

EEL 7825 - Projeto Nível II em Cont e Proc de Sinais I
Daniel Augusto Figueiredo Collier
Exercícios - Aula 4

1. Duas imagens degradadas são fornecidas: B = caverna.pgm e C = russia.pgm.



Figura 1: Caverna.pgm



Figura 2: Russia.pgm

Filtros	sobel	prewitt	laplace1	sh1
sh2	low-pass	Mean	circular,rad	circular-mean,rad

Antes de mostrar os resultados obtidos, valem algumas observações: os filtros *circular* e *circular-mean* apresentam diferença apenas de amplitude e por isso não há diferenças entre os resultados obtidos nas suas aplicações; é esperado que os filtros de característica passa-alta não melhore as imagens já que elas encontram-se deterioradas por componentes de altas frequências (ruído e bordas); a apresentação das imagens processadas com filtragem passa-alta são exibidas somadas a imagem original, isto é:

imagem exibida = imagem original + resultado do filtro passa-alta

- A imagem B foi comprimida usando o algoritmo JPEG de alta compressão, aplique os filtros disponíveis e conclua qual o melhor para este caso.

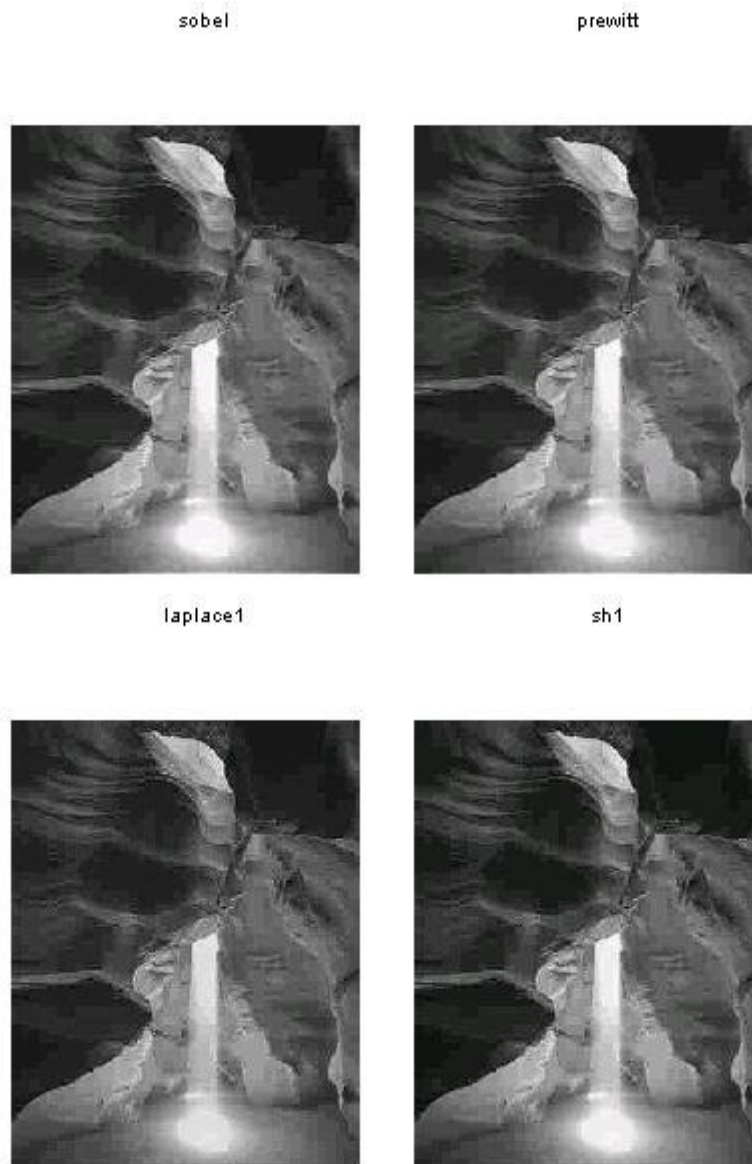


Figura 3: Aplicação dos filtros na figura B.

sh2

low-pass



mean

circular.rad=1



Figura 4: Aplicação dos filtros na figura B.

circular.rad=2



circular.rad=3



Figura 5: Aplicação dos filtros na figura B.

