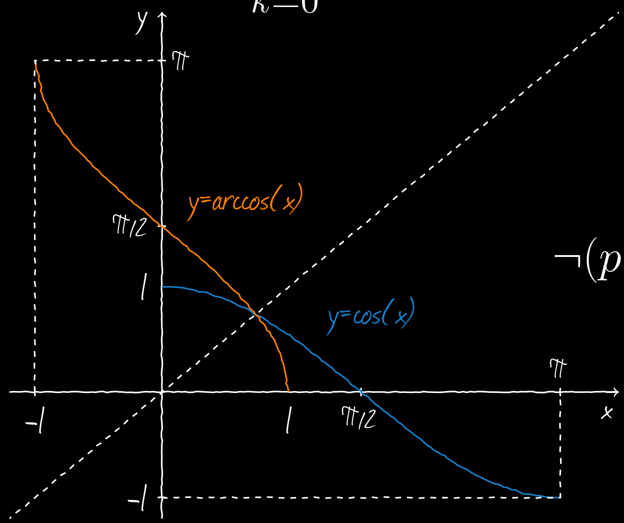
Report for
Digital Image Processing

Fanyong Xue

version 1.1

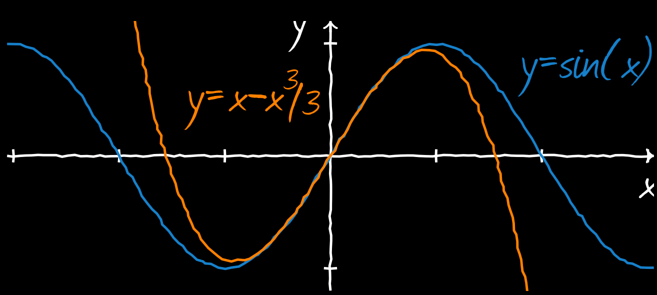
$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}$$



$$\zeta_k = |a|^{1/n} \mathrm{e}^{i(\arg(a)+2k\pi)/n}$$

$$\mathrm{e}^{i\pi} + 1 = 0$$

$$\neg(p \vee q) \equiv (\neg p) \wedge (\neg q)$$



$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Contents

Contents

1.1 OverView

(a) Write a computer program for computing the histogram of an image.
(b) Implement the histogram equalization technique.
(c) Your program must be general to allow any gray-level image as its input.
As a minimum, your report should include the original image, a plot of its histogram, a plot of the transformation function, the enhanced image, and a plot of its histogram.

1.2 Generate the Histogram

1.2.1 Function

Generating the histogram of an image using following function:

$$H(i) = the\ number\ of\ pixel\ whose\ value\ euquals\ to\ i$$

1.2.2 Histogram

The histogram pictures of Fig1.jpg and Fig2.jpg are listed as follows:

