tooEEL 7825 - Projeto Nível II em Cont e Proc de Sinais I
Daniel Augusto Figueiredo Collier
Exercícios - Aula 3

1. Quatro imagens degradadas são fornecidas:

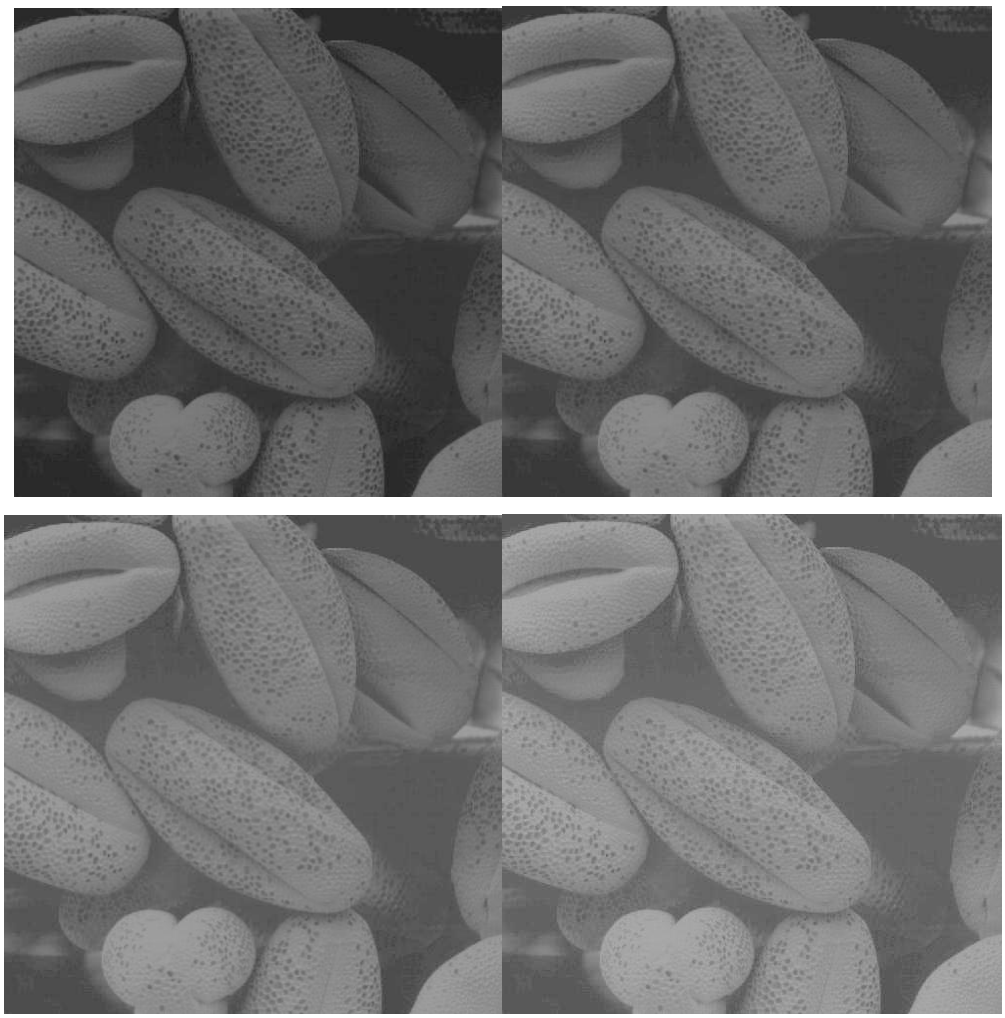
   A = farol.jpg B = pollen_d.jpg C = aerial.jpg D = surf.jpg

   As imagens acima encontram-se degradadas devido à iluminação inapropriada, ou limitação/mal regulagem dispositivo de captura. Aplique correções gama apropriadas que tragam o melhor resultado possível. Teste para diversos valores e anote as escolhas. Faça um relato breve descrevendo seu resultado (apresente as figuras).

