EE 5356 Digital Image Processing

LAB Assignment 2

by

Vasudevan Murali

100723391

1. Uniform Quantizer :
   1. Matlab program :

clc

clear all;

%read image

iImage1 = imread('goldhill256.bmp');

[row col] = size(iImage1);

% quantization level

L = 128;

% determining limits

t = 0:256/L:256;

% quantization

for i = 1:row

for j = 1:col

for k = 1:L

if(iImage1(i,j)>=t(k) && iImage1(i,j) < t(k+1))

iImage2(i,j) = (t(k) + t(k+1))/2;

end

end

end

end

%MSE calculation

MSE = 0;

for i = 1:256

for j = 1:256

MSE = MSE + (double(iImage1(i,j)) - iImage2(i,j))^2;

end

end

MSE = double(MSE / (256^2));

%PSNR calculation

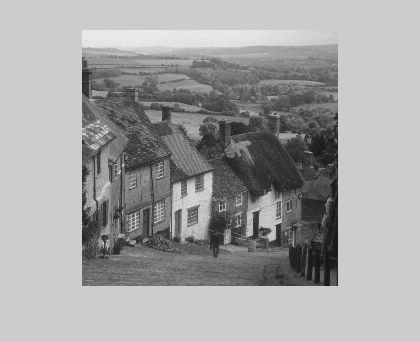
PSNR = 10 \* log10( 255^2 / MSE );

imshow(iImage1);

figure(2);

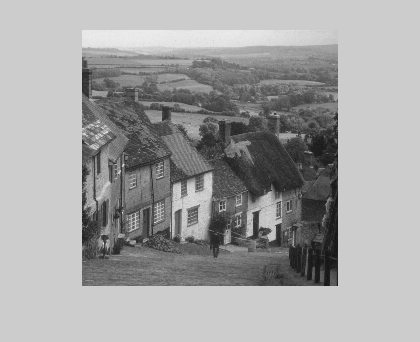
imshow(uint8(iImage2));

* 1. Input image:

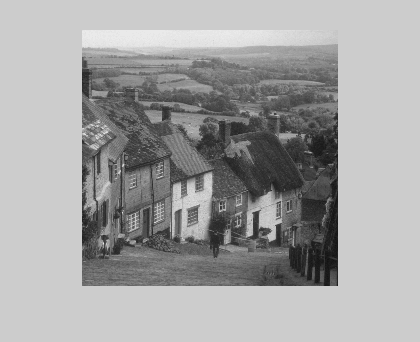


* 1. Output:

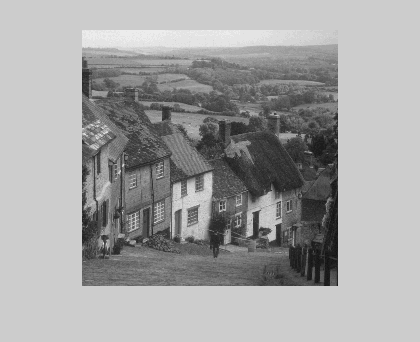
For L = 128, MSE = 0.5015, PSNR = 51.1280



For L = 64, MSE = 1.5006, PSNR = 46.3683



For L = 32, MSE = 5.5036, PSNR = 40.7244



|  |  |  |
| --- | --- | --- |
| Conditions | MSE | PSNR |
| L = 128 | 0.5015 | 51.1280 |
| L = 64 | 1.5006 | 46.3683 |
| L = 32 | 5.5036 | 40.7244 |

1. Contrast Quantizer
2. Matlab program :

%read image

iImage1 = imread('goldhill256.bmp');

[row col] = size(iImage1);

% quantization level

L = 40;

% setting contrast equation values

a=1;

b=1/3;

contrastImage = zeros(row,col);

iImage2 = zeros(row,col);

% determining limits

t = 0:(a \* (256^b))/L:(a \* (256^b));

lenT = length(t);

% contrast quantization

for i = 1:row

for j = 1:col

sample = double(iImage1(i,j));

% luminance to contrast

sample = a\*(sample^b);

% quantization

for k = 1:lenT-1

if(sample >= t(k) && sample < t(k+1))

contrastImage(i,j) = (t(k) + t(k+1)) / 2;

end

end

% contrast to luminance

iImage2(i,j) = (contrastImage(i,j)/a) ^(1/b);

end

end

%MSE calculation

MSE = 0;

for i = 1:row

for j = 1:col

MSE = MSE + (double(iImage1(i,j)) - iImage2(i,j))^2;

end

end

MSE = double(MSE / (row\*col));

%PSNR calculation

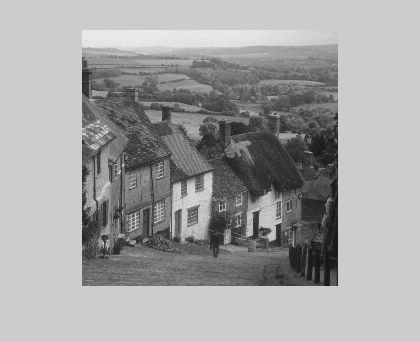
PSNR = 10 \* log10( 255^2 / MSE );

imshow(iImage1);

figure(2);

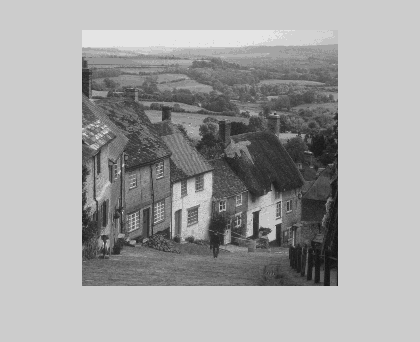
imshow(uint8(iImage2));

