EE 5356 DIGITAL IMAGE PROCESSING

ASSIGNMENT #2

QUANTIZATION

NAME: PAVAI ARCHIMEDES

ST ID: 1001233996

**Apply the following 3 quantization schemes to a test image (8bpp 256 level gray scale).**

**1) Uniform quantizer Calculate and compare those three quantizers (128, 64, and 32 levels) in terms of the PSNR and MSE.**

**2) Contrast quantizer (See Figure 4.21 in the text book pp.120) Use the equation (4.65) to design 3 different uniform quantizers. Let a = 1 and b =1/3. Those quantizers have the different number of quantization levels, 40, 60, and 80, respectively, and then compare in terms of the PSNR and MSE.**

**3) Pseudorandom quantizer (See Figure 4.22 in the textbook pp.121) Design a 3bit quantizer. Use three different values of A for pseudorandom noise generator and compare in terms of the PSNR and MSE.**

PROGRAM FOR UNIFORM QUANTIZER:

clc;

clear all;

N=256;

test\_image=imread('C:\Users\PAVAI ARCHIMEDES\Desktop\cameraman.bmp');

Image\_new=double(test\_image);

[row col]=size(Image\_new);

for count=5:7

L= 2^count;

q = ((N+1)-1)/L;

for k=1:L+1

t(k)=256\*(k-1)/L;

end

%Reconstruction Level values given by

for k=1:L

r(k) = t(k) + q/2;

end

%Quantization

for m=1:row

for n=1:col

for k=1:L

if t(k)<=Image\_new(m,n)

if Image\_new(m,n)<= t(k+1)

q(m,n)=r(k);

else

q(m,n)=Image\_new(m,n);

end

end

end

end

end

%Calculate MSE

MSE=0;

for m=1:row

for n=1:col

MSE\_new=((Image\_new(m,n)-q(m,n))^2); MSE=MSE+MSE\_new; %summing up MSE end

end

end

MSE=(MSE/N^2); %taking mean of the total MSE %Calculate PSNR in dB

PSNR\_dB=10\*log10(((N-1)^2)/MSE);

fprintf('L\t\t\t\t\tMSE\t\t\t\t\tPSNR\_dB\n');

fprintf('%d\t\t\t\t%f\t\t\t\t%f\n',L,MSE,PSNR\_dB);

img=imshow(q,[]);

uint8(img); %display the image in uint8 format

title(['For Quantization interval L=', int2str(L)]);

figure;

end

**OUTPUT:**

L MSE PSNR\_dB

32 5.479950 40.743038

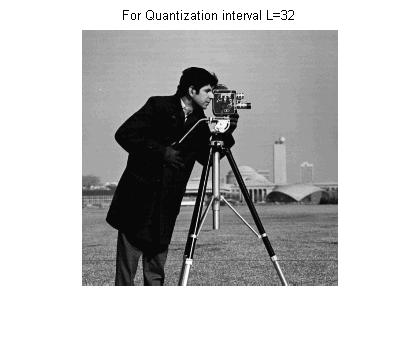
L MSE PSNR\_dB

64 1.488861 46.402262

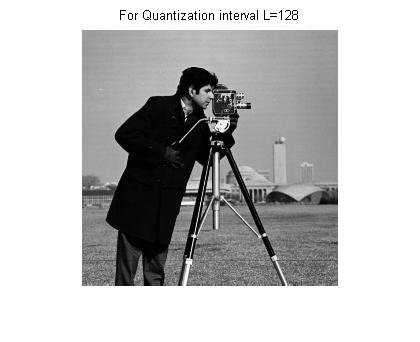
L MSE PSNR\_dB

128 0.502594 51.118631









CONTRAST QUANTIZER:

clc;

clear all;

N=256;

test\_image=imread('C:\Users\PAVAI ARCHIMEDES\Desktop\cameraman.bmp'); % Read the image

imshow(test\_image); % Display the image

title('Cameraman-Original Test Image');

figure;

Image\_new=double(test\_image); % convert the unit8 image into floating point

[row col]=size(Image\_new);

for L = 40:20:80 % Quantization levels defined

alpha=1;

beta=1/3

for x=1:row

for y=1:col

c(x,y)=alpha\*((Image\_new(x,y))^1/3);

end

end

% Transition Level values given by

for k=1:L+1

t(k)=6.32\*(k-1)/L;

end

% Reconstruction Level values given by

for k=1:L

r(k)=t(k)+3.14/L;

end

%Quantization takes place by a MMSE quantizer

for m=1:row

for n=1:col

for k=1:L

if t(k)<=c(m,n)

if c(m,n)<= t(k+1)

q(m,n)=r(k);

else

q(m,n)=c(m,n);

end

end

end

end

end

for x=1:row

for y=1:col

c1(x,y)=(q(x,y)/alpha)^(1/beta);

end

end

%Calculate MSE

MSE=0;

for i=1:row

for alpha=1:col

MSE\_new=((Image\_new(i,alpha)-c1(i,alpha))^2);