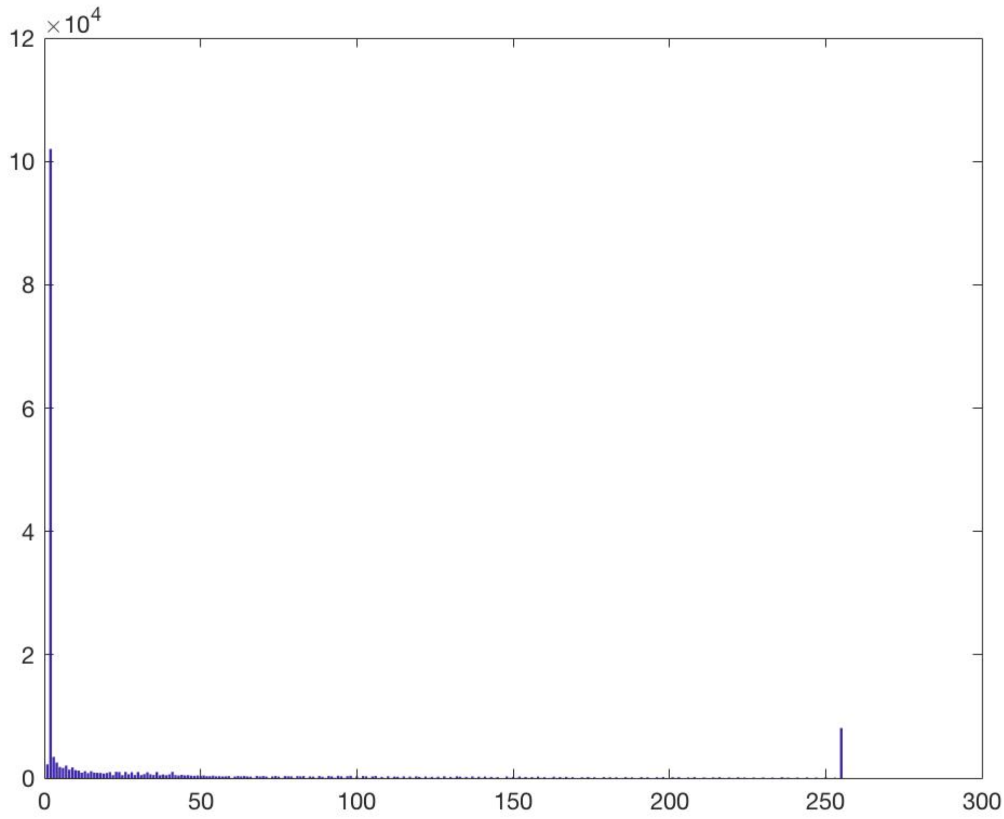


Figure 1.1: Histogram of fig1.jpg

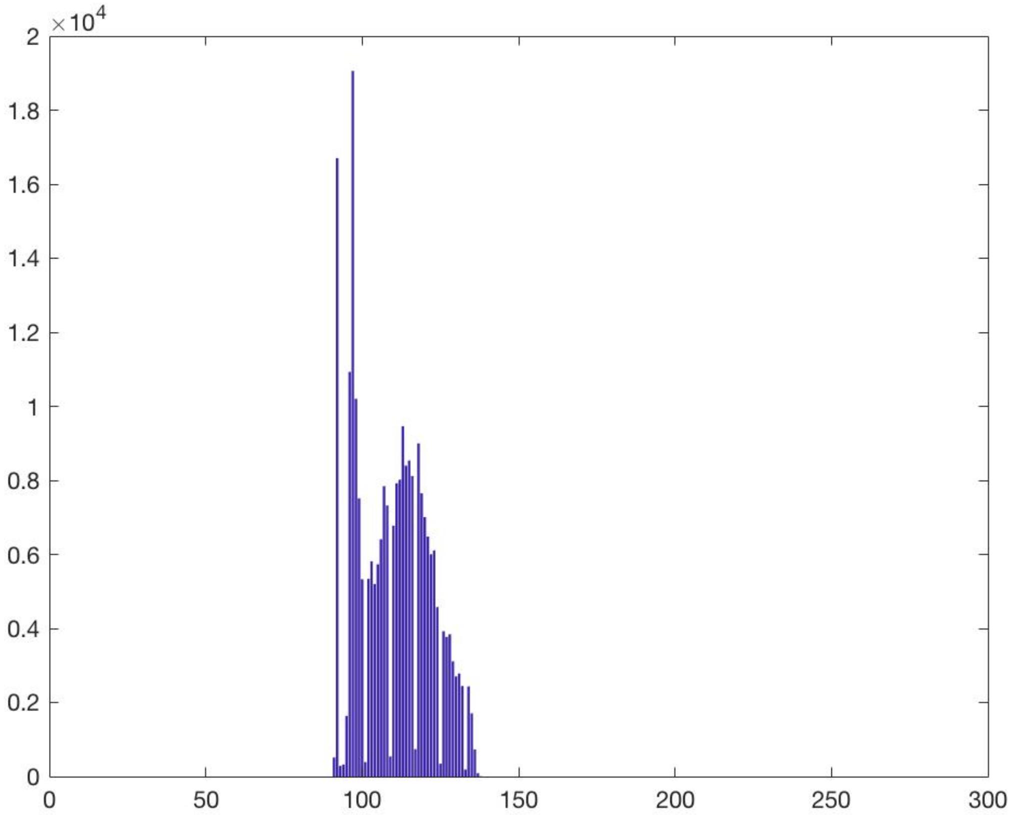


Figure 1.2: Histogram of fig2.jpg

1.3 Transfer Function

1.3.1 Implement the Histogram Equalization Technique

We use those functions to calculate the histogram equalization:

$$L = Max(image(r,c)) \; \forall r \in [1, rows] \; and \; \forall c \in [1, cols]$$
$$s(r_k) = L * T(r_k) = L * \sum_{j=0}^{k} P_r(r_j) = L * \sum_{j=0}^{k} \frac{n_j}{n}$$

1.3.2 Transfer Function

The transfer function of Fig1.jpg and Fig2.jpg are listed as follows:

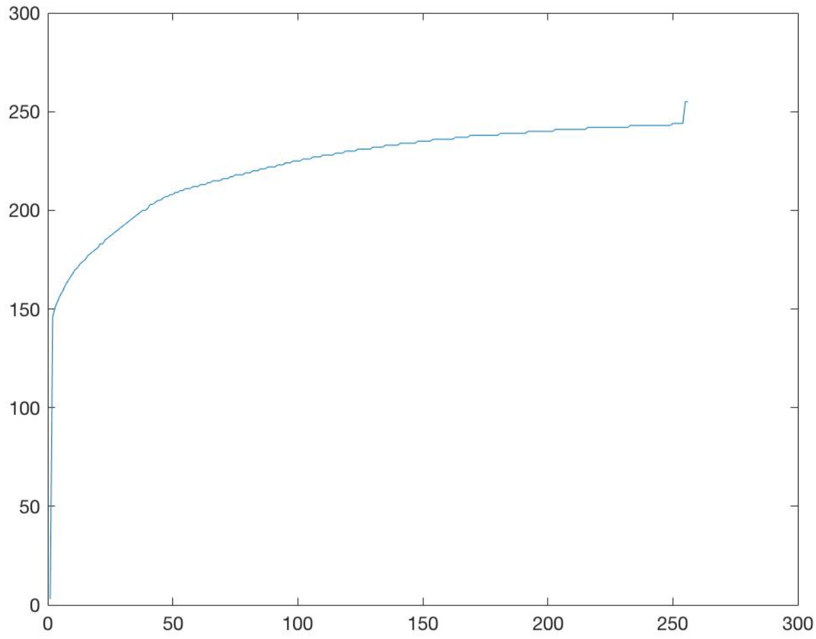


Figure 1.3: Transfer Function of fig1.jpg

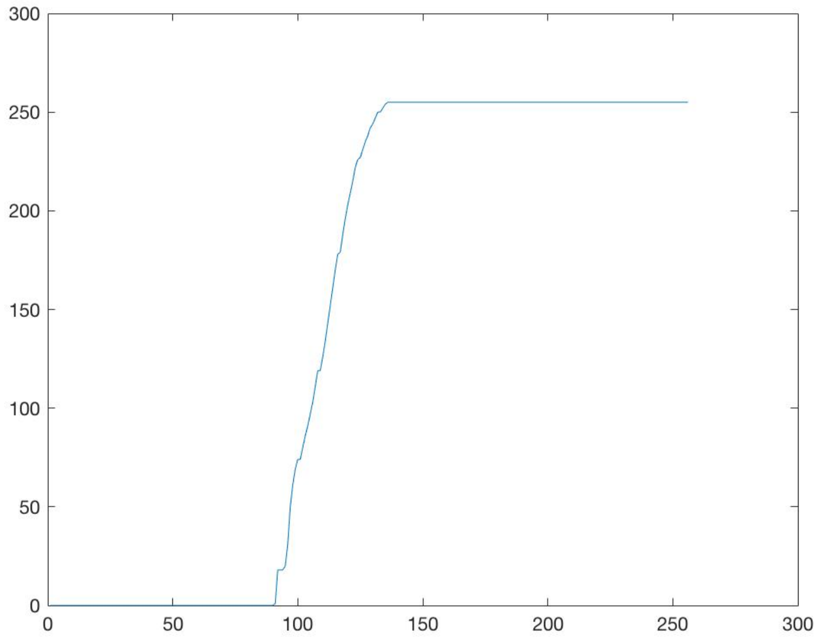


Figure 1.4: Transfer Function of fig2.jpg

1.4 Enhanced Images

1.4.1 Enhanced Function

We use the function

$$New\ Image(r, c) = Transfer\ Function(image(r, c))\ \forall r \in [1, rows]\ and\ \forall c \in [1, cols]$$

to enhance the original images.

1.4.2 Enhanced Images

The original images and enhanced images and histogram comparation are listed as follows.

