

File: hist_equal.m

```
%
% Name: hist_equal
% Type: m file (function)
% Description: An implementation of histogram equalization
% Reference: M Sonka, V Hlavac, R Boyle,
%             Image processing, analysis and machine vision,
%             Thomson, 3rd Edition, 2008
%

function [im_out, H, Hc, T] = hist_equal(im)

%input:
%im [m x n] input image
%output:
%im_out [m x n] equalized image
%H [1x256] histogram of the input image
%Hc [1x256] cumulative histogram of the input function
%T [1x256] transformation function of the intensity

% assuming a gray level of 255
levels = 256;

% note that matlab starts indexes at 1
% this aids coding and actually saves some computing time
% hard coding uint8 is not a good idea
% the pixel size should first be determined
imp = uint8(im)+1;

% allocate memory
H = zeros(1, levels);
% scan all pixels
for i=1:size(im,1)
    for j=1:size(im,2)
        %pixel intensity indexes the accumulator
        H(imp(i,j))=H(imp(i,j))+1;
    end
end

% form the cumulative image histogram Hc
Hc = zeros(size(H));
% this should be done although seldom do we have pure black
% in photo in practice
Hc(1)=H(1);
for i=2:size(Hc,2)
    Hc(i)=Hc(i-1)+H(i);
end

% create the look-up table normalizing
% the cumulative histogram to have integer
% values between 0-(levels-1)
T = round((levels-1)/(size(im,1)*size(im,2))*Hc);

% apply the look-up table to each level in
% the input image and write a new image
im_out = zeros(size(im));
im_out = T(imp);

% converts the pixels of the output image into
% unsigned 8-bit integers
im_out = uint8(im_out);
```

File: image_hist_equal.m

```
%
% Name: image_hist_equal
% Type: m file (main program)
