Bolin He, PID: A53316428, Hw02

Table of Contents

Nov 06,2019

Problem 1. Adaptive Histogram Equalization

```
clear all;
close all;
clc;

B = imread('beach.png');
%original
figure(1)
imshow(B)
title('original')

%HE
figure(2)
subplot(2,2,1)
B_HE=histeq(B);
imshow(B_HE)
title('simple HE')

%win_size=33
subplot(2,2,2)
AHE(B,33)
title('AHE & winsize=33')

%win_size=65
subplot(2,2,3)
AHE(B,65)
title('AHE & winsize=65')

%win_size=129
subplot(2,2,4)
AHE(B,129)
title('AHE & winsize=129')

% function [ ] = AHE(im,win_size)
% impad = padarray(im,[(win_size-1)/2,(win_size-1)/2],'symmetric');
% [x,y] = size(im);
% output = zeros(x,y);
%
% for i = 1:x
```

```
%     for j =1:y
%         rank = 0;
% %         contextual_region = im(i:i+win_size-1,j:j+win_size-1);
%         for k = i:i+win_size-1
%             for t = j:j+win_size-1
%                 if impad((win_size-1)/2+i,(win_size-1)/2+j) >
 impad(k,t)
%                     rank = rank+1;
%                 end
%             end
%         end
%         output(i,j) = rank*255/(win_size*win_size);
%     end
% end
%
% enhanced_image = imshow(output,[])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (1) Both HE and AHE can enhance the contrast, while AHE can get
%     more details than HE.
% (2) I think AHE works better than HE for the beach image.
%     However, choosing AHE or HE depends on the image.
```