Fiquei entre 3 filtros utilizados por apresentarem resultados com qualidade e bastante parecida. Os filtros foram: *low-pass*, *mean* e *circular* com *rad=1*.

- A imagem C está corrompida por ruído, aplique os filtros disponíveis e conclua qual o melhor para este caso.

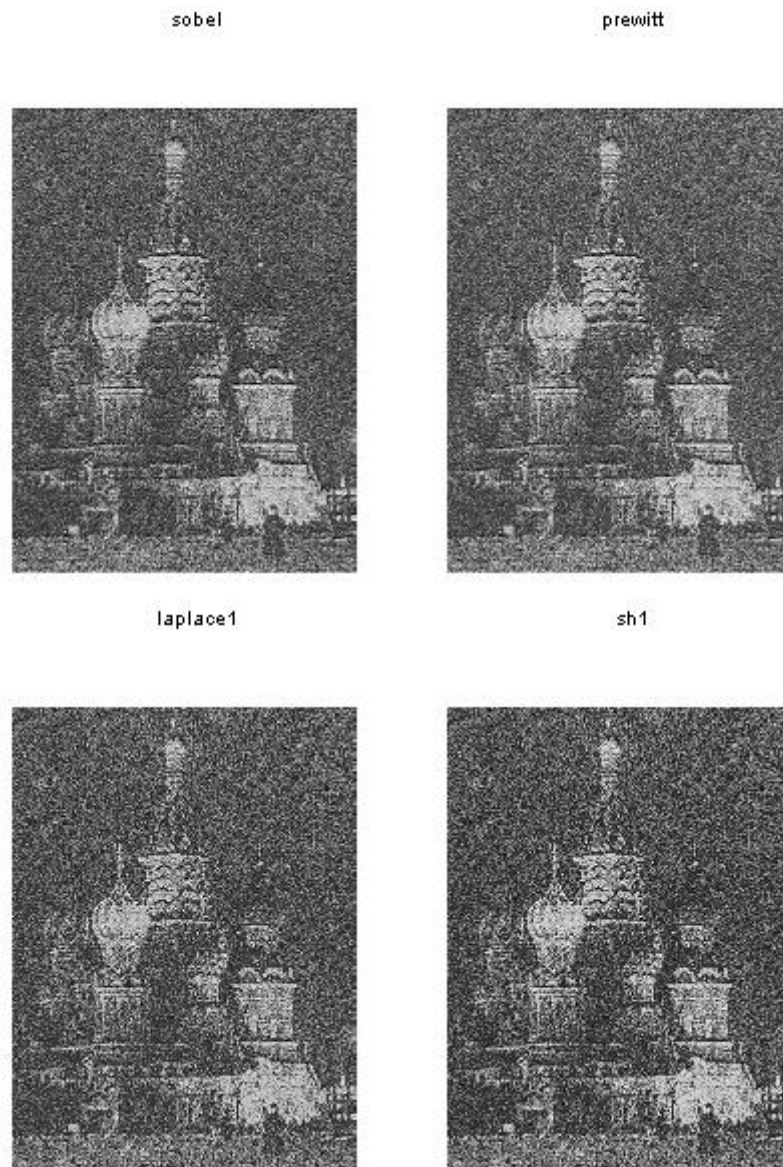
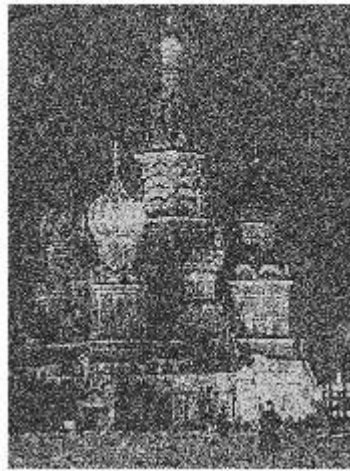


Figura 6: Aplicação dos filtros na figura C.

sh2



low-pass



mean



circular.rad=1



Figura 7: Aplicação dos filtros na figura C.

circular.rad=2



circular.rad=3



Figura 8: Aplicação dos filtros na figura C.

