EE 5351 DIGITAL VIDEO CODING\_ ASSIGNMENT

VECTOR QUANTIZATION

NAME: PAVAI ARCHIMEDES

ST ID: 1001233996

1), Generate a 16-dimensional codebook of size 64 for the Sena image. Construct the vector as a 4x4 block of pixels, an 8 x 2 block of pixels, and a 16x1 block of pixels.

2), Show this table:

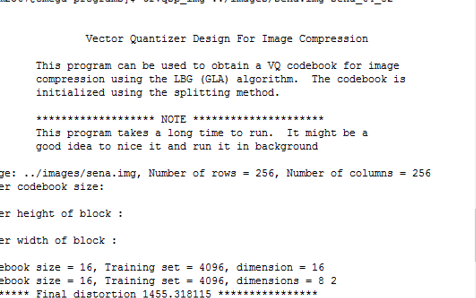
|  |  |  |  |
| --- | --- | --- | --- |
|  | SNR (dB) | PSNR(dB) | Compression Ratio |
| 4 x 4 |  |  |  |
| 8 x 2 |  |  |  |
| 16 x 1 |  |  |  |

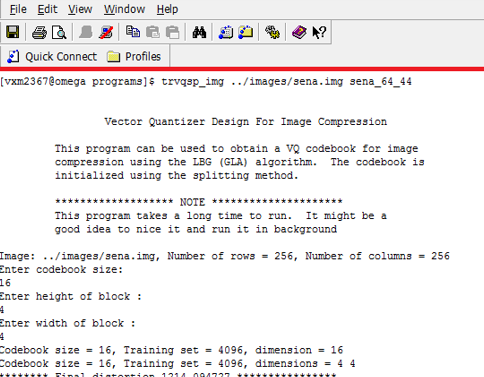
3), Show the reconstructed images.

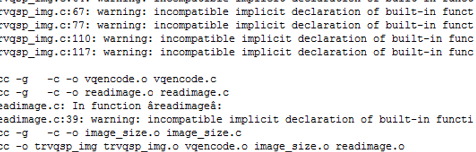
4), Show your source code.

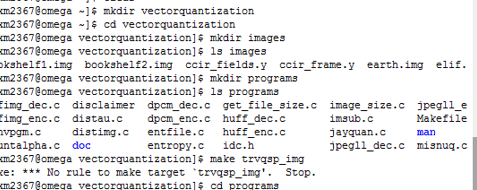
5), Repeat the whole thing for codebook size of 256.

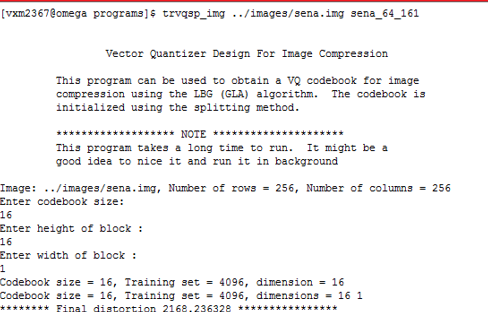
SOLUTION:



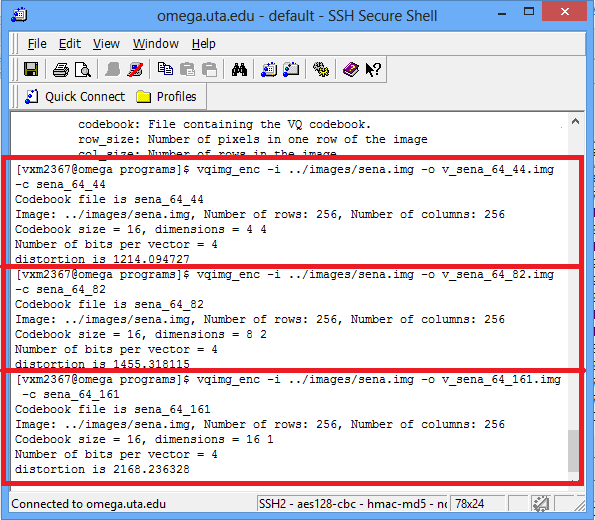


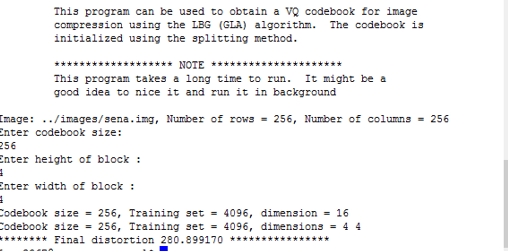


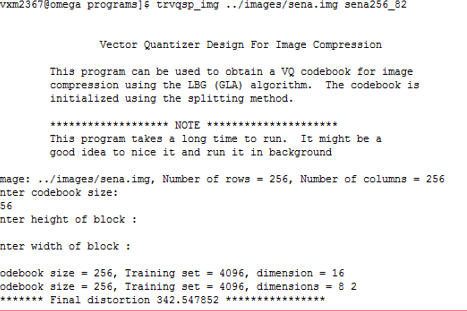


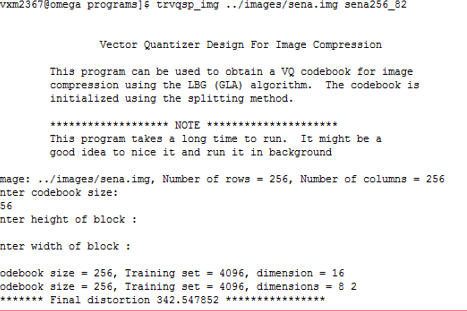


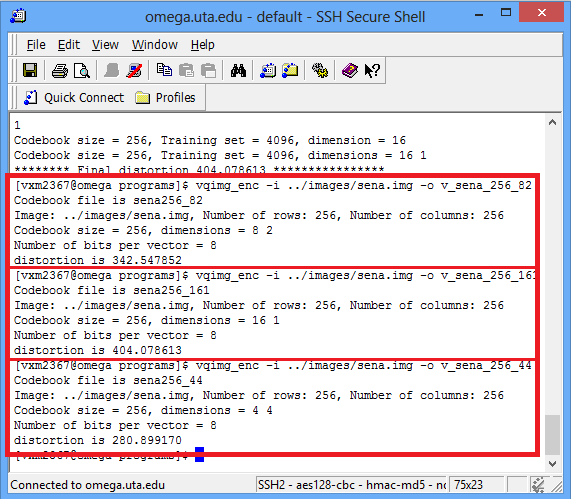
The distortion,codebook size and number of vector bits are manipulated:

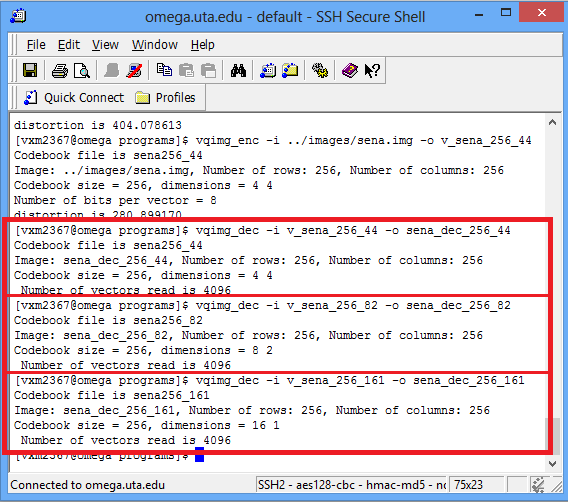












MATLAB CODE FOR 64:

clc;

close all;

clear all;

fid = fopen('sena.img');

Image = fread(fid,[256,256]);

img = Image';

subplot(2,2,1);

imshow(uint8(img));

title('ORIGINAL IMAGE');

fig1 = fopen('sena\_dec\_64\_44');