simple HE

AHE & winsize=33

AHE & winsize=65

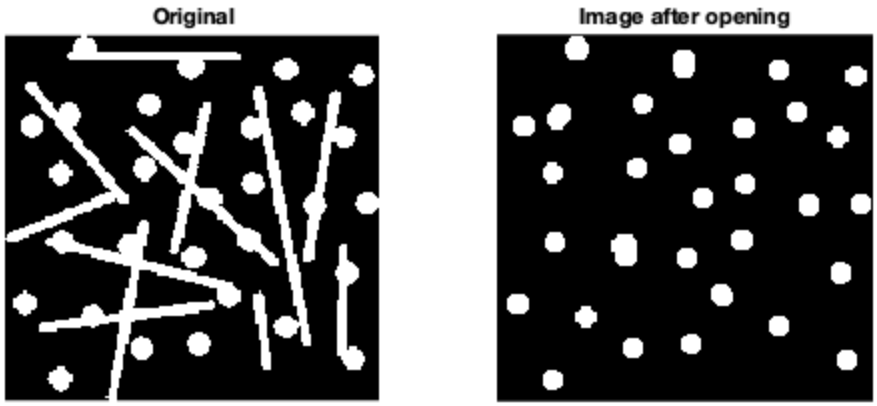
AHE & winsize=129

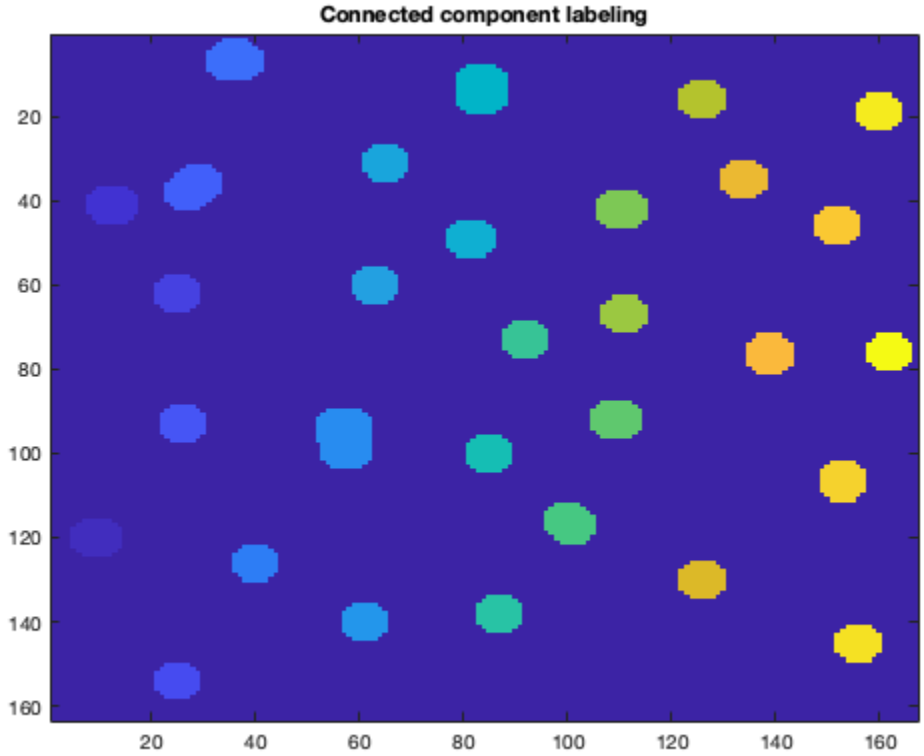
Problem 2. Binary Morphology

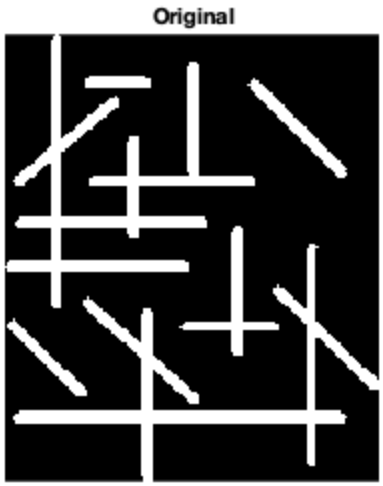
```
close all;
clear all;
clc;

CL = imread('circles_lines.jpg');
CL = im2bw(CL);
L = imread('lines.jpg');
L = im2bw(L);

%%%%% Part(i) %%%%%
% Process Image circles_lines .jpg
se = strel('disk',5);
OpenCL = imopen(CL,se);
[x,y] = bwlabel(OpenCL);

for i = 1:y
    [row,col] = find(i == x);
    centroid_x1(i) = mean(row);
    centroid_y1(i) = mean(col);
    area(i) = length(row);
end

% plot figures
```

```
figure(1)
subplot(1,2,1)
imshow(CL)
title('Original')

subplot(1,2,2)
imshow(OpenCL)
title('Image after opening')

figure(2)
imagesc(x)
title('Connected component labeling')

% table
Centroid = [centroid_x1',centroid_y1'];
Area = area';
table(Centroid,Area)

%%%%% Part(ii) %%%%%
% Process Image lines.jpg
se2 = strel('line',8,90);
OpenL = imopen(L,se2);
[x2,y2] = bwlabel(OpenL);

for i = 1:y2
    [row2,col2] = find(i == x2);
    centroid_x2(i) = mean(row2);
    centroid_y2(i) = mean(col2);
    length(i) = max(row2)-min(row2);
end

% plot figures
figure(3)
subplot(1,2,1)
imshow(L)
title('Original')

subplot(1,2,2)
imshow(OpenL)
title('Image after opening')

figure(4)
imagesc(x2)
title('Connected component labeling')

% table
Centroid = [centroid_x2',centroid_y2'];
Length = length';
table(Centroid,Length)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The type and size for opening are:
%     'disk',5
%     'line',8,90
```

```
ans =

  30×2 table

        Centroid          Area
    ________________    _____

       120       9.5     78
        41      12.5     78
        62        25     69
       154        25     69
        93        26     69
        37        28     95
    6.7283        36     92
       126        40     69
    96.324    57.199    136
       140        61     69
        60        63     69
        31        65     69
        49      81.5     78
      13.5      83.5    108
       100        85     69
       138        87     69
        73        92     69
     116.5     100.5     82
        92     109.5     78
        42     110.5     78
        67       111     69
        16       126     69
       130       126     69
        35       134     69
      76.5       139     78
        46       152     69
     106.5       153     78
       145       156     69
        19       160     69
        76       162     69


ans =

  6×2 table

        Centroid          Length
    ________________    _______

    57.374    21.508      112
      63.5    53.039       41
     149.1    59.017       70
    36.834    78.049       50
     106.3    96.019       52
    132.62    126.51       91
```


Original
Image after opening


Connected component labeling
20
40
60
80
100
120
140
160
20
40
60
80
100
120
140
160


