Homework 2

ECE 253

Mingxuan Wang A53077257

October 24, 2015

- **Problem 1. Histogram Equalization**

Figure 1 Original Image of fog.jpg

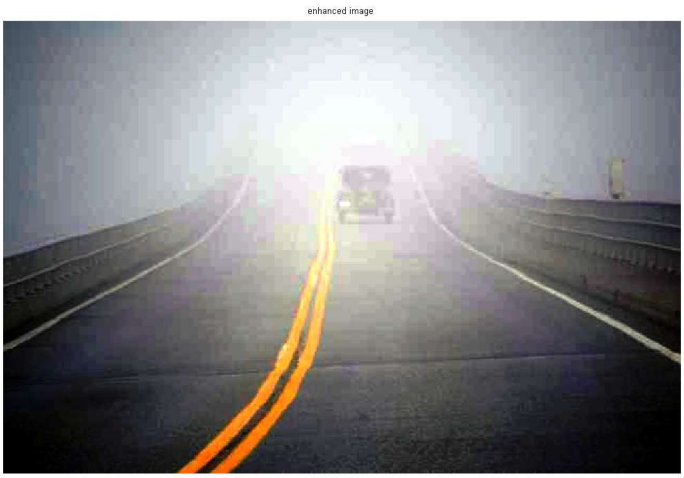

Figure 2 Image of fog.jpg after equalization

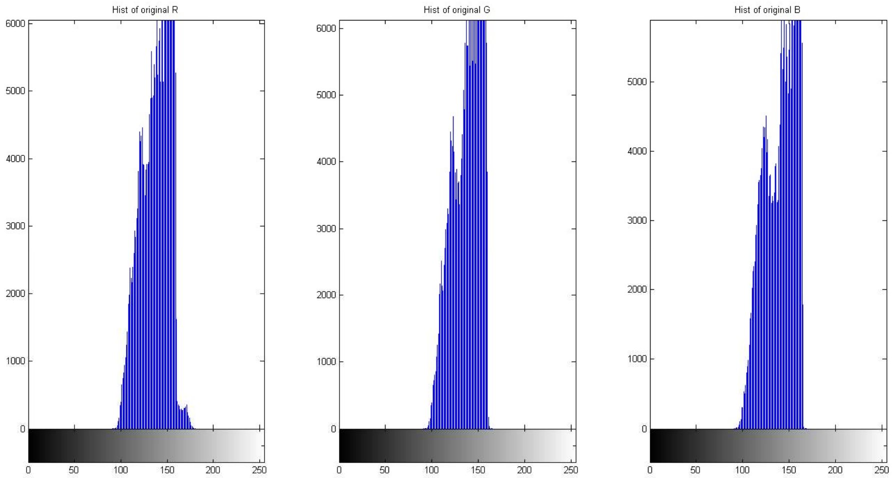

Figure 3 Histogram of Original Image

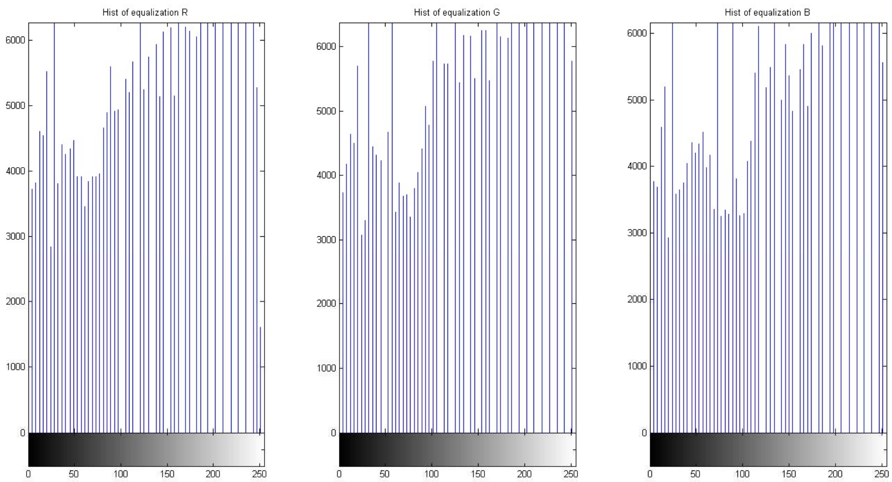

Figure 4 Histogram after equalization

Answers:

1. Statistically, the original image has pixel value in a certain range, about 90 to 170 in gray. This means that each gray level occurs with different probability in the original image.
2. In a visual sense, the original image doesn’t has that much contrast. And after the equalization, the contrast is more obvious.