% Description: Demonstration of histogram equalization
%

% clear all removes all variables etc in memory
clear all;

% generate a menu of choices for user input
method = menu ('Choose an image', ...
    'Rising Moon', ...
    'Singapore Water Front - around 1993', ...
    'Water Front - Different Shades', ...
    'Random Image');

switch method
    case 1
        I = imread('images\rising_moon_sonka.jpg');
        im = rgb2gray(I);

    case 2
        I = imread('images\city_skyline.jpg');
        im = rgb2gray(I);

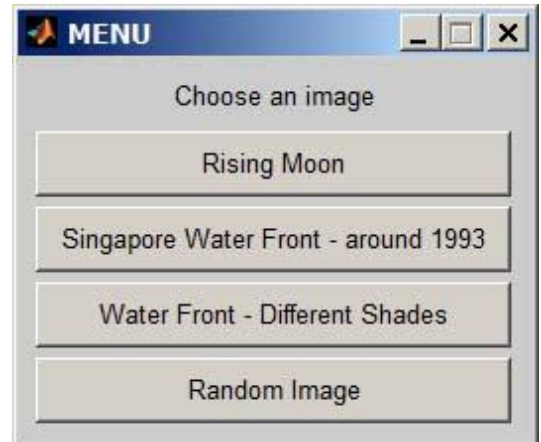
    case 3
        I = imread('images\city_skyline_shaded_2.jpg');
        im = rgb2gray(I);

    case 4
        % create an image with pixel intensity determined
        % by random number generator
        im = uint8( round(255*rand(256, 256)));
end

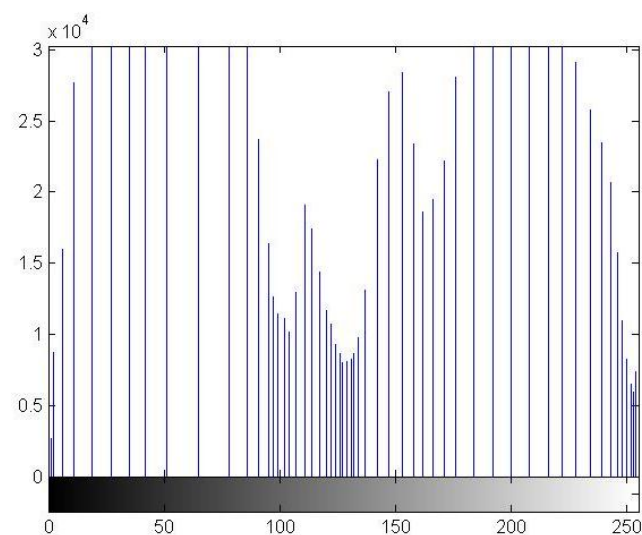