Original

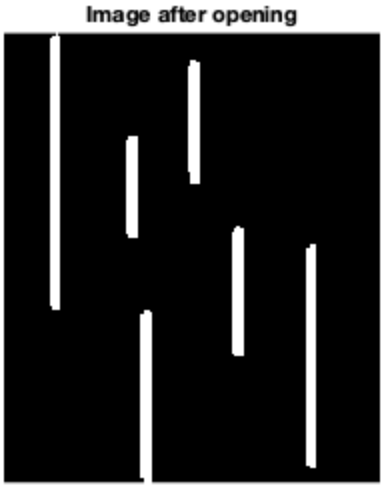
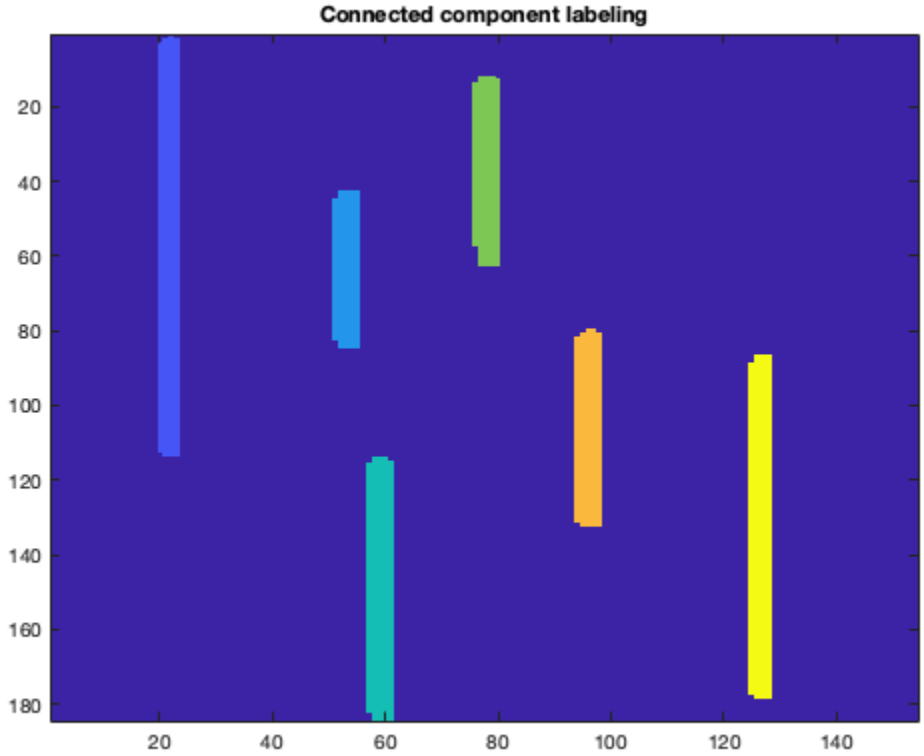


Image after opening


Connected component labeling
20
40
60
80
100
120
140
160
180
20
40
60
80
100
120
140