MSE=MSE+MSE\_new;

end

end

MSE=(MSE/N^2);

%Calculate PSNR in dB

PSNR\_dB=10\*log10(((N-1)^2)/MSE);

fprintf('L\t\t\t\t\tMSE\t\t\t\t\tPSNR\_dB\n');

fprintf('%d\t\t\t\t%f\t\t\t\t%f\n',L,MSE,PSNR\_dB);

img=imshow(q,[]);

uint8(img);

title(['For Quantization interval L=', int2str(L)]);

figure;

end;

**OUTPUT:**

L MSE PSNR\_dB

40 18759485215.867954 -54.601406

L MSE PSNR\_dB

60 18759485242.319183 -54.601406

L MSE PSNR\_dB

80 18759485255.367161 -54.601406





PSEDORANDOM QUANTIZER:

clc;

clear all;

N=256;

test\_image=imread('C:\Users\PAVAI ARCHIMEDES\Desktop\cameraman.bmp');

imshow(test\_image);% Display the image

title('Cameraman-Original Test Image');

figure;

disp(test\_image);

Image\_new=double(test\_image); % convert the unit8 image into floating point image for calculations

[row col]=size(Image\_new);

L=8;

disp L;

disp(L);

q1 = ((N+1)-1)/L;

for A=25:30:60

pr=randn(256,256);

pr=A\*double(pr);

for m=1:row

for n=1:col

q(m,n)=Image\_new(m,n)+pr(m,n); % Adding the pseudorandom noise to the image before quantisation

end

end

for k=1:L+1

t(k)=256\*(k-1)/L;

end

%Reconstruction Level values given by

for k=1:L

r(k) = t(k) + q1/2;

end

%Quantisation

for m=1:row

for n=1:col

for k=1:L

if t(k)<=q(m,n)

if q(m,n)<= t(k+1)

v(m,n)=r(k);

else

v(m,n)=q(m,n);

end

end

end

end

end

for m=1:row

for n=1:col

u(m,n)=v(m,n)-pr(m,n);%pseudo random noise is subtracted from the quantized output end

end

end

%Calculate MSE

MSE=0;

for m=1:row

for n=1:col

MSE\_new=((Image\_new(m,n)-u(m,n))^2);

MSE=MSE+MSE\_new; % summing up MSE

end

end

MSE=(MSE/N^2); %mean of MSE %Calculate PSNR in dB

PSNR\_dB=10\*log10(((N-1)^2)/MSE); fprintf('A\t\t\t\t\tMSE\t\t\t\t\tPSNR\_dB\n');

fprintf('%d\t\t\t\t%f\t\t\t\t%f\n',A,MSE,PSNR\_dB);

img =imshow(v,[]);

uint8(img);

title(['For Quantization interval L=8 and A=', int2str(A)]);

figure;

end;

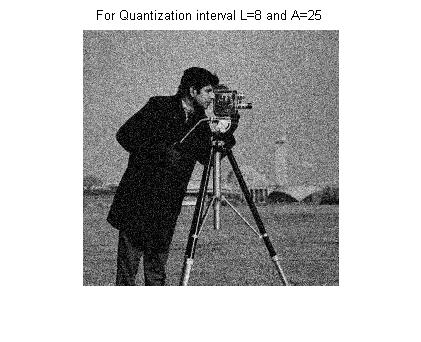
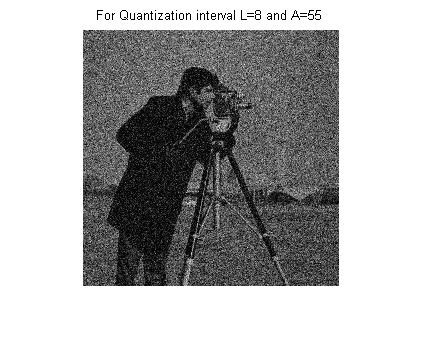
**OUTPUT:**

A MSE PSNR\_dB

25 105.737117 27.888529

A MSE PSNR\_dB

55 645.060047 20.034802

**CONCLUSION:**

**UNIFORM QUANTIZER:**

In this quantizer, the number of quantization levels affect both the mean square error and PSNR. MSE decreases while PSNR increases with increase in the number of quantization levels.

**CONTRAST QUANTIZER:**

Since visual sensitivity is nearly uniform to just noticeable changes in contrast, it is more appropriate to quantize the contrast function. Two nonlinear transformations that have been used for representation of contrast c are

c =alpha ln (l + beta \* u),

c = alpha \* u^beta

Where alpha and beta are constants and u represents the luminance.

**PSEUDORANDOM QUANTIZER:**

Contouring effects can be suppressed by adding a small amount of uniformly distributed pseudorandom noise to the luminance samples before quantization. This pseudorandom noise is also called dither.