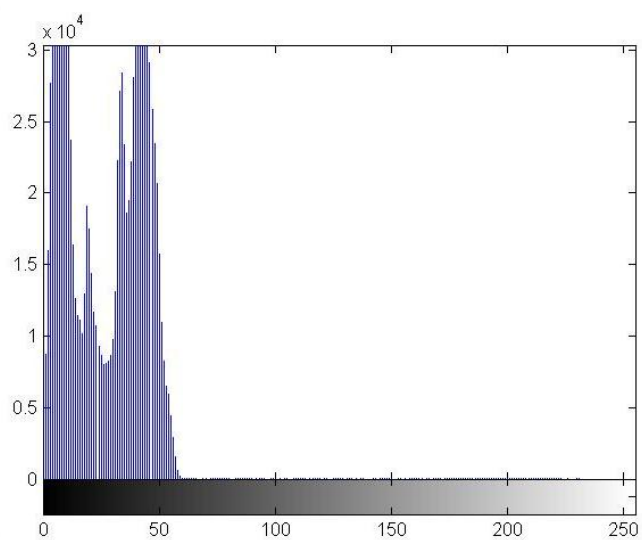
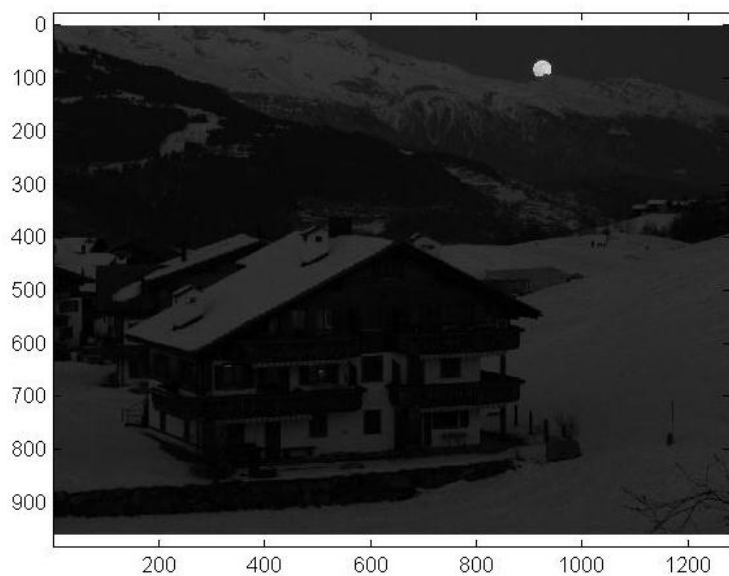
% display the original images and histogram
figure (1); colormap(gray(256)); image(im); axis equal;
figure (2); imhist(im);

% perform histogram equalization
im_out = hist_equal(im);

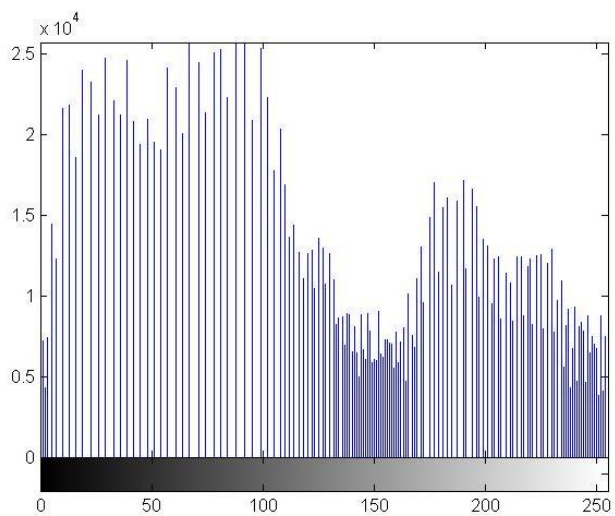
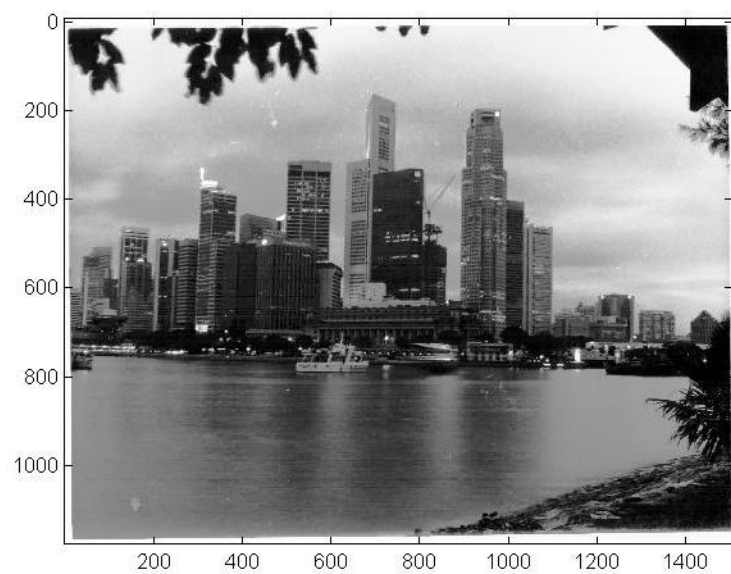
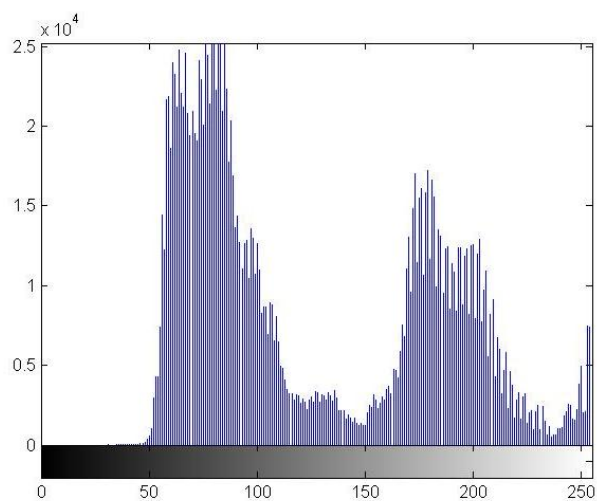
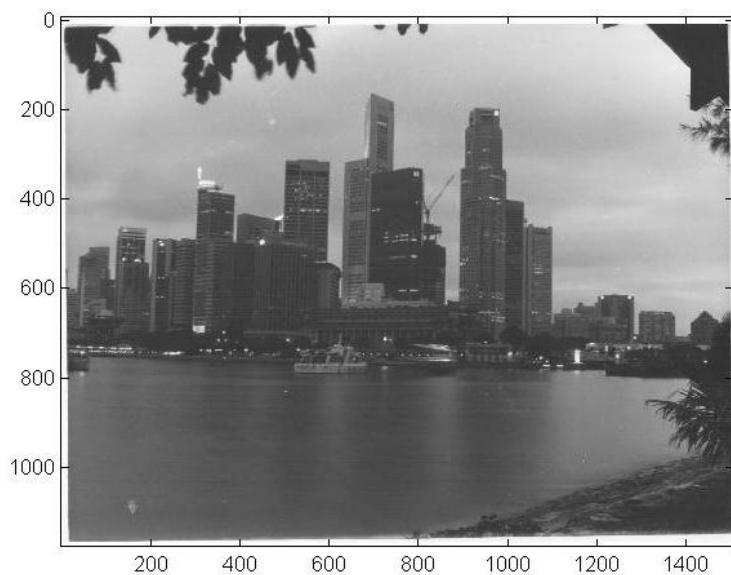
% display the output images and histogram
figure (3); colormap(gray(256)); image(im_out); axis equal;
figure (4); imhist(im_out);
```



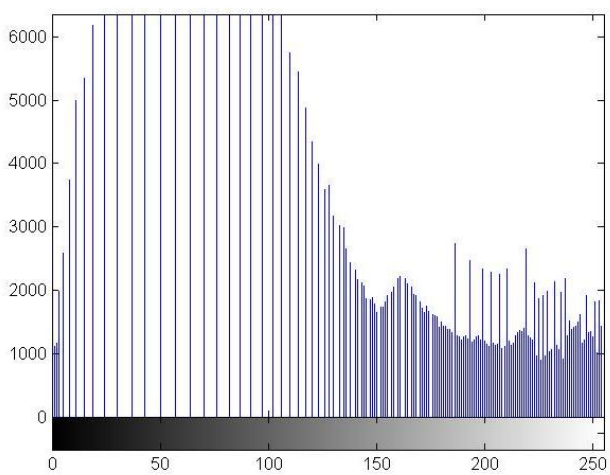
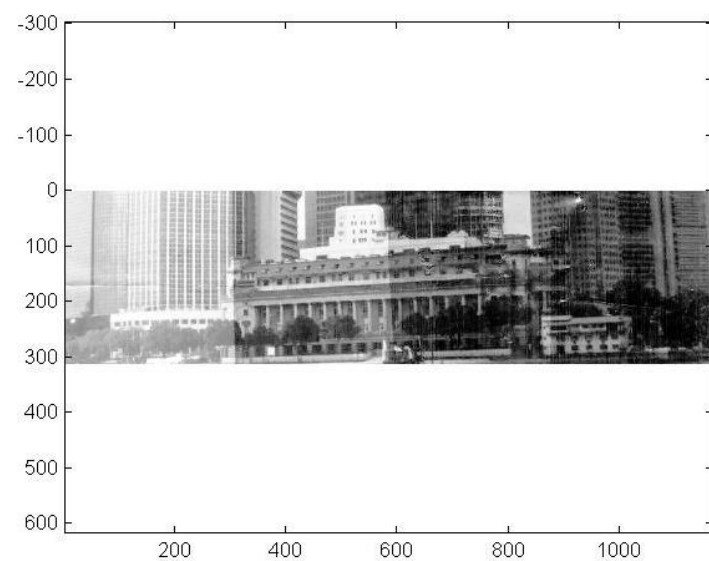
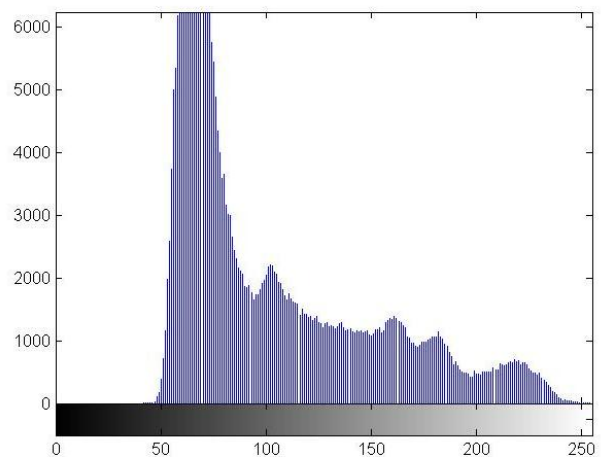
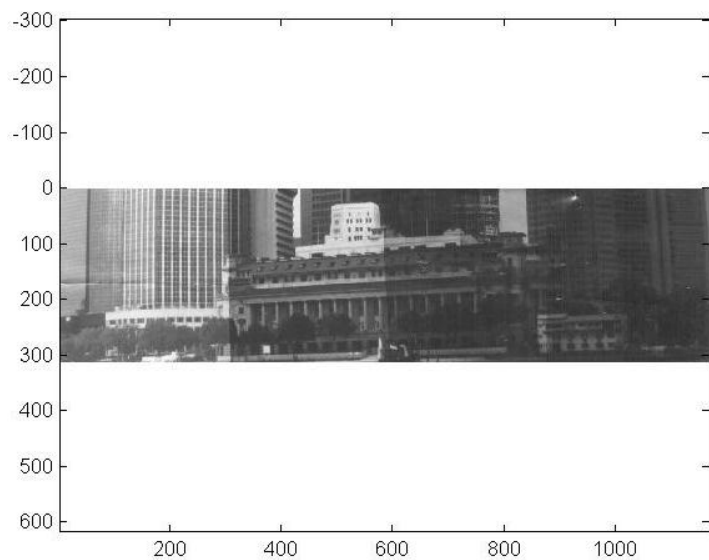
Rising Moon (Image from Sonka etal, 2008)



Singapore Water Front – Around 1993



Water Front – Different Shades



Random Image