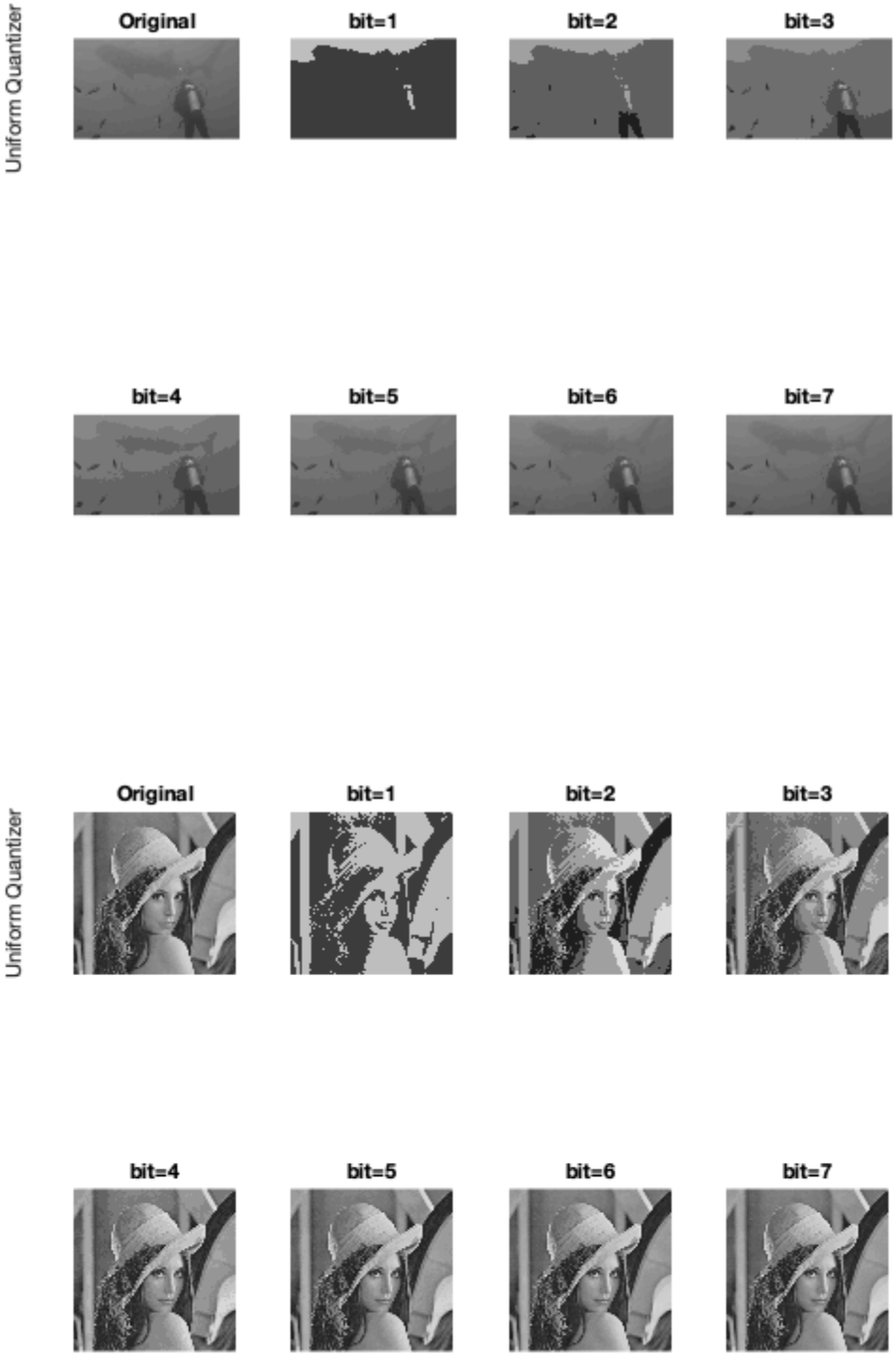
Problem 3. Lloyd-Max Quantizer

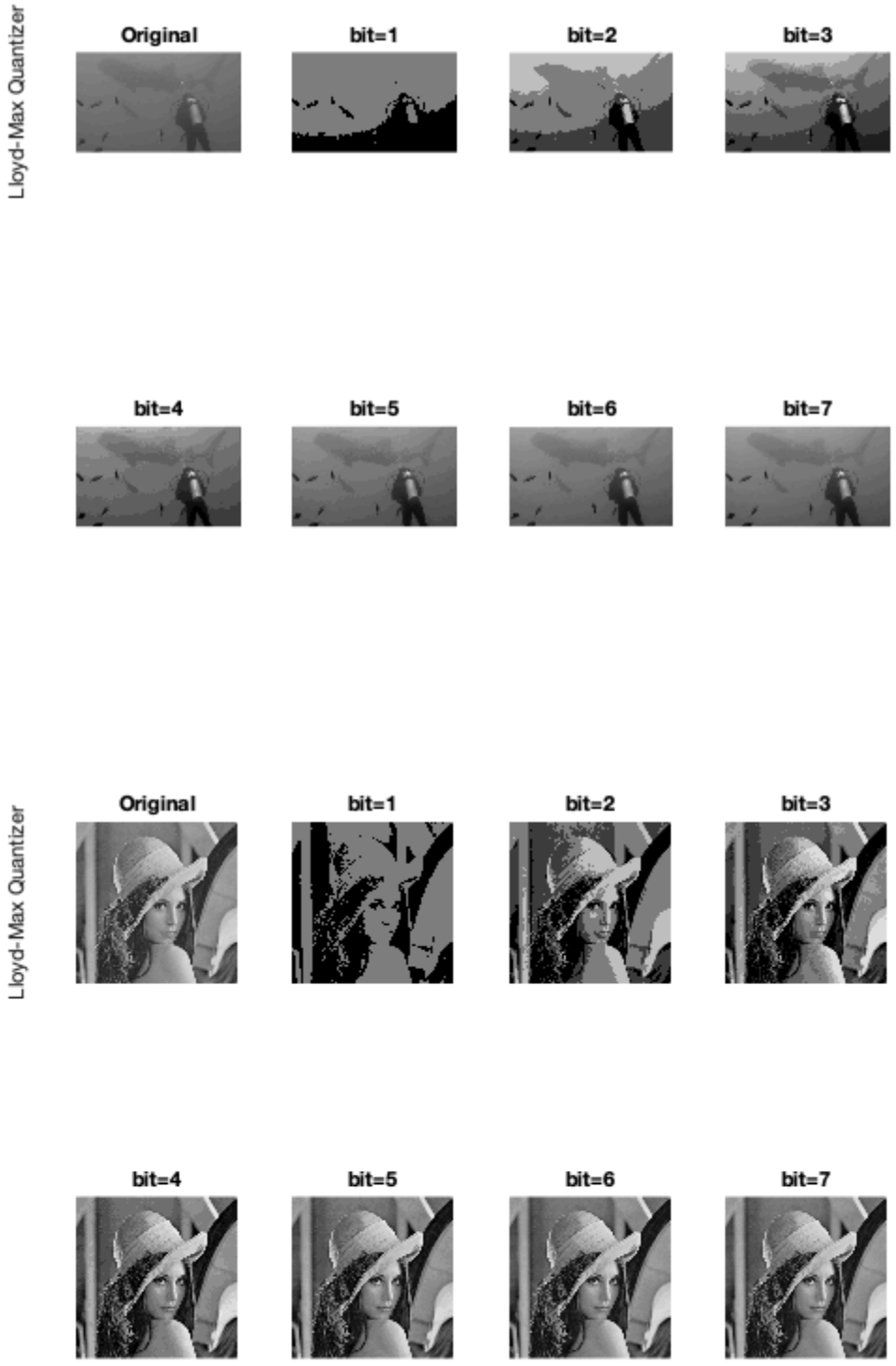
```
close all;
clear all;
clc;

D = imread('diver.tif');
L512 = imread('lena512.tif');

%%%%% Part(i) %%%%%
figure
s = 1:7;
MSE_Uni_1 = myF(D,s);
subplot(2,4,1)
imshow(D)
title('Original')
ylabel('Uniform Quantizer')


figure
MSE_Uni_2 = myF(L512,s);
subplot(2,4,1)
imshow(L512)
title('Original')
ylabel('Uniform Quantizer')
% function [ ] = myF(I,s)
%     I = double(I);
%     [x,y]= size(I);
%     Store = zeros(x,y);
%     interval = 255./2.^s;
%     for i = 1:length(s)
%         T = round(0:interval(i):255);
%         for j = 1:length(T)-1
%             Store(I>T(j) & I<=T(j+1)) = (T(j)+T(j+1))/2;
%         end
%         subplot(2,4,i+1)
%         Store = uint8(Store);
%         imshow(Store)
%         title(['bit=',num2str(i)])
%     end
% end

%%%%% Part(ii) %%%%%
%
[m1,n1] = size(D);
training_set_1 = double(reshape(D,n1*m1,1));
[m2,n2] = size(L512);
training_set_2 = double(reshape(L512,n2*m2,1));

%
figure
subplot(2,4,1)
imshow(D)
```

