ECE253-Fall2017

HW2

Jinzhao Feng

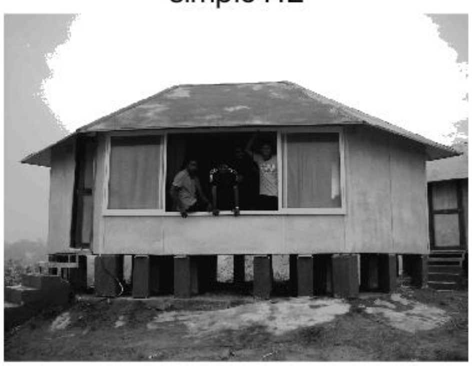
A53213089

Problem1:

```
%problem1
Im_1=imread('beach.png');
%original
figure(1)
imshow(Im_1)
title('original')
%HE
figure(2)
Im_1=histeq(Im_1);
subplot(2,2,1)
imshow(Im_1)
title('simple HE')
%win_size=33
Im_2=imread('beach.png');
Im_2=AHE_func(Im_2,33);
subplot(2,2,2)
imshow(Im_2)
title('AHE & winsize=33')
%win_size=65
Im_3=imread('beach.png');
Im_3=AHE_func(Im_3,65);
subplot(2,2,3)
imshow(Im_3)
title('AHE & winsize=65')
%win_size=129
Im_4=imread('beach.png');
Im_4=AHE_func(Im_4,129);
subplot(2,2,4)
imshow(Im_4)
title('AHE & winsize=129')
```

original

simple HE



AHE & winsize=33

AHE & winsize=65

AHE & winsize=129

(1). HE and AHE both enhance the contrast. However, AHE can get people more detailed than HE.

(2). AHE is better for this image because we need to see more detail about people. However, AHE is not necessarily better than HE. It depends on what image is and what we want.

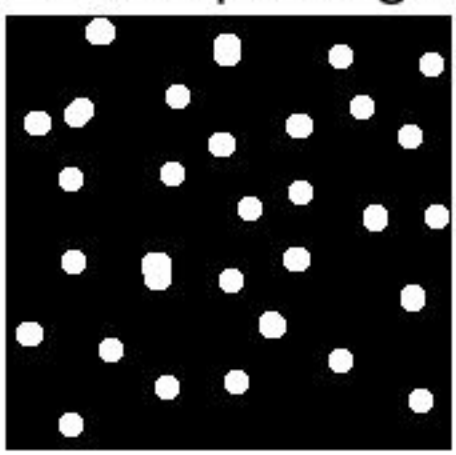
Problem2:

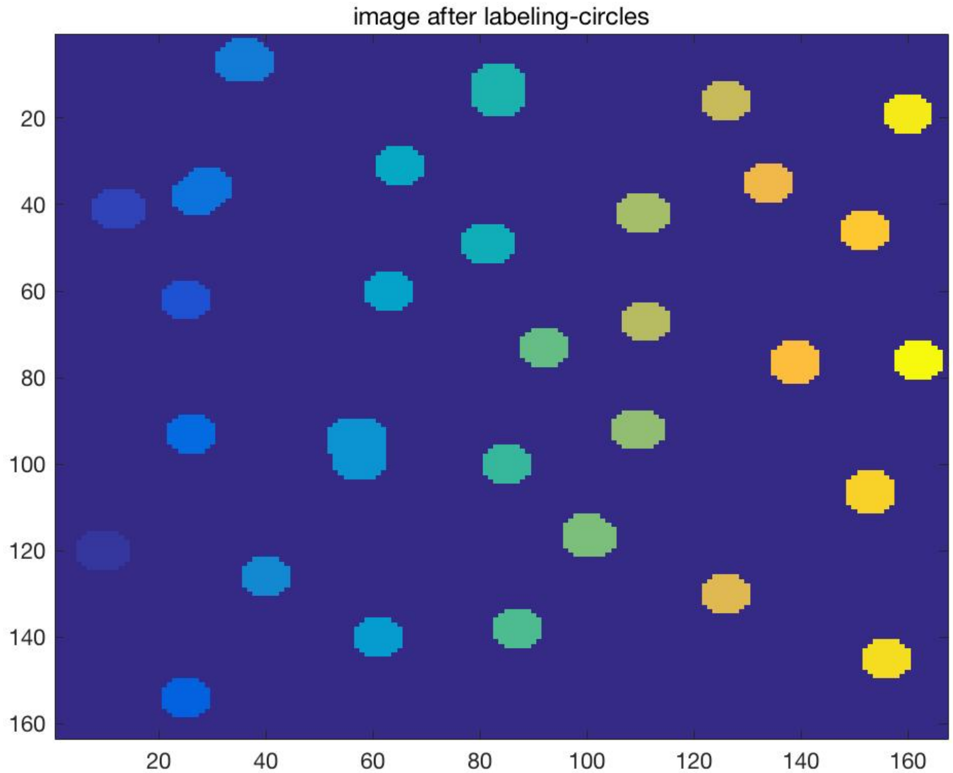
```
%Problem2
%(i)
Im_circle=imread('circles_lines.jpg');
figure(3)
imshow(Im_circle)
title('original-circles')
Im_circle=rgb2gray(Im_circle);
Im_circle=(Im_circle>125);
%imshow(Im_circle)
%type is disk, r=5
y_1=imopen(Im_circle,strel('disk',5));
figure(4)
imshow(y_1)
title('image after opening-circles')
[L_1,N_1]=bwlabel(y_1);
for i=1:N_1
    [r,c]=find(L_1==i);
    Mean_r1(i)=mean(r);
    Mean_c1(i)=mean(c);
    Area(i)=length(r);
end
figure(5)
imagesc(L_1)
title('image after labeling-circles')
disp(Mean_r1);
```

```
disp(Mean_c1);
disp(Area);
%(ii)
Im_lines=imread('lines.jpg');
figure(6)
imshow(Im_lines)
title('original-lines')
Im_lines=rgb2gray(Im_lines);
Im_lines=(Im_lines>125);
%imshow(Im_lines)
%type is line, length=8, angle=90 degree
y_2=imopen(Im_lines,strel('line',8,90));
figure(7)
imshow(y_2)
title('image after opening-lines')
[L_2,N_2]=bwlabel(y_2);
for i=1:N_2
   [r,c]=find(L_2==i);
   Mean_r2(i)=mean(r);
   Mean_c2(i)=mean(c);
   Length(i)=max(r)-min(r)+1;
end
figure(8)
imagesc(L_2)
title('image after labeling-lines')
disp(Mean_r2);
disp(Mean_c2);
disp(Length);
```

original-circles

image after opening-circles





For Image 'circle': type is disk, r=5

(i). Column index of 30 regions' centroid

(1)9.50000000000000

(2)12.5000000000000

(3)25

(4)25

(5)26

(6)28

(7)36

(8)40

(9)57.1985294117647

(10)61

(11)63

(12)65

(13)81.5000000000000

(14)83.5000000000000

(15)85

(16)87

(17)92

(18)100.376470588235

(19)109.500000000000

(20)110.500000000000

(21)111

(22)126

(23)126

(24)134

(25)139

(26)152

(27)153

(28)156

(29)160

(30)162

Row index of 30 region's centroid

(1)120

(2)41

(3)62

(4)154

(5)93

(6)37

(7)6.72826086956522

(8)126

(9)96.3235294117647

(10)140

(11)60

(12)31

(13)49

(14)13.5000000000000

(15)100

(16)138

(17)73

(18)116.623529411765

(19)92

(20)42

(21)67

(22)16

(23)130

(24)35

(25)76.5000000000000

(26)46

(27)106.500000000000

(28)145

(29)19

(30)76

Area

78
78
69
69
69
95
92
69
136
69
69
69
78
108
69
69
69
85
78
78
69
69
69
69
78
69
78
69
69
69

