2); imshow(g44,[],'Border','tight'); title('Imagem equalizada');
69 subplot(2,2,3);bar(p4,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN');title('Histograma orig
70 subplot(2,2,4);bar(p44,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN'); title('Histograma eq
71
72 %%
73 g55 = histEqual4e(g5);
74 p5 = imagehist4e(g5,'n');
75 p55 = imagehist4e(g55,'n');
76 figure;
77 subplot(2,2,1); imshow(g5,[],'Border','tight'); title('Filtro com enfase D_o = 40 k_1 = 1 k_2 = 0
78 subplot(2,2,2); imshow(g55,[],'Border','tight'); title('Imagem equalizada');
79 subplot(2,2,3);bar(p5,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN');title('Histograma orig
80 subplot(2,2,4);bar(p55,0.5);ylabel('p(r_k)');xlabel('r_k');legend('n_k/MN'); title('Histograma eq

```

0.4 Funções implementadas

```

1 function g = filtro_gaussiano_alto(f,Do)
2 %FILTRO_GAUSSIANO_ALTO Realiza a filtragem no dominio da frequencia
3 %   Aplica um filtro gaussiano passa-alta com Do
4
5 %   f = imagem a ser filtrada #1, DO
6 [M,N] = size(f);
7
8 M_new =2*M;
9 N_new =2*N;
10
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 %% computa as coordenadas usadas em filtros centradas em (M/2,N/2)
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14
15 % fixando vetores de coordenadas
16 u = 0:(M_new - 1);
17 v = 0:(N_new - 1);
18
19 % centralizando os vetores de coordenadas
20 u = u - M_new/2;
21 v = v - N_new/2;
22
23 % calculo do array meshgrid
24
25 %No nosso caso:
26 [U,V] = meshgrid(v,u);
27
28
29 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
30 %% Calcula o filtro Gaussiano
31 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32
33 % Calcula as distancias D(U,V)
34 D = sqrt(U.^2 + V.^2);

```

```

35
36 % Deteminacao do filtro H(u,v)
37 H = exp( -(D.^2)./(2*(Do^2)) );
38
39 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40 %% PRE- PROCESSAMENTO
41 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
42
43 % Multiplicar a imagem por (-1)x+y (objetivo: centralizar o espectro de frequência) e realizar o
44
45 f_new = im2double(zeros(M_new,N_new));
46 % padding
47 for x = 1:M
48     for y = 1:N
49         f_new(x,y) = f(x,y);
50     end
51 end
52 %deslocamento da imagem
53 for x=1:M
54     for y=1:N
55         k=x+y;
56         f_new(x,y)=f_new(x,y).*((-1)^k);
57     end
58 end
59
60
61 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
62 %% FILTRAGEM
63 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
64 % Transformada de Fourier
65 F = fft2(f_new);
66
67 % g = G-1 -> G(u,v) = F(u,v)*H(u,v)
68 H1 = 1 - H; %passa-alta
69 g_hf = real(ifft2(H1.*F));
70
71 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
72 %% POS-PROCESSAMENTO
73 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
74 % crop para o tamanho original
75
76 g = g_hf(1:M,1:N);
77
78
79 % compensar o deslocamento de espectro
80 for x=1:M
81     for y=1:N
82         k=x+y;
83         g(x,y)=g(x,y).*((-1)^k);
84     end
85 end
86
87 g = mat2gray(g);
88
89 end
90
91 function g = filtro_enfase(f,Do,k1,k2)
92 %FILTRO_ENFASE Realiza a filtragem no dominio da frequencia
93 % Aplica um filtro gaussiano, com combinação de k1 e k2, passa-alta e com Do
94 % f = imagem a ser filtrada #1, DO= #2, k1=#3,k2=#4
95 [M,N] = size(f);

```