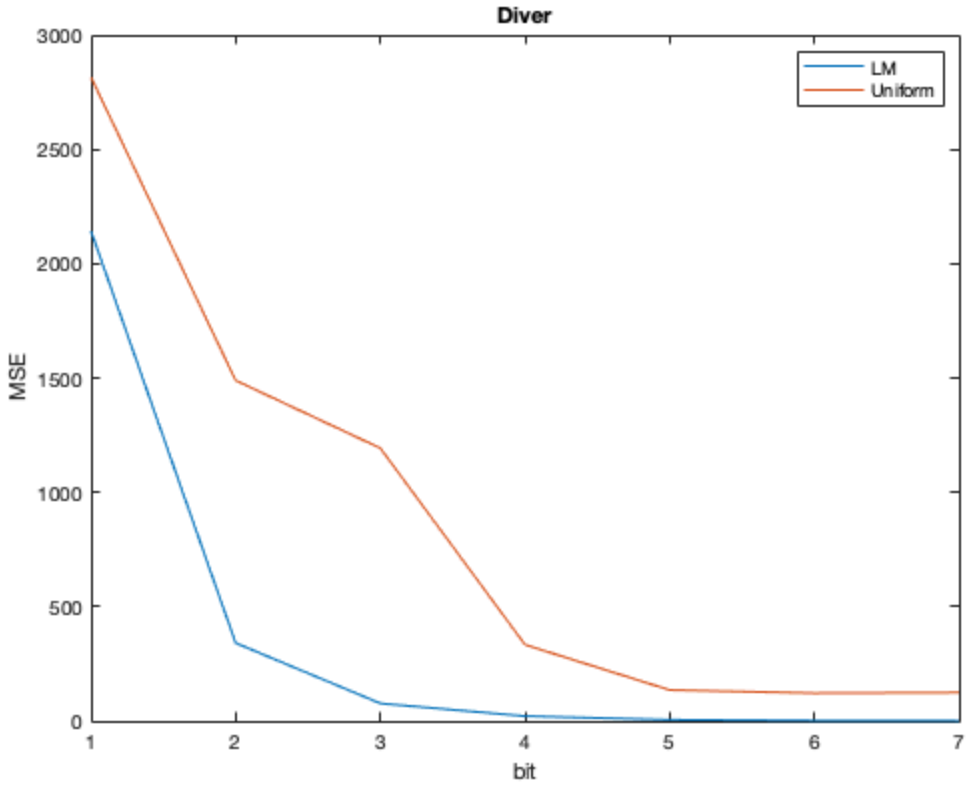
```
title('Original')
ylabel('Lloyd-Max Quantizer')
for s = 1:7
    len = 2.^s;
    [partition_1, codebook_1] = lloyds(training_set_1, len);
    [idx_1,quantv_1] = quantiz(training_set_1,partition_1,codebook_1);
    idx_1 = idx_1/len*255;
    idx_1 = uint8(reshape(idx_1,[m1,n1]));
    subplot(2,4,s+1)
    imshow(idx_1)
    title(['bit=',num2str(s)])
    MSE_1(s) = sum(sum((double(idx_1)-double(D)).^2))/numel(D);
end


%
figure
subplot(2,4,1)
imshow(L512)
title('Original')
ylabel('Lloyd-Max Quantizer')
for s = 1:7
    len = 2.^s;
    [partition_2, codebook_2] = lloyds(training_set_2, len);
    [idx_2,quantv_2] = quantiz(training_set_2,partition_2,codebook_2);
    idx_2 = idx_2/len*255;
    idx_2 = uint8(reshape(idx_2,[m2,n2]));
    subplot(2,4,s+1)
    imshow(idx_2)
    title(['bit=',num2str(s)])
    MSE_2(s) = sum(sum((double(idx_2)-double(L512)).^2))/numel(L512);
end
%
figure
plot(1:7,MSE_Uni_1)
hold on
plot(1:7,MSE_1)
legend('LM','Uniform')
title('Diver')
ylabel('MSE')
xlabel('bit')

figure
plot(1:7,MSE_Uni_2)
hold on
plot(1:7,MSE_2)
legend('LM','Uniform')
title('lena512')
ylabel('MSE')
xlabel('bit')


%%%%% Part(iii) %%%%%
%
```