   As escolhas dos valores para os testes foram feitas da seguinte maneira: para imagens escurecidas valores de gama menor que um e, para imagens claras valores de gama maior que um. Para a imagem

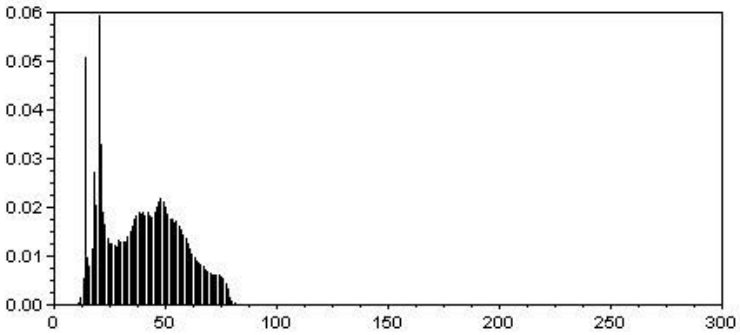
- A = farol.jpg. Foi possível melhorar a qualidade da imagem aplicando a correção gama. É apresentado imagens com correções de 0.8, 0.6, 0.4 e 0.3, respectivamente. A correção com gama igual a 0.4 ficou boa.

- B = pollen_d.jpg. É apresentado imagens com correções de 0.6, 0.5, 0.4 e 0.35, respectivamente. A correção gama não apresentou bons resultados, continuando as imagens com a mesma quantidade de detalhes. Esse efeito foi visualizado nos histogramas das imagens que mantiveram um agrupamento praticamente constante.

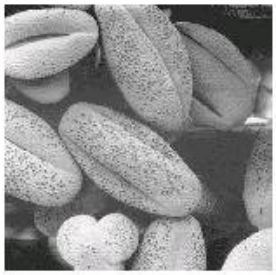


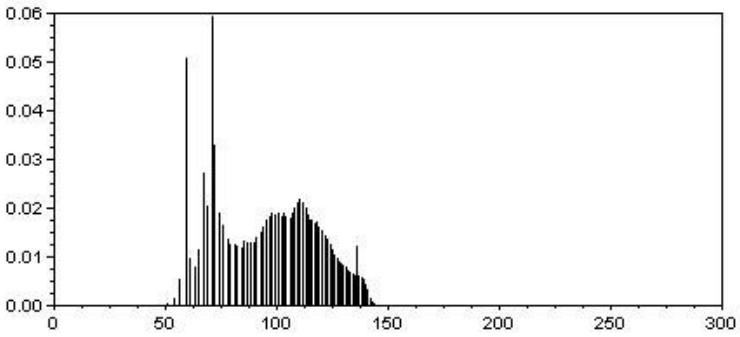
Os histogramas mostram que não há muita diferença na distribuição dos pixels nas imagens. Não apresentando a melhoria esperada.

pollen_d.jpg

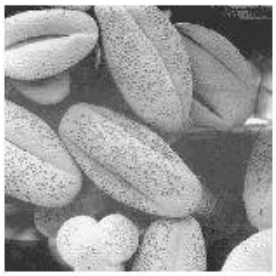


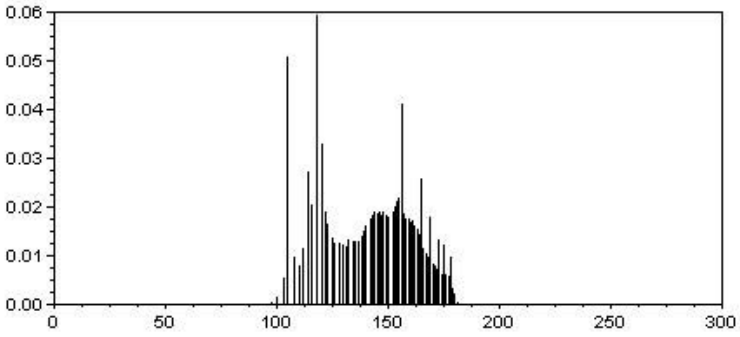
gama = 0.5





gama = 0.3





- C = aerial.jpg. Foi possível melhorar a qualidade da imagem aplicando a correção gama. É apresentado imagens com correções de 2, 4 e 5, respectivamente. A correção com gama igual a 5 ficou boa por apresentar uma quantidade maior de detalhes no fundo da imagem.