Fiquei entre 2 filtros utilizados por apresentarem resultados com qualidade e bastante parecida. Os filtros foram: *low-pass* e *circular* com $rad=1$.

- Implemente o filtro de mediana descrito na página 191 do livro texto. Após implementá-lo, verifique seu funcionamento nas imagens B e C. Comente o resultado.



Figura 9: Mediana com vizinhança 3x3.



Figura 10: Mediana com vizinhança 5x5.



Figura 11: Mediana com vizinhança 3x3.



Figura 12: Mediana com vizinhança 5x5.

A aplicação da filtragem mediana na figura imagem B apresentou um resultado de boa qualidade preservando as formas e diminuindo os efeitos da sua degradação. Na imagem C os resultados apresentaram uma boa redução do ruído, mas a mediana com vizinhança 5x5 apresentou um acentuado efeito passa-baixa deixando borrada a imagem.

A listagem do programa:

```
function M = mediana(M, v)
// MEDIANA_
// M: imagem em tons de cinza
// v: vizinhança do tipo v x v, 3 ou 5
//
// Uso:
// I = imread('figura.jpg');
// M = mediana(I, 3);
// imshow(M)
//
[m n] = size(M);
// análise do tamanho da vizinhança
if (v==3)
    d = 1;
elseif (v==5)
    d = 2;
end
//
m1 = m + 2*d;
n1 = n + 2*d;
p0 = 1 + d;
X = -ones(m1,n1);
X(p0:(m1-d),p0:(n1-d)) = M;
//
for i=p0:(m1-d)
    for j=p0:(n1-d)
        N = X(i-d:i+d,j-d:j+d);
        if ( max(1*mtlb_any(N==-1)) )
            p = sum(1*((-ones(v,v))==(N)));
            s = N(:);
            M(i-d,j-d) = median(s(1:(v^2-p)));
        else
            M(i-d,j-d) = median(N(:));
        end
    end
end
//
M = round(M);
M = M - min(M);
M = M / max(M);

endfunction
```

Desafio: Embora não tenhamos trabalhado com a FFT, existe uma técnica de melhoramento de imagens bastante interessante, *Filtragem Homomórfica* (páginas 213218), que faz uso das propriedades da Transformada de Fourier para o melhoramento de imagens. A implementação desta técnica é simples e pode ser aplicada sobre a imagem farol.pgm (cuja imagem original é farol-g.pgm). Com parâmetros adequados, o resultado desta técnica fica explícito nesta figura. Implemente a filtragem homomórfica e tente obter, a partir da imagem farol.pgm, uma imagem mais próxima possível da imagem original farol-g.pgm.

Estou colocando a implementação que eu fiz. Tive alguns problemas em saber se está 100% correta porque não consegui parâmetros adequados para obter o resultado esperado. Sei que o espectro do filtro tem que ficar como na Figura 13.