- **Problem 2. Sampling & Quantization**

Figure 5 Original Image of pepper.png

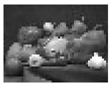

Figure 6 Image after sampling

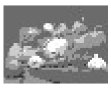

Figure 7 image after both sampling and quantization

Answer:
The sampling operation make the image much smaller than the original one because only a few pixels remaining after sampling. This is how it compress image.

Problem 3. Lloyd-Max Quantize

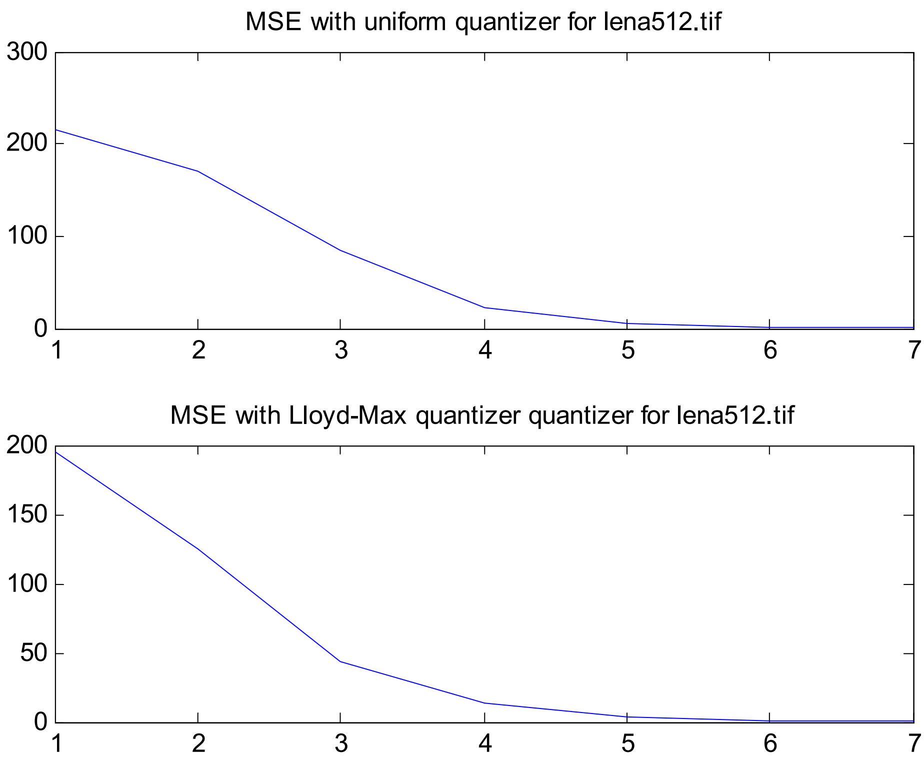

Figure 8 MSE with lena512.tif before equalization

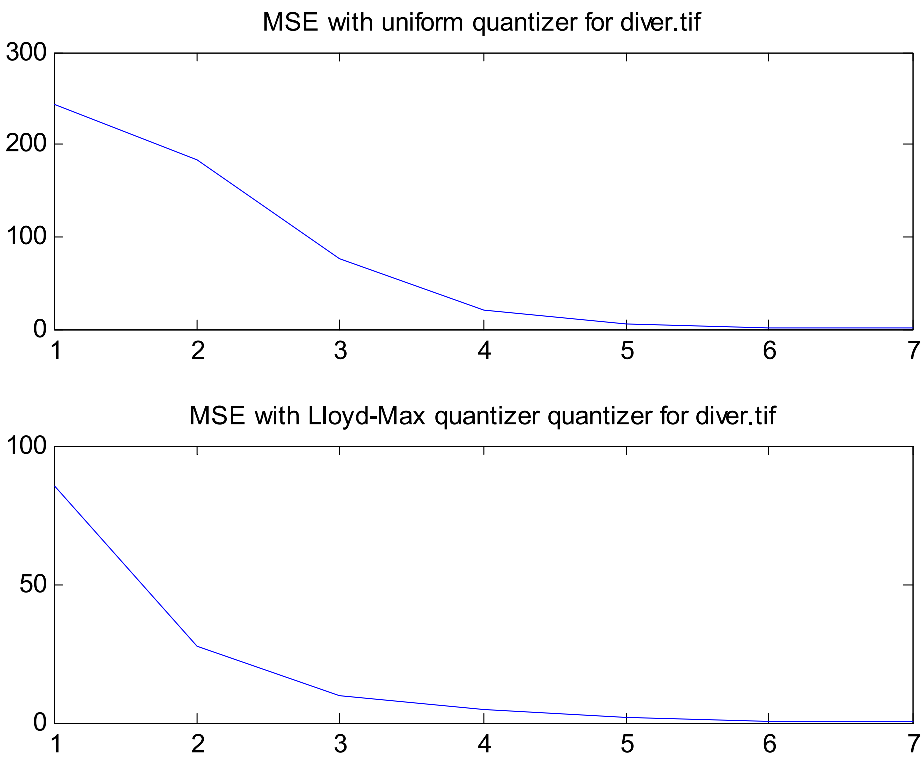

Figure 9 MSE with diver.tif before equalization

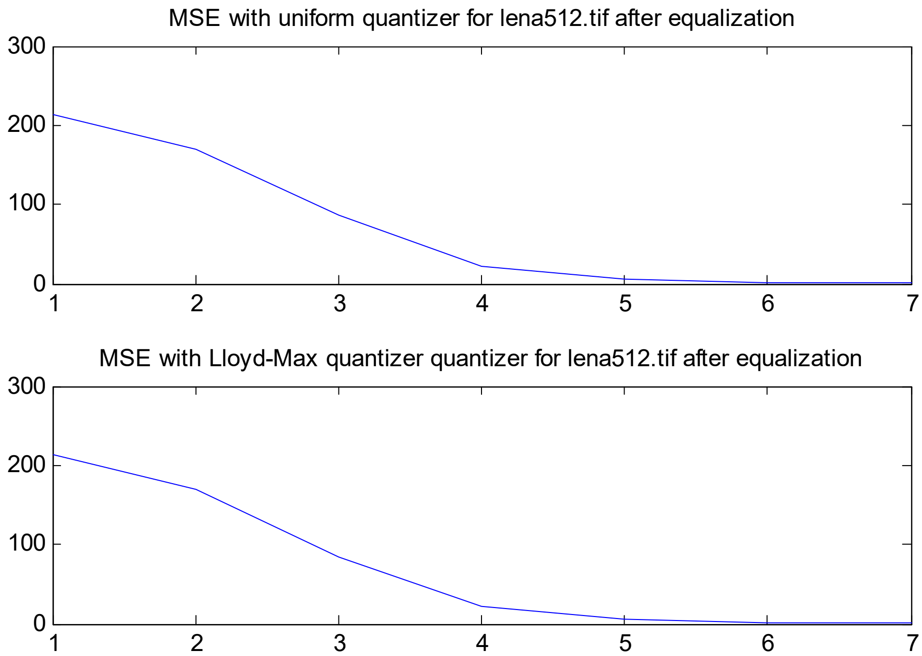

Figure 10 MSE with lena512. tif after equalization

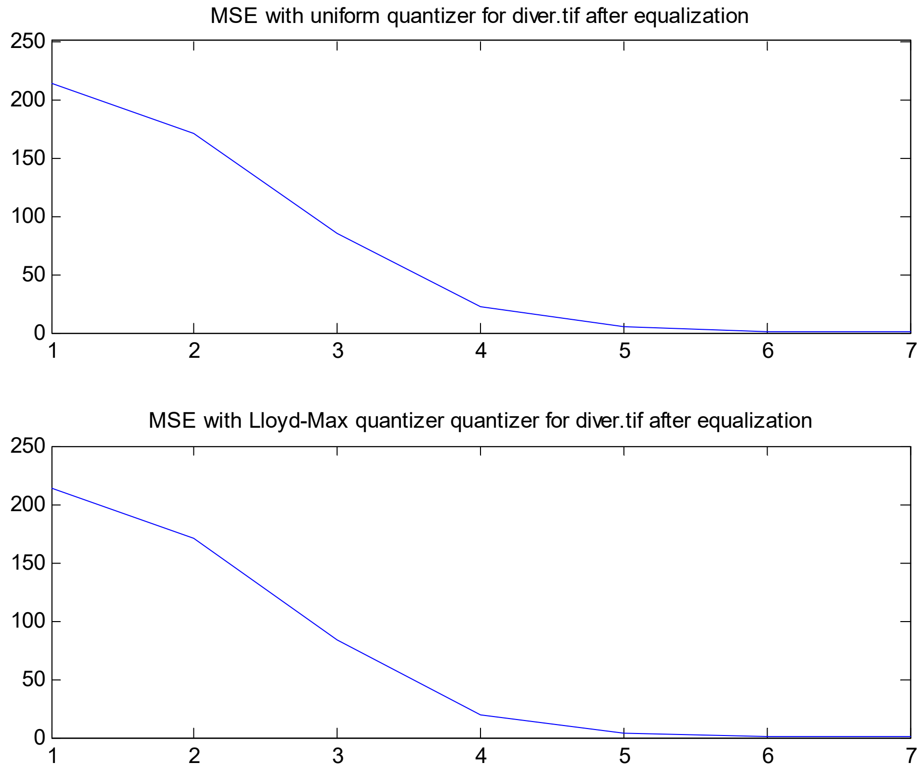

Figure 11 MSE with diver.tif after equalization

Answer:
ii. For the uniform quantizer, we treat each level the same no matter what image it is. But for Lloyds-max quantizer, the partition and codebook depends on the given image (minimal the square error). So generally speaking, Lloyds-max will perform better than uniform one.
For different images, lena512 has higher contrast than diver. Diver has pixel values more like average/median value in a level while lena512 is not. So for diver.tif, the mse is smaller than lena512.tif.