- D = surf.jpg. Foi possível melhorar a qualidade da imagem aplicando a correção gama. É apresentado imagens com correções de 0.5, 0.4, 0.35 e 0.3, respectivamente. A correção com gama igual a 0.3 ficou boa.

2. Temos a disposição no scilab um comando para visualizar o histograma de uma imagem, porém, se desejarmos processar de alguma forma este histograma isso não é possível, pois não temos seus valores armazenados. Crie um programa que compute o histograma de uma imagem e armazene os dados em um vetor.

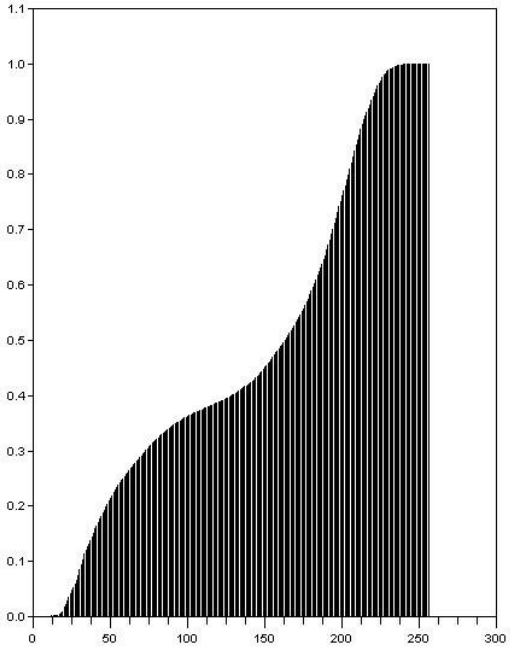
Para calcular o histograma da imagem calcula-se a freqüência de cada pixel (*nk*) na imagem e divide-se cada valor pela quantidade total (*n*) de pixels na imagem. O valor em cada ponto *k* do histograma é:

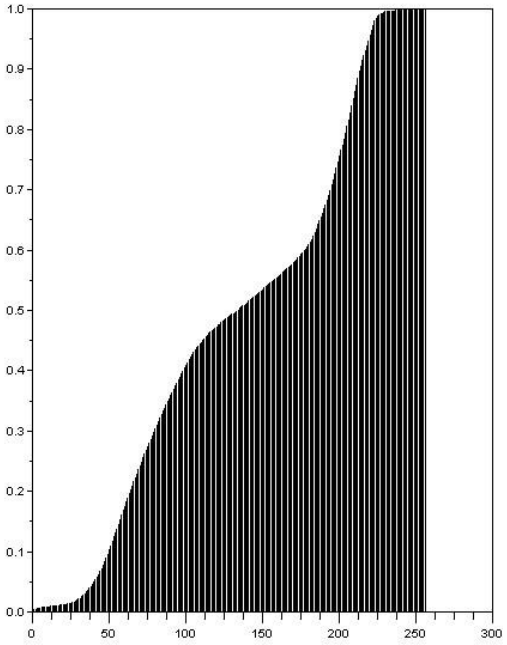
$$P(r_k) = \frac{n_k}{n}$$

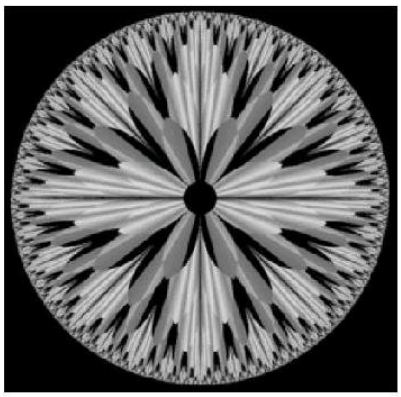
A listagem do programa:

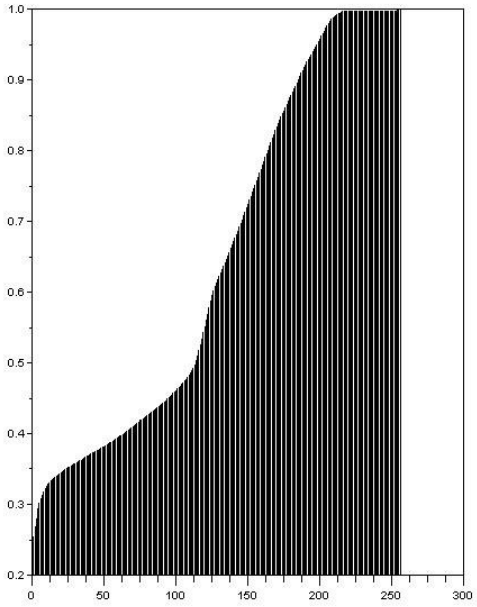
```
function h = histograma(x, I)
// HISTOGRAMA_
// h: histograma normalizado da imagem
// x: número de varições de intensidade na imagem (256 ou 2)
// I: imagem: em tons de cinza de 1 a 256, ou binária
// Uso:
// I = imread('figura.jpg');
// h = histograma(256, I);
// plot2d3(1:256,h) // plotar o histograma em barras
//
[m n]= size(I);
f = zeros(1,x);

I = int(I);
// calcula a freqüência (histograma) dos pixels na imagem
for i=1:m
  for j=1:n
    f(I(i,j)) = f(I(i,j)) + 1;
  end
end
// normaliza o histograma
h = f/(m*n);

endfunction
```

3. Para a equalização de histograma é necessário calcular o histograma cumulativo. Supondo um histograma com X componentes, o histograma cumulativo é a soma dos componentes do histograma, do primeiro (intensidade 1) até a posição X (que no máximo, é 256). Faça um programa para calcular o histograma cumulativo.









A listagem do programa:

```
function c = cum_hist(h)
// CUM_HIST_
// c: histograma cumulativo do histograma
// h: histograma
//
// Uso:
// I = imread('figura.jpg');
// c = cum_hist(h);
// plot2d3(1:256,c)
//
c = cumsum(h);

endfunction
```

4. Na seção 4.2.4 do livro texto é apresentada uma forma de melhoria de imagens bastante interessante. A média de imagens ruidosas é uma técnica matematicamente explicável e que possui resultados ótimos quando lidamos com certos tipos de ruídos. Para este exercício faça o seguinte:

```
--> im = imread('ararauna.pgm');
-->[M,N] = size(im);
//geração de ruído gaussiano
-->noise = sqrt(1000)*rand(M,N,'normal'); noise = noise-
mean(noise);
-->im_noise = im + noise;
```

Agora, com a imagem ruidosa im_noise e o equacionamento apresentado na seção 4.2.4 do livro texto, gere um conjunto de imagens similar ao apresentado na figura 4.18 do livro. Ou seja, o resultado da mediação de 1, 2, 8, 16, 32 e 128 imagens ruidosas. Compare a imagem original com todas as 6 versões geradas e apresente suas conclusões.

Para gerar a imagem ruidosa com o equacionamento criei a função a*verage* que retorna a imagem com a média do ruído para *n* iterações.

Os resultados mostraram que a diferença (pelo menos perceptiva) para os valores 32 e 128 não foi muito grande. O que já era esperado uma melhoria da qualidade da imagem com o aumento das médias.

A listagem do programa:

```
function A = average(I, n)
// AVERAGE_
// I: imagem em tons de cinza
// n: número de iterações
//
// Uso:
// I = imread('figura.jpg');
// A = average(I, 8);
// imshow(A, [])
//
y = sqrt(1000);
s = size(I);
I = I(:);
// adiciona média de ruídos gaussianos a imagem
noise = y*rand(prod(s), n, 'normal');
noise = noise * ones(n,1);
A = I + noise./n;
//  ajuste  da  matriz  para  retornar  como  matriz  de
intensidade entre 0 e 1
A = matrix(A, s);
A = A - min(A);
A = A / max(A);

endfunction
```