1. Input Image:

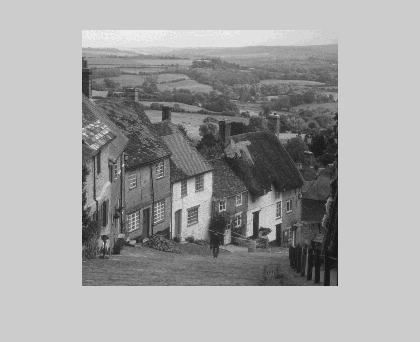


1. Output:

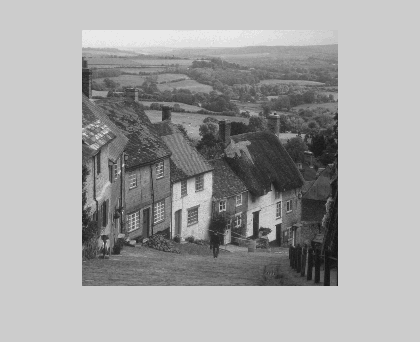
For L = 40, MSE = 10.0243 and PSNR = 38.1203



For L = 60, MSE = 4.8899 and PSNR = 41.2378



For L = 80, MSE = 2.6359 and PSNR = 43.9216



|  |  |  |
| --- | --- | --- |
| Conditions | MSE | PSNR |
| L = 40 | 10.0243 | 38.1203 |
| L = 60 | 4.8899 | 41.2378 |
| L = 80 | 2.6359 | 43.9216 |

1. Pseudorandon Quantizer
   1. Matlab code :

clc

clear all;

%read image

iImage1 = imread('goldhill256.bmp');

% noise limit

A = 15;

% 3 bit quantizer level

L = 8;

% setting intervals

t = 0:256/L:256;

lenT = length(t);

% generating psuedorandom noise

noise = A .\* ( -1 + (1-(-1)) .\* rand(256,256));

% adding noise to image

iImage2 = double(iImage1) + noise;

% quantization

for i = 1:256

for j = 1:256

sample = double(iImage2(i,j));

for k = 1:lenT-1

if(sample >= t(k) && sample < t(k+1))

iImage3(i,j) = (t(k) + t(k+1)) / 2;

end

end

end

end

iImage4 = iImage3 - noise;

%MSE calculation

MSE = 0;

for i = 1:256

for j = 1:256

MSE = MSE + (double(iImage1(i,j)) - iImage4(i,j))^2;

end

end

MSE = double(MSE / (256^2));

%PSNR calculation

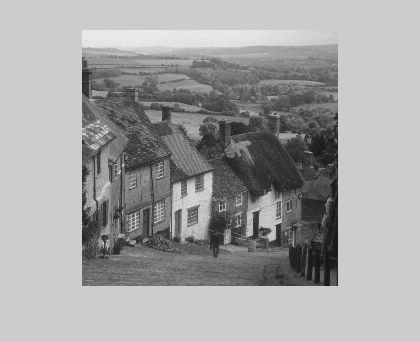
PSNR = 10 \* log10( 255^2 / MSE );

imshow(iImage1);

figure(2);

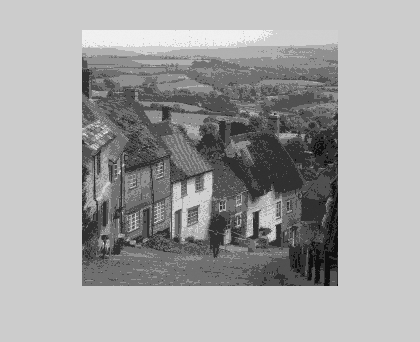
imshow(uint8(iImage4));

* 1. Input :

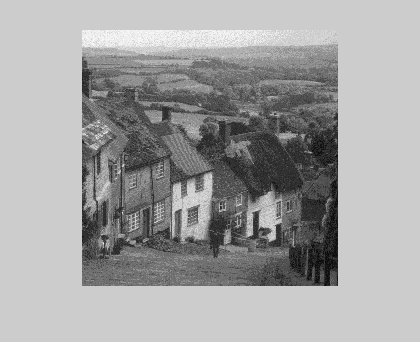


* 1. Output:

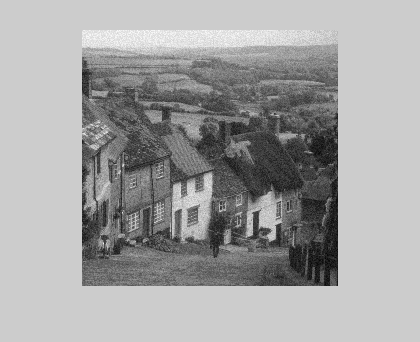
For A = 1, MSE = 85.9605 and PSNR = 28.7878



For A = 10, MSE = 85.7254 and PSNR = 28.7997



For A = 20, MSE = 84.5380 and PSNR = 28.8603



In this case, the MSE and PSNR change randomly even for the same value of A.

|  |  |  |
| --- | --- | --- |
| Conditions | MSE | PSNR |
| A=1 | 85.9605 | 28.7878 |
| A=10 | 85.7254 | 28.7997 |
| A=20 | 84.5380 | 28.8603 |