iii. After equalization, the two images has higher contrast than original ones. For lena512, it won't do much work after equalization so we see the mse is similar to the original one. But for diver, with higher contrast, the value of each pixel becomes more far away from the average/median value in each level, so the mse becomes larger than the original one.
Another thing is that after equalization, the two quantizer perform nearly the same.

iv. For 7 bit, there are 128 levels in an image. In this case, for each pixel, it only suffer influence of its one neighbor. Plus after equalization, the distribution is more uniform-like and for each pixel, its neighbor is more similar to itself. I notice that the partition of Lloyds-max for 7-bits is nearly one or two value gap from 20+ to 240+, so it is similar to the original image. For these reasons, the mse is near zero.
For the thought that equalization is not to the advantage of Lloyds-max quantizer, I think it depends. For different images, it may have different performance. Even if equalization does make the distribution uniform like, it may not lead to the result that mse is smaller because the differences between the median value of each average and actual value may be bigger.

- **Appendix**

Hw2.m:

```
clear all

clc

%% Problem 1

filename = 'D:\ucsd\ece253\hw2\fog.jpg';

eq_img = equalization(filename);


%% Problem 2

filename2 = 'D:\ucsd\ece253\hw2\peppers.png';

samp_quan_img = samp_quan(filename2);


%% Problem 3

% i & ii

% lena