original-lines

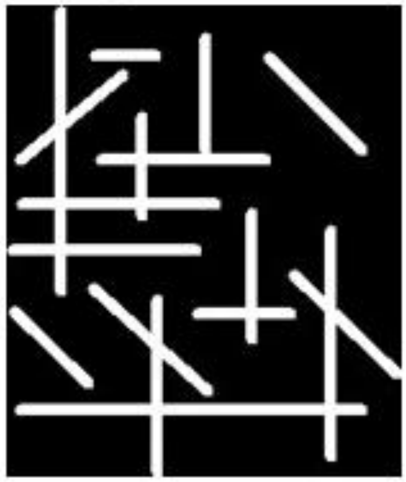
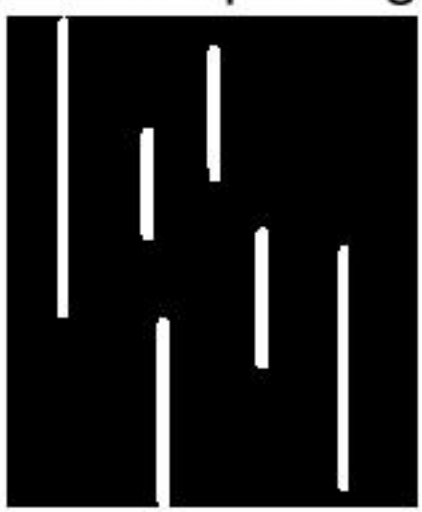
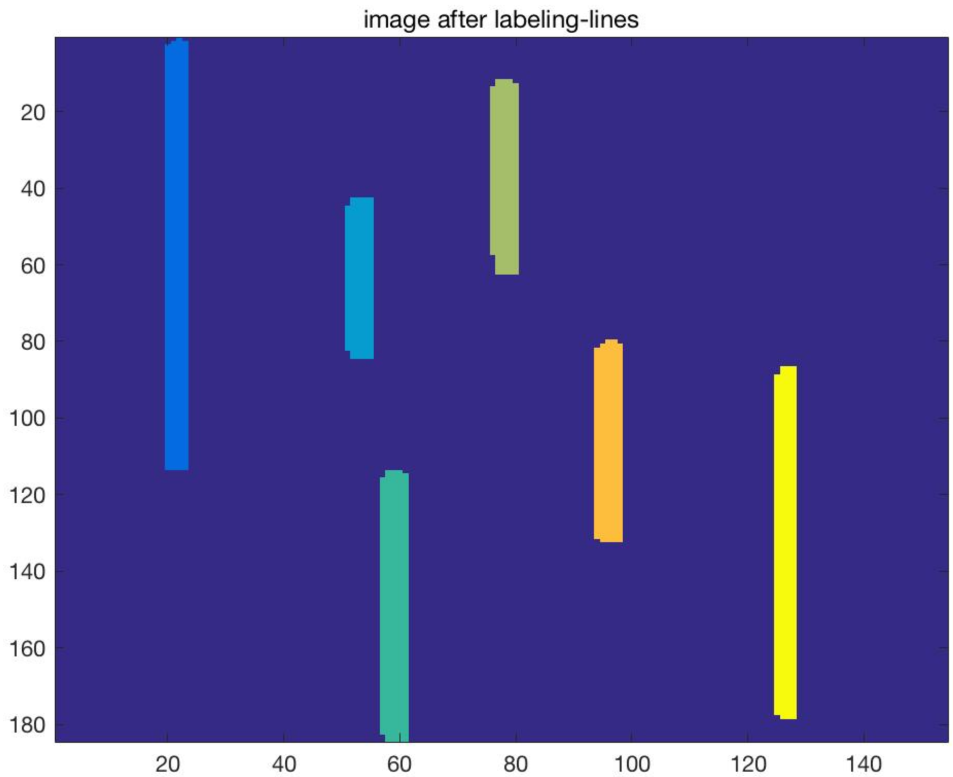


image after opening-lines





For image 'line': type is line, length=8, angle=90 degree

(ii). Column **index of 6 lines' centroid**

(1)21.5044642857143
(2)53.0388349514563
(3)59.0171428571429
(4)78.0485829959514
(5)96.0192307692308
(6)126.512328767123

Row **index of 6 lines' centroid**

(1)57.4977678571429
(2)63.5000000000000
(3)149.100000000000
(4)36.8340080971660
(5)106.296153846154
(6)132.621917808219