```

6
7 M_new =2*M;
8 N_new =2*N;
9
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 %% computa as coordenadas usadas em filtros centradas em (M/2,N/2)
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13
14 % fixando vetores de coordenadas
15 u = 0:(M_new - 1);
16 v = 0:(N_new - 1);
17
18 % centralizando os vetores de coordenadas
19 u = u - M_new/2;
20 v = v - N_new/2;
21
22 % calculo do array meshgrid
23
24 %No nosso caso:
25 [U,V] = meshgrid(v,u);
26
27
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 %% Calcula o filtro Gaussiano
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31
32 % Calcula as distancias D(U,V)
33 D = sqrt(U.^2 + V.^2);
34 % Deteminacao do filtro H(u,v)
35 H = exp( -(D.^2)./(2*(Do^2)) );
36
37
38 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
39 %% PRE- PROCESSAMENTO
40 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
41
42 % Multiplicar a imagem por (-1)x+y (objetivo: centralizar o espectro de frequência) e realizar o
43
44 f_new = im2double(zeros(M_new,N_new));
45 % padding
46 for x = 1:M
47     for y = 1:N
48         f_new(x,y) = f(x,y);
49     end
50 end
51 %deslocamento da imagem
52 for x=1:M
53     for y=1:N
54         k=x+y;
55         f_new(x,y)=f_new(x,y).*((-1)^k);
56     end
57 end
58
59
60 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
61 %% FILTRAGEM
62 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
63 % Transformada de Fourier
64 F = fft2(f_new);
65

```

```

66 % g = G^-1 -> G(u,v) = F(u,v)*H(u,v)
67 H1 = 1 - H; %passa-alta
68 %disp(H1);
69 H_new = (k1 + k2*H1);
70 %disp(H_new);
71 g_hf = real(ifft2(H_new.*F));
72
73 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
74 %% POS-PROCESSAMENTO
75 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
76 % crop para o tamanho original
77
78 g = g_hf(1:M,1:N);
79
80
81 % compensar o deslocamento de espectro
82 for x=1:M
83     for y=1:N
84         k=x+y;
85         g(x,y)=g(x,y).*((-1)^k);
86     end
87 end
88
89 g = mat2gray(g);
90 g = im2uint8(g);
91 end

```

```

1 function [g] = histEqual4e(f)
2 %HISTEQUAL4E Executa a equalização do histograma em uma imagem de
3 % entrada de 8 bits, f
4 % f = imagem MxN #1
5 % Os níveis de cinza para a imagem original f
6 % e para a imagem equalizada g são representados por rk
7 % e sk, respectivamente, com o 0kL-11
8
9 % Calculo do histograma da imagem a ser equalizada
10 p = imagehist4e(f,'n'); %normalizado
11 L = 256;
12 s = zeros(1,L-1);
13 for k= 1:L
14     % arredondar valor para o nível de cinza mais próximo
15     s(k) = round((L-1)*sum(p(1:k)));
16 end
17
18 [M,N] = size(f);
19 g = zeros(M,N);
20
21 for x = 1:M
22     for y = 1:N
23         %obter a imagem final equalizada
24         g(x,y) = s(f(x,y)+1);
25     end
26 end
27 imagehist4e(g,'n');

```

```

1 function h = imagehist4e(f,mode)
2 %IMAGEHIST4E Histograma de uma imagem em tons de cinza
3 % Processa o histograma de uma imagem , f, de 256 niveis de cinza,
4 % cujas as intensidades são não-negativas.
5 % Se mode='n' o histograma deve ser normalizado
6 % Se mode='u' deve ser desnormalizado.

```

```

7  % f = imagem #1; mode = normalização #2.
8
9  % Pegando as dimensões da imagem
10 [M,N] = size(f);
11 h = zeros(1,256);
12
13 for x = 1:M % Varredura vertical
14     for y = 1:N % Varredura horizontal
15         % Pega o valor da intensidade do pixel atual
16         r = f(x,y) + 1;
17         % Incrementa no bin da intensidade correspondente. Do 1 ate
18         % 256
19         h(r) = h(r) + 1;
20     end
21 end
22 switch mode
23     case 'n' % Hist. Normalizado
24
25         h = h/(M*N);
26         % bar(h,0.5);
27         % title('p(r_k) = n_k/MN versus r_k ');
28         % ylabel('p(r_k)');
29         % xlabel('r_k');
30         % legend('n_k/MN');
31
32
33     case 'u' % Hist. Não Normalizado
34
35         % bar(h,0.5);
36         % title('h(r_k) = n_k versus r_k ');
37         % ylabel('h(r_k)');
38         % xlabel('r_k');
39         % legend('n_k');
40
41
42     otherwise
43         warning('modo não definido, somente Normalized (n) ou unnormalized (u)')
44
45 end

```