lena = imread('D:\ucsd\ece253\hw2\lena512.tif');

[mse_lena_uni, mse_lena_lloyds] = mse(lena);

x = 1:7;

figure(1);

subplot(2,1,1);

plot(x, mse_lena_uni);
```

```
title('MSE with uniform quantizer for lena512.tif');

subplot(2,1,2);

plot(x, mse_lena_lloyds);

title('MSE with Lloyd-Max quantizer quantizer for lena512.tif');

%diver

diver = imread('D:\ucsd\ece253\hw2\diver.tif');

[mse_diver_uni, mse_diver_lloyds] = mse(diver);

x = 1:7;

figure(2);

subplot(2,1,1);

plot(x, mse_diver_uni);

title('MSE with uniform quantizer for diver.tif');

subplot(2,1,2);

plot(x, mse_diver_lloyds);

title('MSE with Lloyd-Max quantizer quantizer for diver.tif');


% iii

lena_equal = histeq(lena, 256);

diver_equal = histeq(diver, 256);

figure; imshow(lena_equal);

figure; imshow(diver_equal);

[mse_lena_equni, mse_lena_eqll] = mse(lena_equal);
```

```
[mse_diver_equni, mse_diver_eqll] = mse(diver_equal);

figure(3);

subplot(2,1,1);

plot(x, mse_lena_equni);

title('MSE with uniform quantizer for lena512.tif after equalization');

subplot(2,1,2);

plot(x, mse_lena_eqll);

title('MSE with Lloyd-Max quantizer quantizer for lena512.tif after equalization');


figure(4);

subplot(2,1,1);

plot(x, mse_diver_equni);

title('MSE with uniform quantizer for diver.tif after equalization');

subplot(2,1,2);

plot(x, mse_diver_eqll);

title('MSE with Lloyd-Max quantizer quantizer for diver.tif after equalization');
```