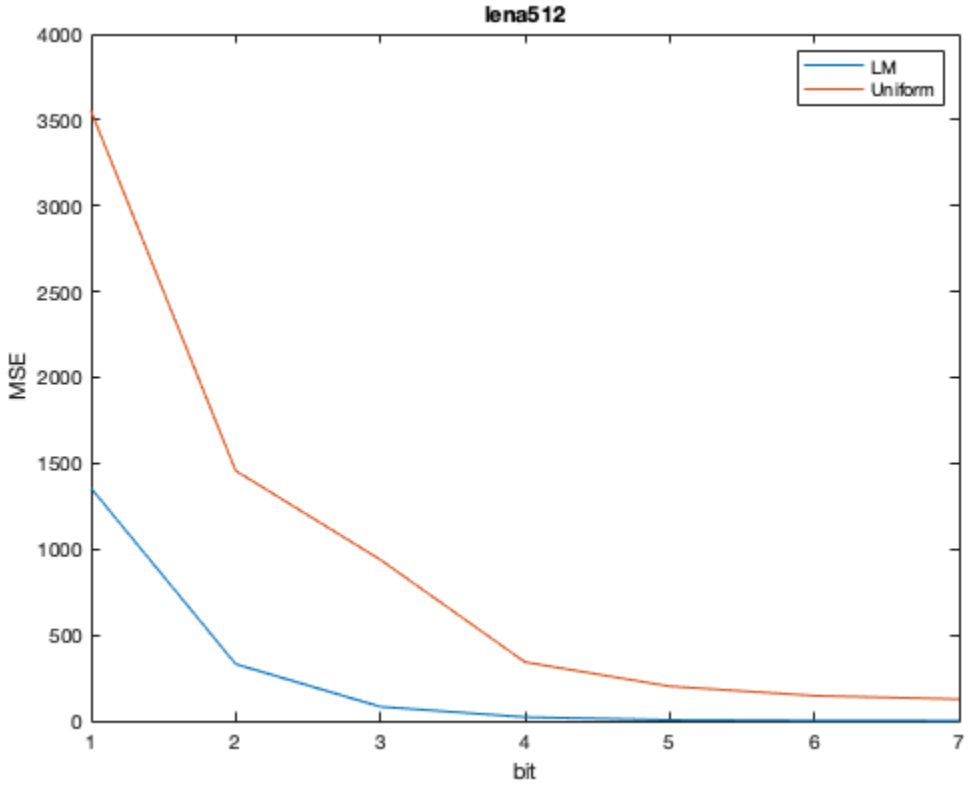
```
a = histeq(D,256);
b = histeq(L512,256);

% Uniform
s = 1:7;
a = double(a);
[x,y]= size(a);
Store1 = zeros(x,y);
interval = 255./2.^s;
for i = 1:length(s)
    T = round(0:interval(i):255);
    for j = 1:length(T)-1
        Store1(a>T(j) & a<=T(j+1)) = (T(j)+T(j+1))/2;
    end
    MSE_Uni_a(i) = sum(sum((double(Store1)-double(a)).^2))/numel(a);
end

s = 1:7;
b = double(b);
[x,y]= size(b);
Store2 = zeros(x,y);
interval = 255./2.^s;
for i = 1:length(s)
    T = round(0:interval(i):255);
    for j = 1:length(T)-1
        Store2(b>T(j) & b<=T(j+1)) = (T(j)+T(j+1))/2;
    end
    MSE_Uni_b(i) = sum(sum((double(Store2)-double(b)).^2))/numel(b);
end

a = histeq(D,256);
b = histeq(L512,256);

% LM
[m1,n1] = size(a);
training_set_1 = double(reshape(a,n1*m1,1));
[m2,n2] = size(b);
training_set_2 = double(reshape(b,n2*m2,1));

for s = 1:7
    len = 2.^s;
    [partition_1, codebook_1] = lloyds(training_set_1, len);
    [idx_1,quantv_1] = quantiz(training_set_1,partition_1,codebook_1);
    idx_1 = idx_1/len*255;
    idx_1 = reshape(idx_1,[m1,n1]);
    MSE_a(s) = sum(sum((double(idx_1)-double(a)).^2))/numel(a);
end

for s = 1:7
    len = 2.^s;
    [partition_2, codebook_2] = lloyds(training_set_2, len);
    [idx_2,quantv_2] = quantiz(training_set_2,partition_2,codebook_2);
    idx_2 = idx_2/len*255;
    idx_2 = reshape(idx_2,[m2,n2]);
```