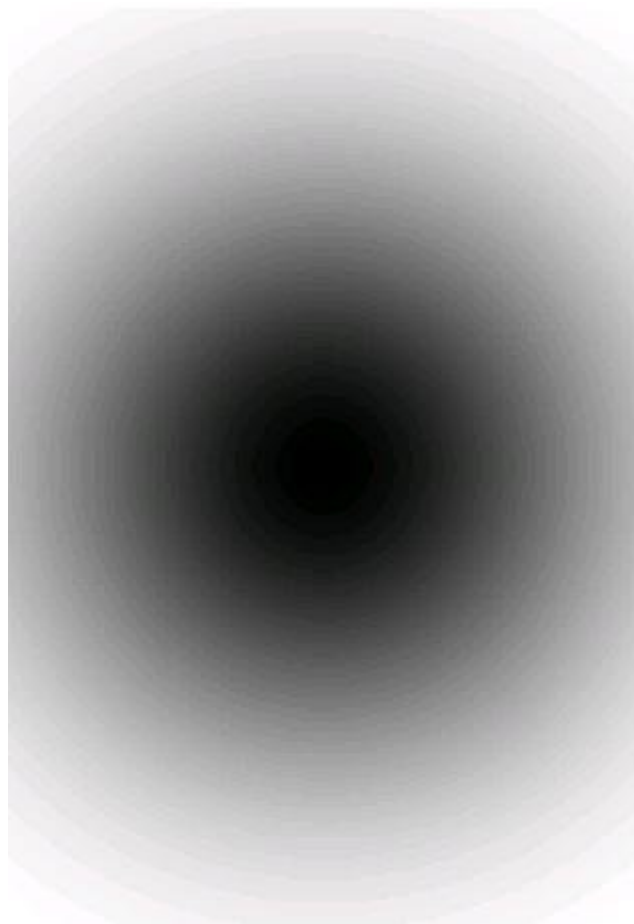


Figura 13: Espectro do filtro usado na filtragem homomórfica.

O resultado da filtragem com parâmetros $H = \text{homomorphic}(I, 0.3, 4, 90, 1)$ é apresentado na Figura 14.

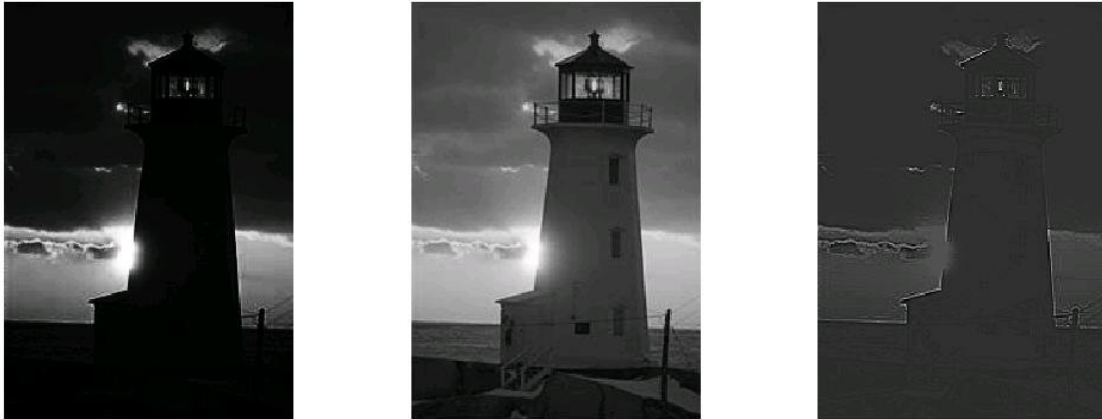


Figura 14: A última figura é o resultado da filtragem homomórfica.

A listagem do programa:

```
function H = homomorphic(I, gL, gH, D0, c)
// HOMOMORPHIC_
// I: imagem em tons de cinza
// gL: parâmetro da curva do filtro homomórfico, gL < 1
// gC: parâmetro da curva do filtro homomórfico, gH > 1
// D0: frequência de corte do filtro
// c: parâmetro de controle de "sharpness"
//
// Uso:
// I = imread('figura.jpg');
// H = homomorphic(I,0.7,5,200,1);
// imshow(H, [])
//
I = I - min(I);
I = I / max(I);
[m n] = size(I);
// Aplicação de ln
H = log(1+I);
// Cálculo da fft
H = fft2(H);
// Projeto do Filtro
u = [1:round(m/2)]'*ones(1,round(n/2));
v = ones(round(m/2),1)*[1:round(n/2)];
DUV = u.^2 + v.^2;
//gH = 2; gL = 0.5; c = 0.5; D0 = 120;
Fi = (gH - gL)*( 1-exp(-c*(DUV)/(D0^2)) ) + gL;
Fi = [usd(Fi) usd( Fi(:,round(n/2):-1:1)) ; Fi(:,round(n/2):-1:1)
Fi];
Fi = fftshift(Fi); // desfaz espectro simétrico
Fi = Fi(1:m,1:n);
// Aplicação do filtro
H = H.*Fi;
// Cálculo da ifft
H = real( fft2(conj(H)) ) / (m*n);
```

```
// Aplicação de exp
H = exp(H);
//
H = H - min(H);
H = H / max(H);

endfunction
```