Image1 = fread(fig1,[256,256]);

sena\_rec = Image1';

subplot(2,2,2);

imshow(uint8(sena\_rec));

title('RECONSTRUCTED IMAGE OF BLOCK SIZE 4\*4');

ms = sum(sum((img - sena\_rec).^2))/(65536);

si = sum(sum((img).^2))/(65536);

snr\_44 = 10\*log10(si/ms)

peak = max(max((img).^2));

psnr\_44 = 10\*log10(peak/ms);

fi2 = fopen('sena\_dec\_64\_82');

Image2 = fread(fi2,[256,256]);

sena1 = Image2';

subplot(2,2,3);

imshow(uint8(sena1));

title('RECONSTRUCTED IMAGE OF BLOCK SIZE 8\*2');

ms = sum(sum((img - sena1).^2))/(65536);

si1 = sum(sum((img).^2))/(65536);

snr\_82 = 10\*log10(si1/ms)

peak\_signal = max(max((img).^2));

psnr\_82 = 10\*log10(peak\_signal/ms)

fi3 = fopen('sena\_dec\_64\_161');

Image3 = fread(fi3,[256,256]);

sena22 = Image3';

subplot(2,2,4);

imshow(uint8(sena22));

title('RECONSTRUCTED IMAGE OF BLOCK SIZE 16\*1');

mss = sum(sum((img - sena22).^2))/(65536);

sigg = sum(sum((img).^2))/(65536);

snr\_161 = 10\*log10(sigg/mss)

peak\_signal2 = max(max((img).^2));

psnr\_161= 10\*log10(peak\_signal2/mss)

OUTPUT:



|  |  |  |  |
| --- | --- | --- | --- |
|  | SNR(DB) | PSNR(DB) | Compression Ratio |
| 4x4 | 20.2485 | 27.6431 | 31.64 |
| 8x2 | 19.4614 | 26.8560 | 31.64 |
| 16x1 | 15.3951 | 22.7897 | 31.62 |

CODE FOR SIZE 256:

clc;

close all;

clear all;

fid = fopen('sena.img');

Image = fread(fid,[256,256]);

img = Image';

subplot(2,2,1);

imshow(uint8(img));

title('ORIGINAL IMAGE');

fig1 = fopen('sena\_dec\_256\_44');

Image1 = fread(fig1,[256,256]);

sena\_rec = Image1';

subplot(2,2,2);

imshow(uint8(sena\_rec));

title('RECONSTRUCTED IMAGE OF BLOCK SIZE 4\*4');

ms = sum(sum((img - sena\_rec).^2))/(65536);

si = sum(sum((img).^2))/(65536);

snr\_44 = 10\*log10(si/ms)

peak = max(max((img).^2));

psnr\_44 = 10\*log10(peak/ms);

fi2 = fopen('sena\_dec\_256\_82');

Image2 = fread(fi2,[256,256]);

sena1 = Image2';

subplot(2,2,3);

imshow(uint8(sena1));

title('RECONSTRUCTED IMAGE OF BLOCK SIZE 8\*2');

ms = sum(sum((img - sena1).^2))/(65536);

si1 = sum(sum((img).^2))/(65536);

snr\_82 = 10\*log10(si1/ms)

peak\_signal = max(max((img).^2));

psnr\_82 = 10\*log10(peak\_signal/ms)

fi3 = fopen('sena\_dec\_256\_161');

Image3 = fread(fi3,[256,256]);

sena22 = Image3';

subplot(2,2,4);

imshow(uint8(sena22));

title('RECONSTRUCTED IMAGE OF BLOCK SIZE 16\*1');

mss = sum(sum((img - sena22).^2))/(65536);

sigg = sum(sum((img).^2))/(65536);

snr\_161 = 10\*log10(sigg/mss)

peak\_signal2 = max(max((img).^2));

psnr\_161= 10\*log10(peak\_signal2/mss)

OUTPUT:



|  |  |  |  |
| --- | --- | --- | --- |
|  | SNR(DB) | PSNR(DB) | Compression Ratio |
| 4x4 | 26.6055 | 34.0001 | 15.91 |
| 8x2 | 25.7438 | 33.1384 | 15.91 |
| 16x1 | 18.0988 | 25.4934 | 15.90 |