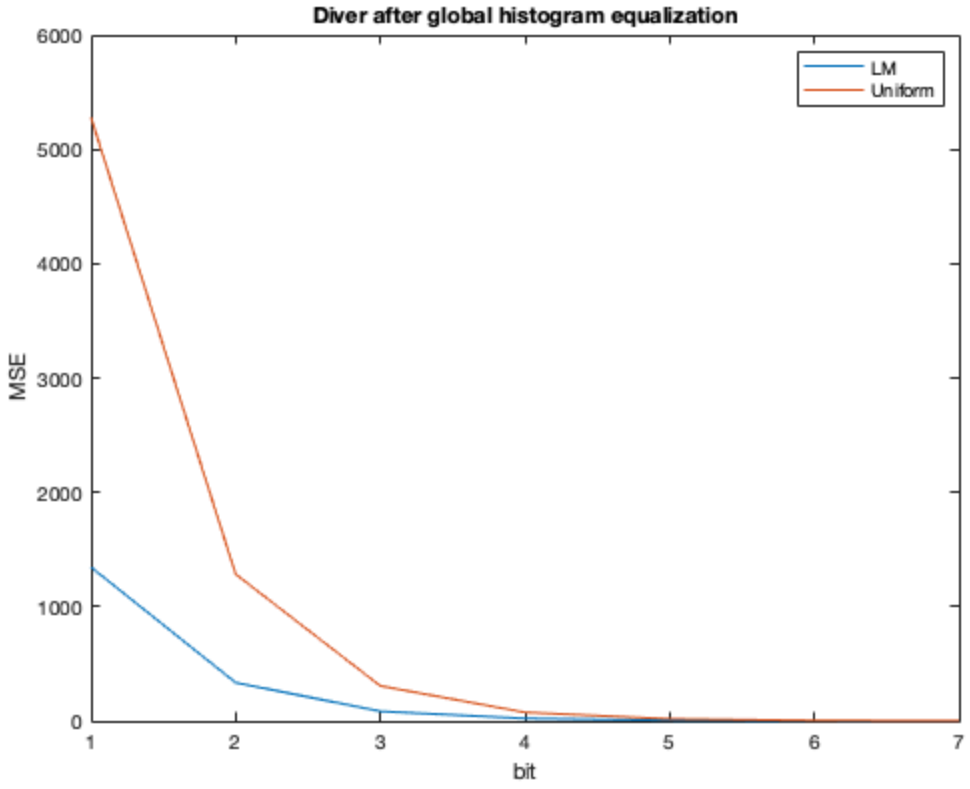
```
    MSE_b(s) = sum(sum((double(idx_2)-double(b)).^2))/numel(L512);
end

%
figure
plot(1:7,MSE_Uni_a)
hold on
plot(1:7,MSE_a)
legend('LM','Uniform')
ylabel('MSE')
xlabel('bit')
title('Diver after global histogram equalization')

%
figure
plot(1:7,MSE_Uni_b)
hold on
plot(1:7,MSE_b)
legend('LM','Uniform')
ylabel('MSE')
xlabel('bit')
title('lena512 after global histogram equalization')
%%%%% Part(iv) %%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (ii)
% Generally, LM quantizer should be better than the uniform quantizer.
% Based on the images we processed, because lena512 has a more uniform
% internsity histogram than diver, so the result is different between
 them.
%
% (iii)
% After global histogram equalization, the MSE of two images are very
% close for the same quantizers. But for the same images, the gap
 becomes
% larger. This is because HE made histogram equally distributed. The
 effect
% is different between them.
%
% (iv)
% As the bits increase, it makes sense that the MSE will decrease.
% When it comes to 7 bits, the error between each pixel is within 1,
% which is relatively small compared with the lower bits.
% Moreover, with different quantizers processing, the histogram
 distribution
% becomes more uniform, which will decrease the error among them.
```