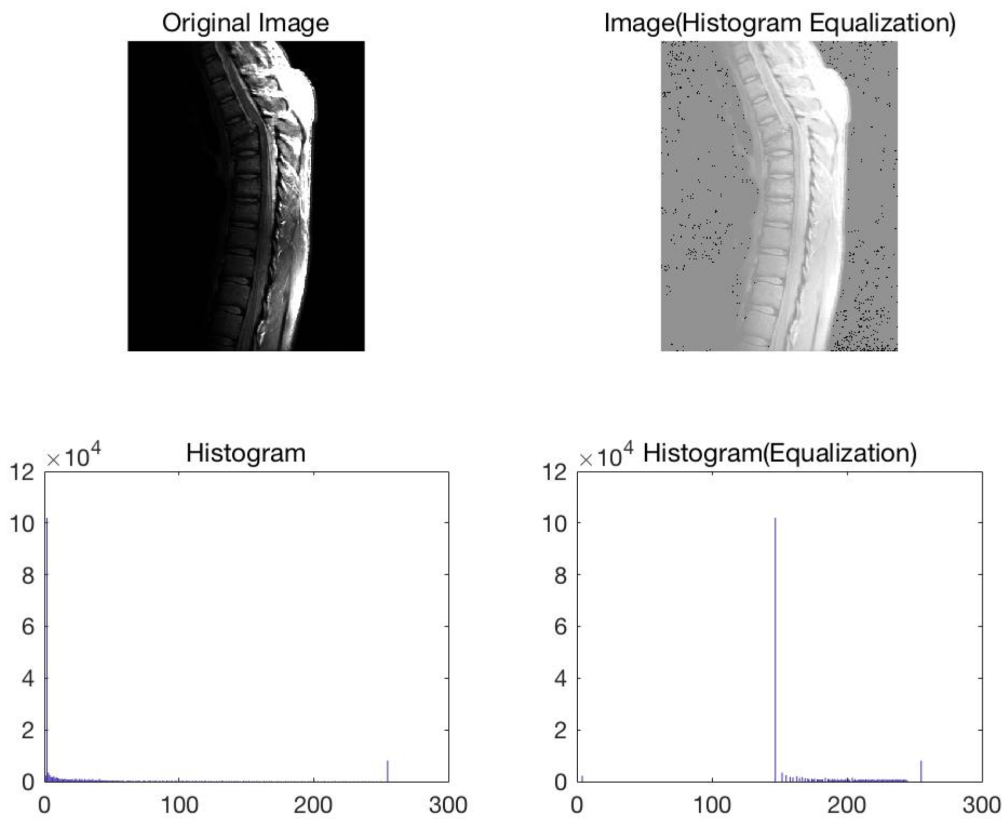


Figure 1.5: original image and enhanced image and histogram comparation of fig1.jpg

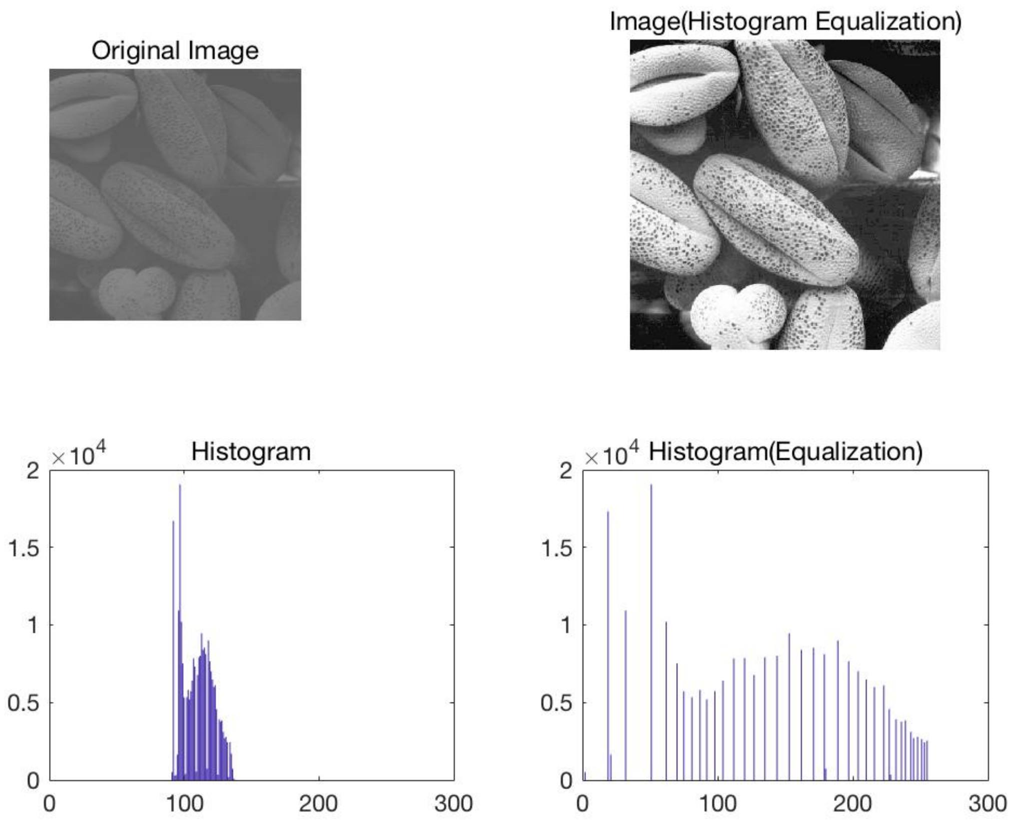


Figure 1.6: original image and enhanced image and histogram comparation of fig2.jpg

2.1 OverView

Implement the image enhancement task of Section 3.7 (Fig 3.43, page 171). The image to be enhanced is skeleton_orig.tif. You should implement all steps in Figure 3.43. (You are encouraged to implement all functions by yourself, not to directly use Matlab functions such as imfilter or fspecial.)

2.2 Image b

2.2.1 Laplacian Transform Filter

We apply Laplacian Transform on the original image to get the image (b) using the following filter.