Length

(1)113
(2) 42
(3)71
(4)51
(5)53
(6)92

Problem3:

```
%Problem3(2)
Im_lena=imread('lena512.tif');
Im_lena=double(Im_lena);
[m_lena,n_lena]=size(Im_lena);
lena_set=double(reshape(Im_lena,1,m_lena*n_lena));

Im_diver=imread('diver.tif');
Im_diver=double(Im_diver);
[m_diver,n_diver]=size(Im_diver);
diver_set=double(reshape(Im_diver,1,m_diver*n_diver));
for s=1:7
    [pa_lena,co_lena]=lloyds(lena_set,2^s);
    pa_lena=round(pa_lena);
    co_lena=round(co_lena);
    for j=1:length(pa_lena)-1
        lena_LM(Im_lena>pa_lena(j)&Im_lena<=pa_lena(j+1))=co_lena(j+1);
    end
    lena_LM(Im_lena<=pa_lena(1))=round(co_lena(1));
    lena_LM(Im_lena>pa_lena(end))=round(co_lena(end));
    MSE_LM_lena(s)=sum(sum((lena_LM-lena_set).^2))/numel(Im_lena);
    Sq_lena=S_quantize(Im_lena,s);
    MSE_un_lena(s)=sum(sum((double(Sq_lena)-
double(Im_lena)).^2))/numel(Im_lena);



   [pa_d,co_d]=lloyds(diver_set,2^s);
```

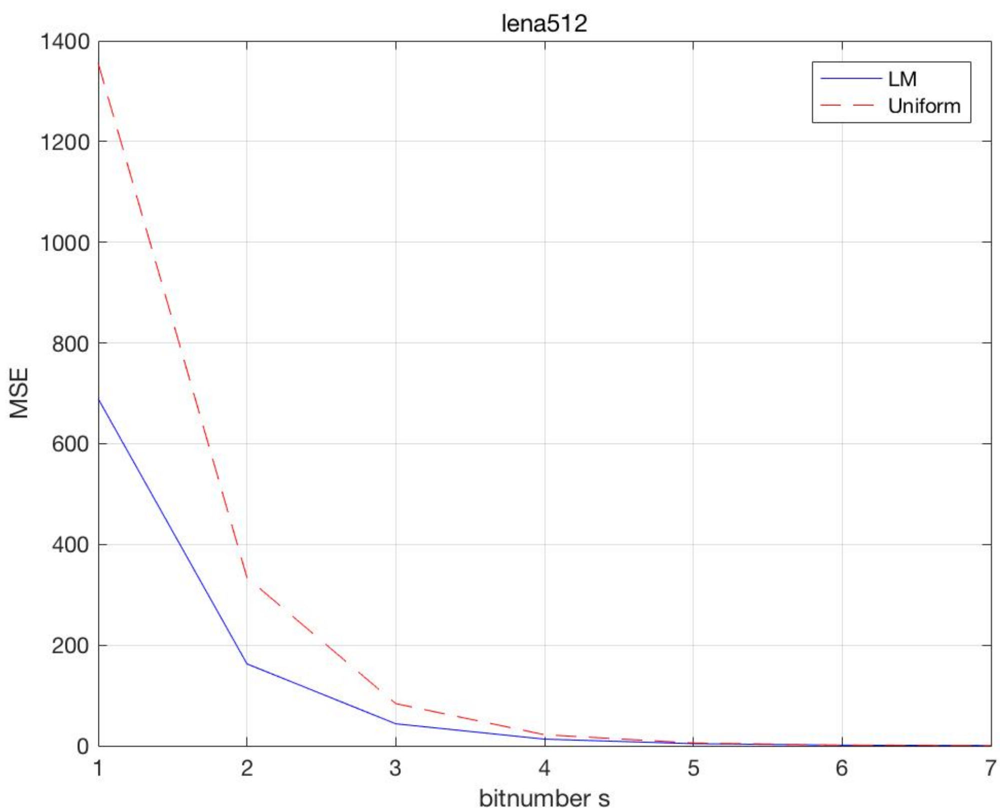
```
    pa_d=round(pa_d);
    co_d=round(co_d);
    for j=1:length(pa_d)-1
        d_LM(Im_diver>pa_d(j)&Im_diver<=pa_d(j+1))=co_d(j+1);
    end
    d_LM(Im_diver<=pa_d(1))=round(co_d(1));
    d_LM(Im_diver>pa_d(end))=round(co_d(end));
    MSE_LM_diver(s)=sum(sum((d_LM-diver_set).^2))/numel(Im_diver);
    Sq_diver=S_quantize(Im_diver,s);
    MSE_un_diver(s)=sum(sum((Sq_diver-Im_diver).^2))/numel(Im_diver);
end
figure(9)
plot(MSE_LM_lena,'b-')
hold on;
plot(MSE_un_lena,'r--')
grid on;
xlabel('bitnumber s')
ylabel('MSE')
legend('LM','Uniform')
title('lena512')
figure(10)
plot(MSE_LM_diver,'b-')
hold on;
plot(MSE_un_diver,'r--')
grid on;
xlabel('bitnumber s')
ylabel('MSE')
title('diver')
legend('LM','Uniform')
```