Equalization.m:

```
function img = equalization( filename )

    img=imread(filename);

    imgR=img(:,:,1);
```

```
imgG=img(:,:,2);

imgB=img(:,:,3);

subplot(131);imhist(imgR);

title('Hist of original R');

subplot(132);imhist(imgG);

title('Hist of original G');

subplot(133);imhist(imgB);

title('Hist of original B');

hnewR=histeq(imgR);

figure;

subplot(131);imhist(hnewR);

title('Hist of equalization R');

hnewG=histeq(imgG);

subplot(132);imhist(hnewG);

title('Hist of equalization G');

hnewB=histeq(imgB);

subplot(133);imhist(hnewB);

title('Hist of equalization B');

figure;

enhanced_img=cat(3,hnewR,hnewG,hnewB);

imshow(enhanced_img);

title('enhanced image')
```

```
end
```

Samp_quan.m:

```
function img = samp_quan( filename )
img_ori = imread(filename);
figure(1);imshow(img_ori);
f = 10;
[row, col, ~] = size(img_ori);
%sample
img_new = img_ori(1:f:row, 1:f:col, :);
figure(2);imshow(img_new);

[row_sam, col_sam] = size(img_new);
img = img_new;
for i = 1: row_sam
    for j = 1:col_sam
        quan = floor(img_new(i,j)/52);
        switch quan
            case 0
                img(i, j) = 51;
            case 1
                img(i, j) = 107;
```

```
                case 2
                    img(i, j) = 153;
                case 3
                    img(i,j) = 204;
                case 4
                    img(i,j) = 255;
            end
        end
    end
    figure(3);imshow(img, [0, 255]);
end
```

---

quan.m:

```
function  final_image = quan(ori_image, s)
    ori_image = double(ori_image);
    [row, col] = size(ori_image);
    final_image = ori_image;
    level = 2^s;
    step = 256/level;
    period = zeros(level, 2);
    key = zeros(level, 1);
    for i = 1:level
```

```
        period(i, 1) = (i-1)*step;
        period(i, 2) = (i-1)*step+step-1;
        key(i) = floor(period(i, 1) + (period(i, 2)-period(i ,1))/2);
    end

    for i = 1:row
        for j = 1:col
            target_period = floor(ori_image(i, j)/step)+1;
            final_image(i, j) = key(target_period);
        end
    end

    final_image = uint8(final_image);

end
```

Lloyds_quan.m:

```
function final_image = lloyds_quan( partition, codebook, ori_image )
    [row, ~] = size(ori_image);
    final_image = ori_image;
    for i   = 1: row
        [~, quants] = quantiz(ori_image(i, :), partition, codebook);
```

```
        final_image(i, :) = quants;
    end
end
```

---

mse.m:

```
function [mse_uni, mse_lloyds] = mse( lena )
    mse_uni = zeros(1,7);
    mse_lloyds = zeros(1,7);
    for s = 1:7
        lena_uniquan = quan(lena, s);
        training_set = double(lena(:));
        [partition, codebook] = lloyds(training_set, 2^s);
        lena_lloyds = lloyds_quan(partition ,codebook, lena);
        D = abs(lena_uniquan-lena).^2;
        mse_uni(s) = sum(D(:))/numel(lena);
        D = abs(lena_lloyds-lena).^2;
        mse_lloyds(s) = sum(D(:))/numel(lena);
    end
end
```