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

2.2.2 Laplacian Transform

Image b:

Figure 2.1: Laplacian Transform of skeleton_orig.tif

2.3 Image c

The we add the Laplacian of the original image to the original image, we will get the new image c. The new image c is a rather noisy sharpened image.
Image c:

Figure 2.2: Laplacian Transform of skeleton_orig.tif

2.4 Image d

2.4.1 Sobel Gradient Masks

We will use two mask to separately get the components g_x and g_y. Then add the two components together, we will get the the sober gradient of the original image. The new image is as follows. As we can see, edges are much more dominant in this image than in the Laplacian image.

g_x:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

g_y:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

2.4.2 Image d

Image d:

Figure 2.3: Laplacian Transform of skeleton_orig.tif

2.5 Image e

Image e is formed by smoothing image d by 5*5 mean filter.
Image e:

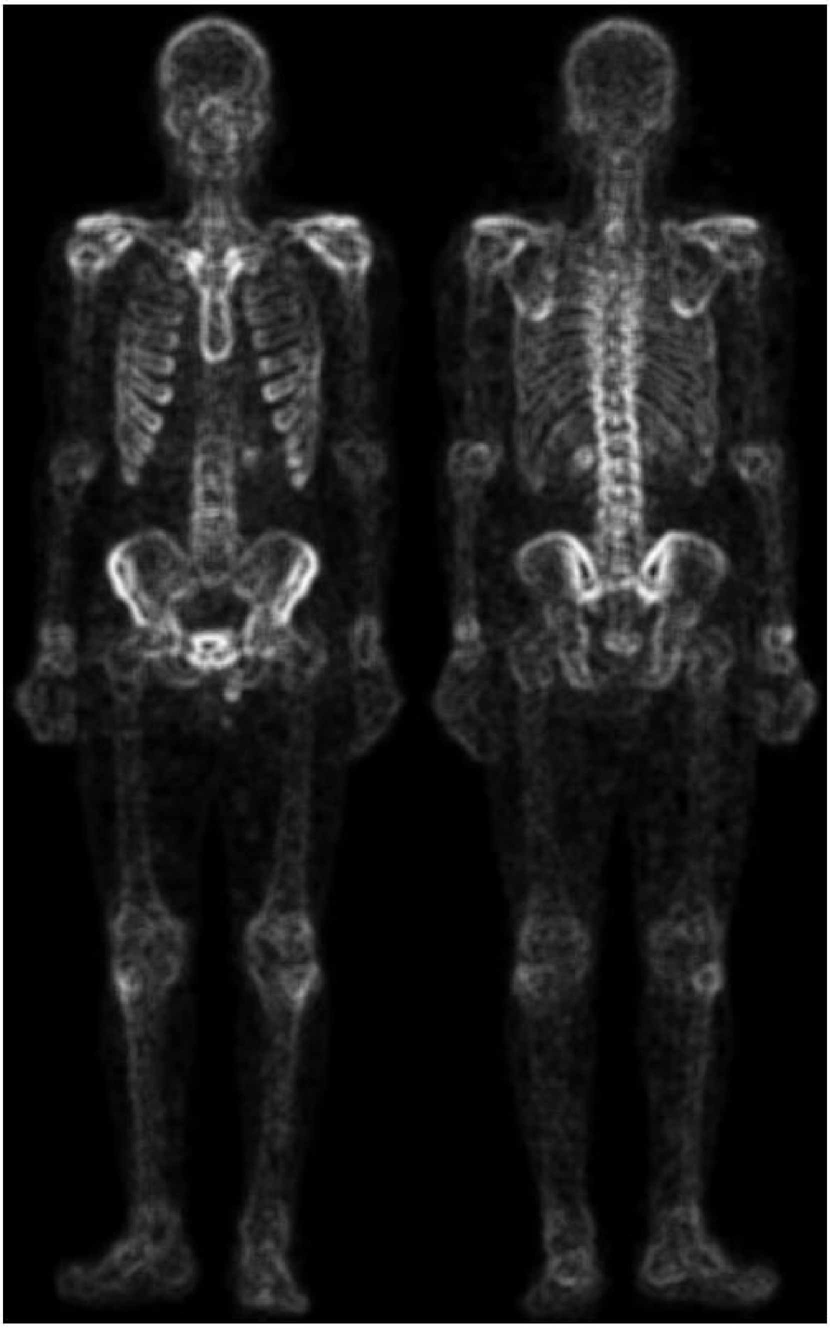


Figure 2.4: Laplacian Transform of skeleton_orig.tif

2.6 Image f

Image f is formed by the product of Laplacian and smoothed-gradient image. The dominance of the strong edges and the relative lack of visible noise, which is the key objective behind masking the Laplacian with a smoothed gradient image.
Image f:

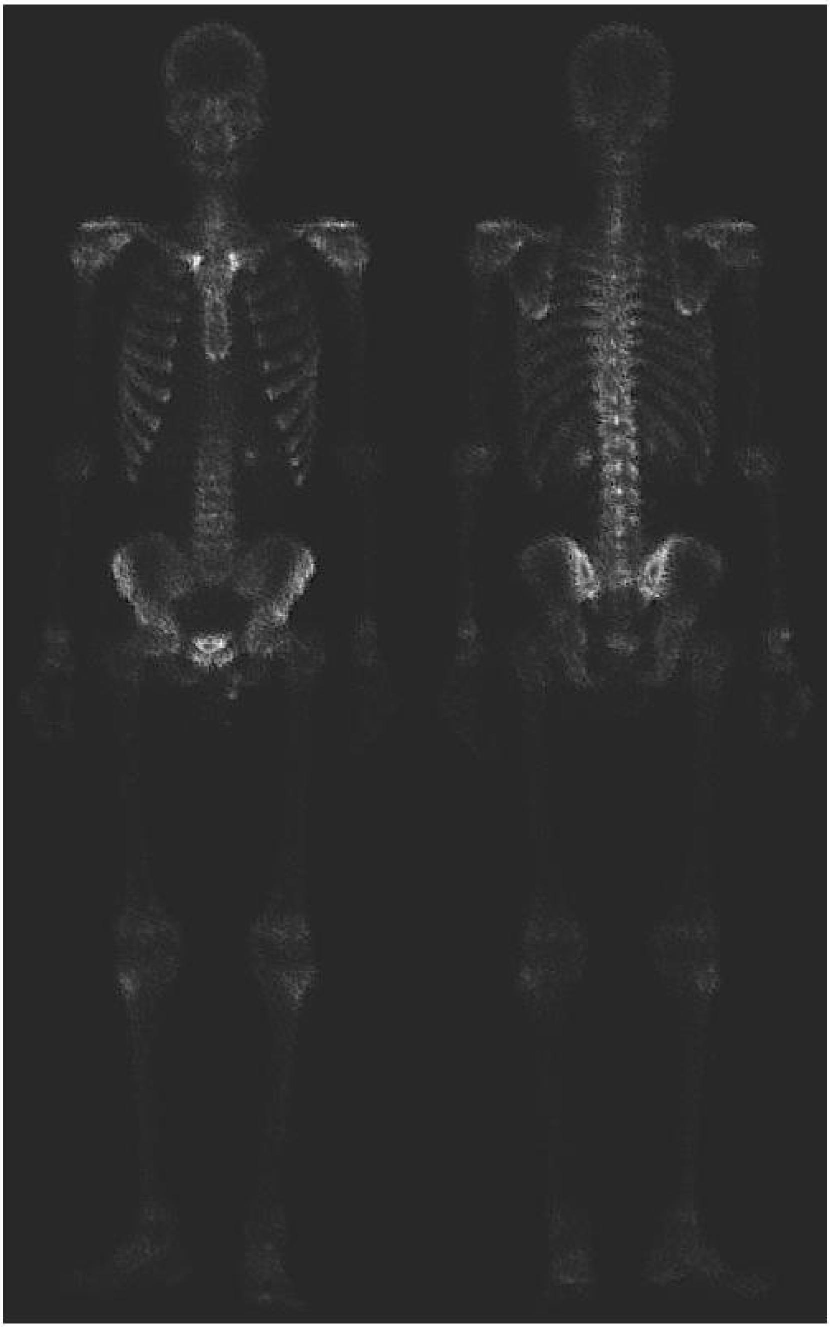


Figure 2.5: Laplacian Transform of skeleton_orig.tif

2.7 Image g

Adding the image f to the original image and then we get image g.
image g:

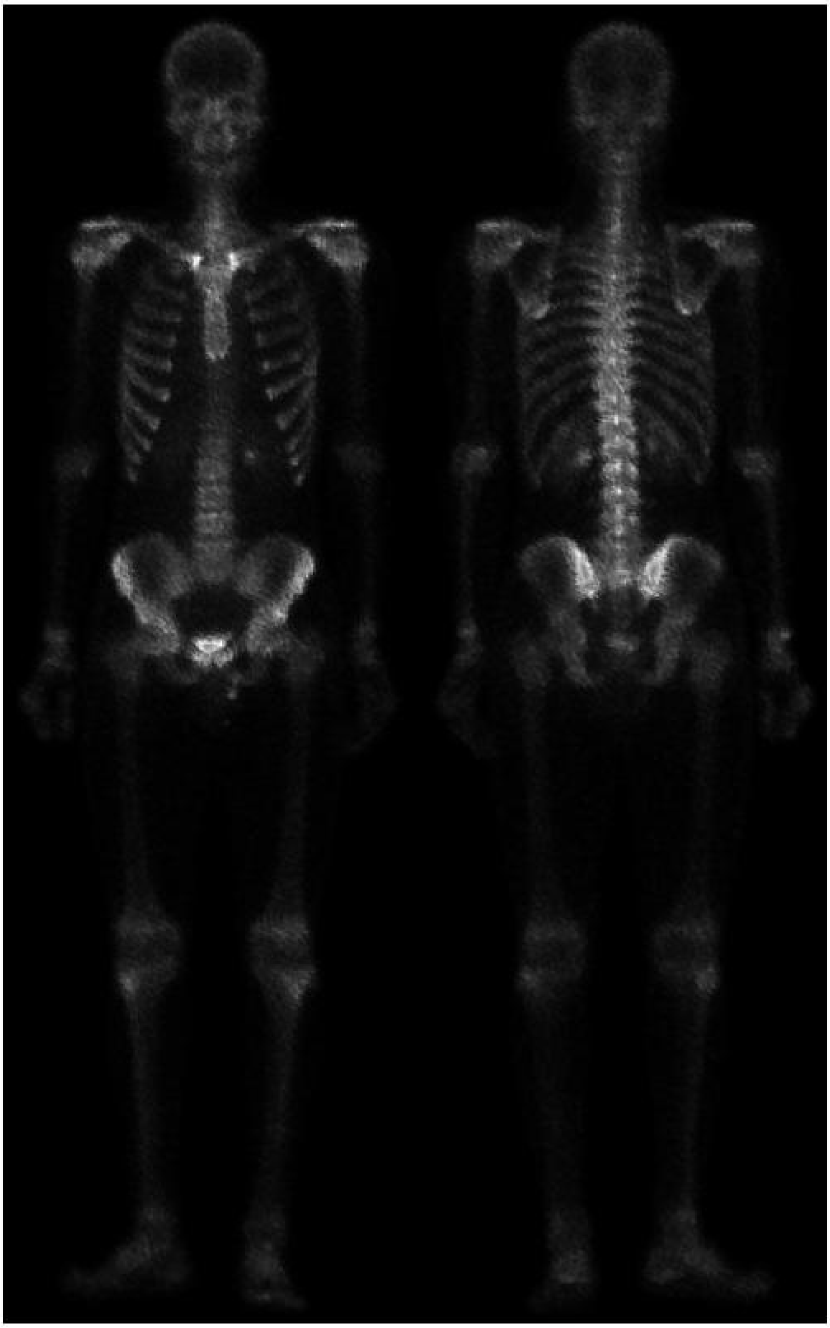


Figure 2.6: Laplacian Transform of skeleton_orig.tif

2.8 Image h

2.8.1 Power-Law Transformation

We use the following function to perform Power-Law Transformation on image g.

$$s = cr^{\gamma} \ (c = 1 \ and \ \gamma = 1)$$

2.8.2 Image h

Image h:

Figure 2.7: Laplacian Transform of skeleton_orig.tif

3.1 OverView

Implement the ideal, Butterworth and Gaussian lowpass and highpass filters and compare the results under different parameters using the image characters_test_pattern.tif (this image file can be found at the ftp server ftp://ftp.cs.sjtu.edu.cn:990/lu-ht/DIP/images) as the test pattern.

3.2 This is a section

3.2.1 This is a subsection

3.2.2 This is a subsection

4.1 OverView

In this problem, you are required to write a program to generate different types of random noise (Uniform, Gaussian, Rayleigh, Gamma, Exponential and Impulse, first started from the uniform noise and then use some functions to convert the uniform noise to Gaussian, Rayleigh, Gamma and Exponential; Impulse noise is generated in a different way, consulting the textbook and some other references) and then add these noises to the test patter image Fig0503(original_pattern).tif to compare the visual results of the noisy images.
Add some of these noises to the circuit image Circuit.tif (images can be found at ftp://ftp.cs.sjtu.edu.cn:990/lu-ht/DIP/images) and investigate the noise reduction results using different mean filters and order statistics filters as the textbook did at pages 344-352 (Pages 322-329 in the electronic version of the textbook).