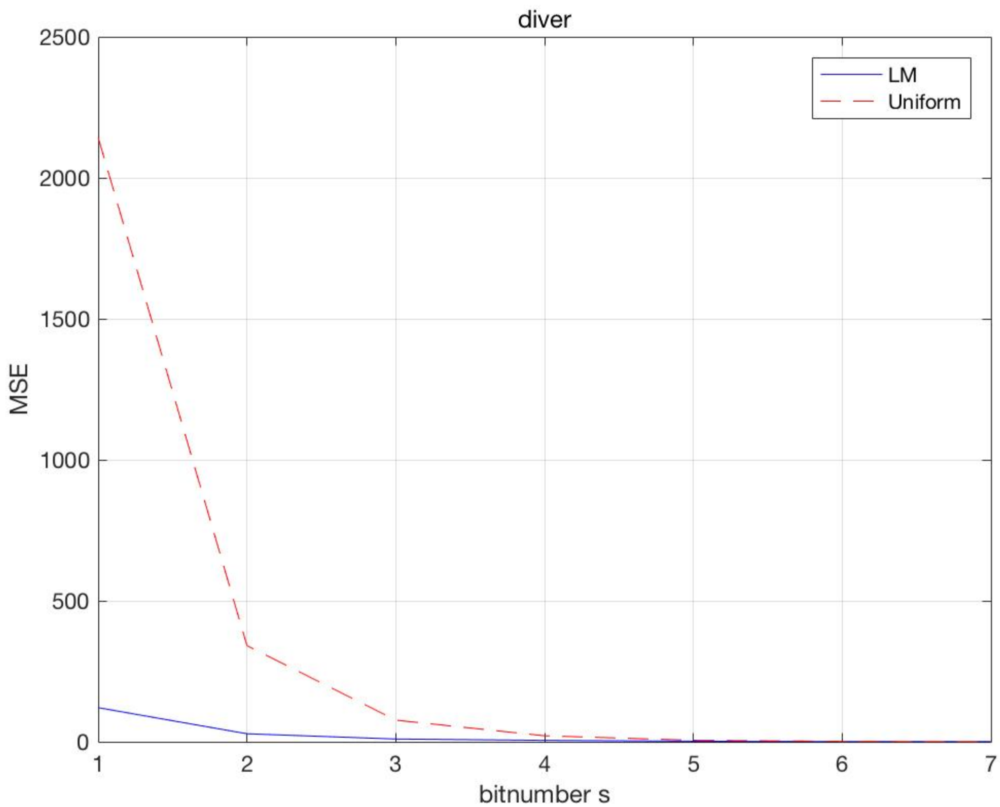
(1) function S_quantize

```
function y=S_quantize(Im,s)
Im=double(Im);
assert(s>=1&s<=7);
step=round(255/2^s);
T=0:step:255;
T(end+1)=255;
%[m,n]=size(Im);
y=zeros(size(Im));

for i=1:length(T)-1
    y(Im>T(i)&Im<=T(i+1))=round((T(i)+T(i+1))/2);
end
```

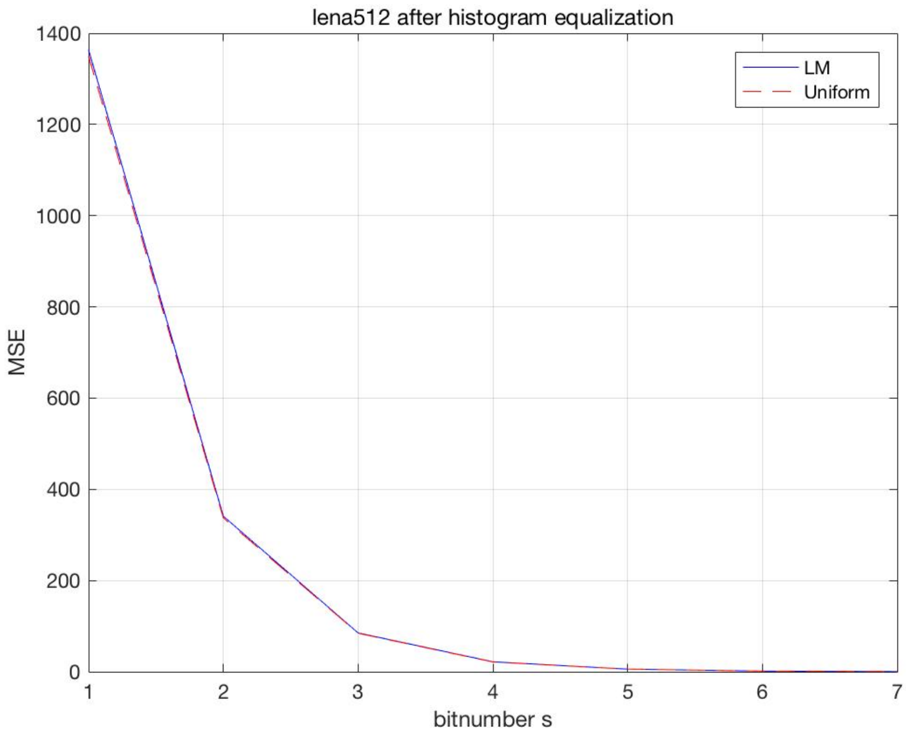
(2)