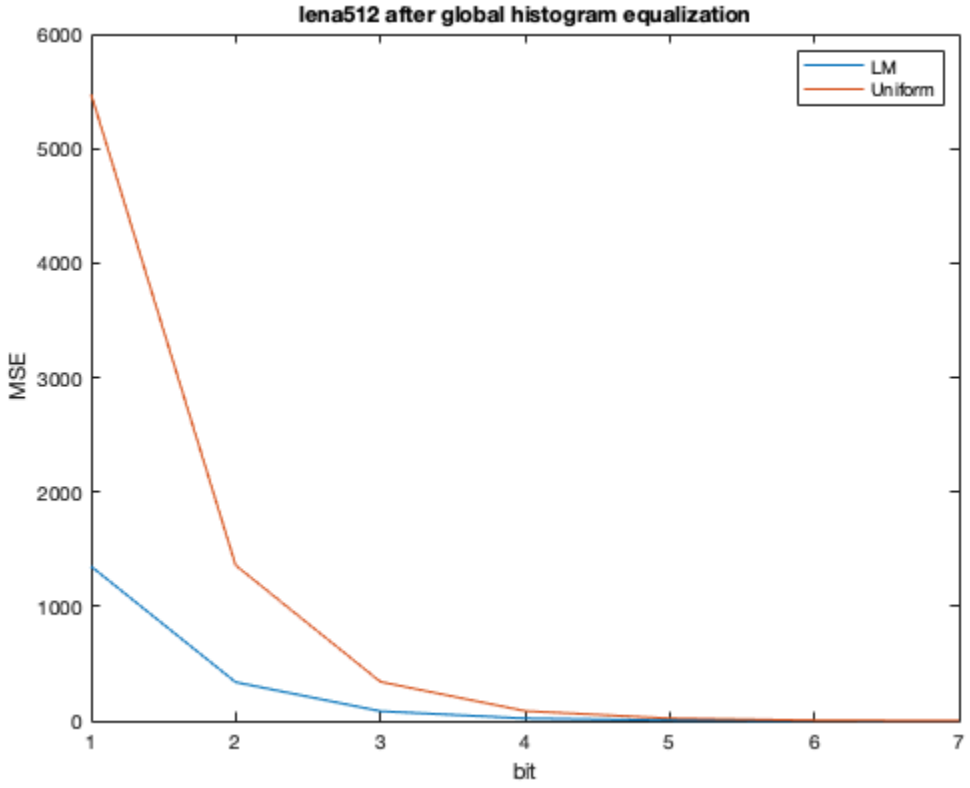

Uniform Quantizer
Original
bit=1
bit=2
bit=3
bit=4
bit=5
bit=6
bit=7
Uniform Quantizer
Original
bit=1
bit=2
bit=3
bit=4
bit=5
bit=6
bit=7


Lloyd-Max Quantizer
Original
bit=1
bit=2
bit=3
bit=4
bit=5
bit=6
bit=7
Lloyd-Max Quantizer
Original
bit=1
bit=2
bit=3
bit=4
bit=5
bit=6
bit=7


Diver
LM
Uniform
MSE
bit
0
500
1000
1500
2000
2500
3000
1
2
3
4
5
6
7


lena512
LM
Uniform
MSE
bit
0
500
1000
1500
2000
2500
3000
3500
4000
1
2
3
4
5
6
7


Diver after global histogram equalization
LM
Uniform
MSE
bit
0
1000
2000
3000
4000
5000
6000
1
2
3
4
5
6
7


lena512 after global histogram equalization
LM
Uniform
MSE
bit
0
1000
2000
3000
4000
5000
6000
1
2
3
4
5
6
7

Published with MATLAB® R2018a