4.2 This is a section

4.2.1 This is a subsection

4.2.2 This is a subsection

5.1 OverView

5.2 This is a section

5.2.1 This is a subsection

5.2.2 This is a subsection

A.1 Problem1

```
% Problem 1
% by Xue Fanyong
% Student ID:515030910443
% Histogram Equalizatio

%% Main Part
image1 = imread('Image Path/Fig1.jpg');
image2 = imread('Image Path/Fig2.jpg');

[histogram1,histogram_e1,transfer_f1,image_e1] =
histogram_equalization(image1);
[histogram2,histogram_e2,transfer_f2,image_e2] =
histogram_equalization(image2);

plot_data(image1,image_e1,histogram1,histogram_e1,transfer_f1);
plot_data(image2,image_e2,histogram2,histogram_e2,transfer_f2);

%% Functions Part

% get histogram of image
% image: get histogram of it
% histogram: the histogram of image
function histogram = get_histogram(image)
    histogram = zeros(256,1);
    [row,col]=size(image);
    for r = 1:row
        for c = 1:col
            gray = image(r,c);
            histogram(gray+1)=histogram(gray+1)+1;
        end
    end
end

% do the histogram_equalization for image
% image: do histogram_equalization for it
% histogram: original histogram; histogram_e:
% histogram after histogram
% equalizatio; transfer_f: transfer function;
% image_e: image after histogram
% equalizatio
function [histogram,histogram_e,transfer_f,image_e] =
histogram_equalization(image)
    [row,col]=size(image);
    transfer_f = zeros(256,1);
    histogram = get_histogram(image);
    transfer_f(1) = 256*histogram(1)/(row*col);

    for i = 2:256
        transfer_f(i) = transfer_f(i-1)+255*histogram(i)/(row*col);
```

```
    end
    transfer_f = round(transfer_f);

    image_e = image;
    for r = 1:row
        for c = 1:col
            image_e(r,c)=transfer_f(image(r,c)+1);
        end
    end
    histogram_e = get_histogram(image_e);
end

% plot data
% image:original image; image_e:
% image after histogram equalizatio;
% histogram: original histogram;
% histogram_e: histogram after histogram equalizatio;
% transfer_f: transfer function
function plot_data(image,image_e,histogram,histogram_e,transfer_f)
    figure();
    subplot(2,3,1);
    imshow(image);
    title(" Original Image");
    subplot(2,3,2);
    imshow(image_e);
    title(" Image(Histogram Equalization)");
    subplot(2,3,3);
    bar(histogram);
    title(" Histogram");
    subplot(2,3,4);
    bar(histogram_e);
    title(" Histogram(Equalization)");
    subplot(2,3,5);
    plot(transfer_f);
    title(" Transfer Funciton");
end
```

A.2 Problem 2

```
% Problem 2
% by Xue Fanyong
% Student ID:515030910443
% Combining spatial enhancement methods

%% Main Part
image = imread(' Image Path/skeleton_orig.tif');
[row,col] = size(image);
mask = [-1 -1 -1;-1 8 -1;-1 -1 -1];
mask = double(mask);
```

```
b_image = laplace_transformations(image,mask);
c_image = b_image+im2double(image);
d_image = sobel_gradient(image);
e_image = smooth(d_image);
f_image = im2double(e_image).*c_image;
g_image = abs(f_image)+im2double(image);
h_image = sqrt(g_image);
plot_data(image,b_image,c_image,d_image,
          e_image,f_image,g_image,h_image);

%% Function Part

% Laplace Transfromation for image using mask
% Input:
%   image:image you want to perform
%   mask:Laplace mask you want to use
% Output:
%   image_l:image after laplace transformation

function image_l = laplace_transformations(image,mask)
    [row,col] = size(image);
    mask = double(mask);
    %append image
    image_l = im2double(image);
    image = [zeros(row,2) image zeros(row,2)];
    image = [zeros(2,col+4);image;zeros(2,col+4)];
    image_append = im2double(image);

    for r = 1:row
        for c = 1:col
            image_l(r,c) = sum(sum(image_append(r:r+2,c:c+2).*mask));
        end
    end
end
%{
    sobel gradient for image
%}
function image_s = sobel_gradient(image)
    [row,col] = size(image);
    x_mask = [-1 -2 -1;0 0 0;1 2 1];
    y_mask = [-1 0 1;-2 0 2;-1 0 1];
    image_s = image;
    image = double(image);

    for r = 2:row-1
        for c = 2:col-1
            image_s(r,c) =
            abs(sum(sum(image(r-1:r+1,c-1:c+1).*x_mask)))+
            abs(sum(sum(image(r-1:r+1,c-1:c+1).*y_mask)));
        end
    end
```

```
end
%{
    smooth image using 5*5 mean filter
%}
function image_s = smooth(image)
    [row,col] = size(image);
    image_s = image;
    for r = 3:row-2
        for c = 3:col-2
            image_s(r,c) = mean(mean(image(r-2:r+2,c-2:c+2)));
        end
    end
end
%{
    plot data
%}
function plot_data(a,b,c,d,e,f,g,h)
    figure();

    subplot(241);
    imshow(a);
    title(' (a) Oringinal Image' );

    subplot(242);
    imshow(b,[]);
    title(' (b) Laplacian of (a)' );

    subplot(245);
    imshow(c,[]);
    title(' (c) Sharpened image' );

    subplot(246);
    imshow(d);
    title(' (d) Sobel gradient' );

    subplot(243);
    imshow(e);
    title(' (e) Smoothed sobel image' );

    subplot(244);
    imshow(f,[]);
    title(' (f) Product of (c) and (e)' );

    subplot(247);
    imshow(g,[]);
    title(' (g) Sharpened image' );

    subplot(248);
    imshow(h,[]);
    title(' (h) Final result' );
end
```