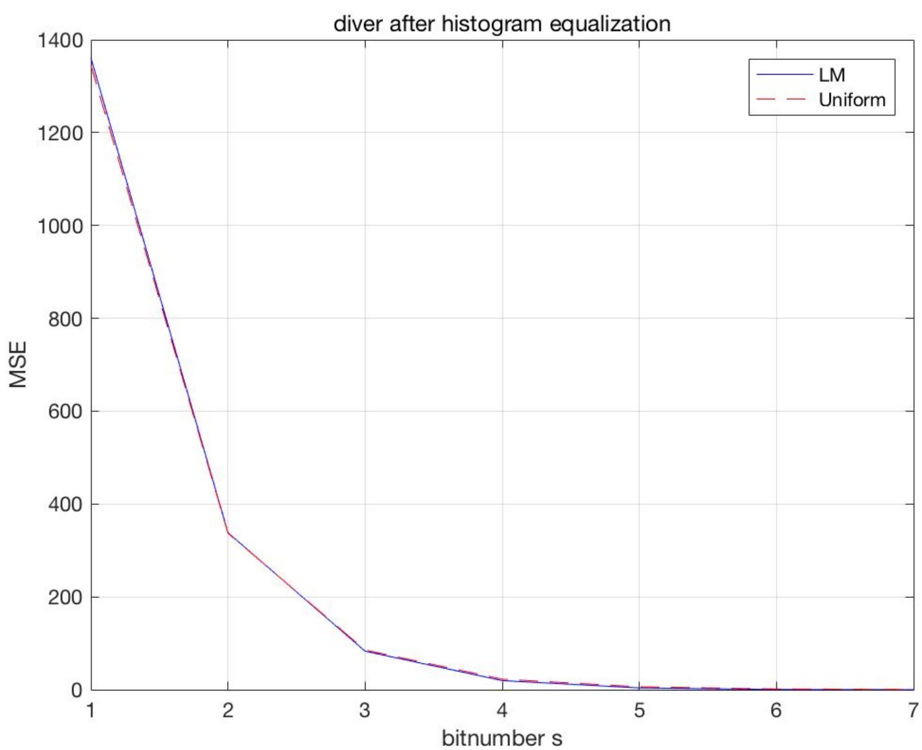




(ii)The reason why gap of ’ lena512’ between two algorithms is much smaller than that of ‘diver’ is when you compare these two pictures, ‘lena’ 512 obviously has a more uniform intensity histogram

(iii)


lena512 after histogram equalization
LM
Uniform
MSE
bitnumber s
1400
1200
1000
800
600
400
200
0
1
2
3
4
5
6
7


diver after histogram equalization
LM
Uniform
MSE
bitnumber s
1400
1200
1000
800
600
400
200
0
1
2
3
4
5
6
7

Now the gap between two algorithms can almost be ignored. This is because HE has made histogram equally distribute.

(iiii)

For 7 bits quantizer, (x and x+1) is in the same level, the max value of one pixel’ s error is 1. Furthermore, MMSE is calculated by sum of error, then divided by number of pixels, so it will be almost zero. For equalization, it cannot split the number of one gray intensity value into two parts. It just can sum the number of two gray intensity value. So, equalization actually will not make distribution truly equal because images are discrete signals.