**Image and Video Processing**

**Programming Assignment**

**Week 10 – Wavelet Transform**

2

function [ nonZero, PSNR, outImg] = dwtQuant2( x,wname, stepsize )

% Function to quantize wavelet coefficients and get PSNR and non-zero

% count for various quantization step sizes.

% Get a 3 level wavelet decomposition

[C S] = wavedec2(x,3,wname);

% Uniform quantization

for i = 1:length(C)

% Check if first coefficient, in which case use mean = 128

if i == 1

mean = 128;

else

mean = 0;

end

quantC(i) = floor((C(i)-mean+0.5\*stepsize)/stepsize)\*stepsize+mean;

end

% Reconstruct using quantized coefficients

outImg = waverec2(quantC,S,wname);

% Counting number of nonZero approximation coefficients after quantization in the bottom level

nonZero = 0;

for i = 1:length(quantC)

if quantC(i) ~= 0

nonZero = nonZero+1;

else

continue;

end

end

disp('Number of Non-Zero Coefficients: ');

disp(nonZero);

% Computing PSNR

PSNR = psnr(outImg,double(x));

disp('PSNR: ');

disp(PSNR);

subplot(1,2,1), imshow(uint8(x)); title('Original');

subplot(1,2,2), imshow(uint8(outImg)); title('Quantized');

end

% quantScript.m

% Read an image

x = imread('hands.jpg');

% Extract grayscale version

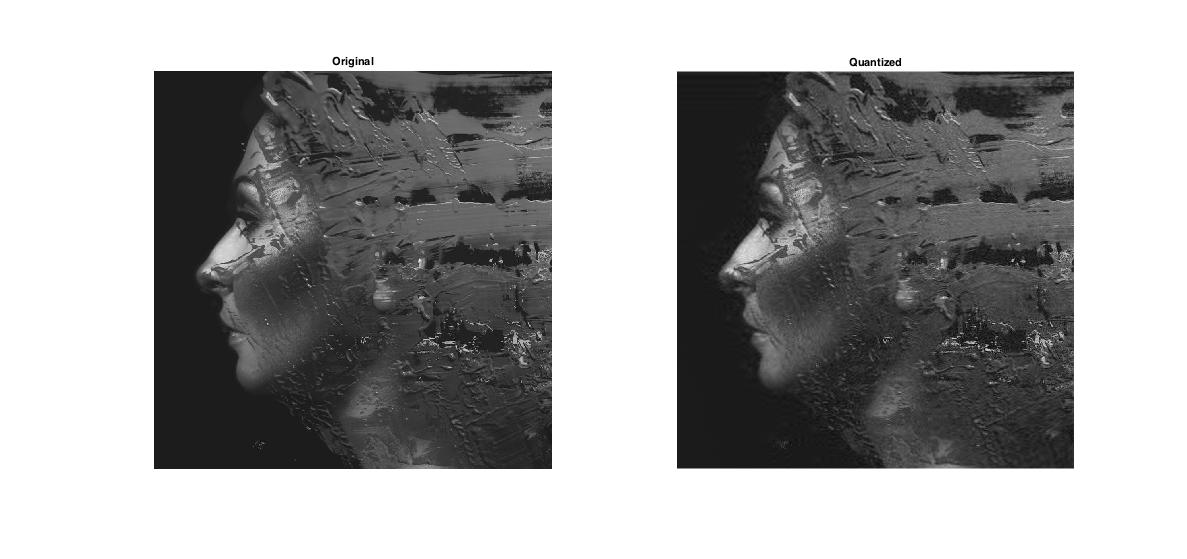
ycbcr = rgb2ycbcr(x);

gray = ycbcr(:,:,1);

% imshow(gray);

% APply function

[nonZ, PSNR, out] = dwtQuant2(gray,'db45',51);



\*Notice the brushstrokes getting blurred in the quantized image.

3

% Comparison Script

% Read an image

x = imread('hands.jpg');

% Extract grayscale version

ycbcr = rgb2ycbcr(x);

gray = ycbcr(:,:,1);

% Initializing

stepsize = [1 4 16 32];

nonZeroH = zeros(1,length(stepsize));

PSNRH = zeros(1,length(stepsize));

nonZeroDB = zeros(1,length(stepsize));

PSNRDB = zeros(1,length(stepsize));

% Computing Non-Zero and PSNR values

for i = 1:length(stepsize)

[nonZeroH(i), PSNRH(i), outH] = dwtQuant2(gray,'haar',stepsize(i));

[nonZeroDB(i), PSNRDB(i), outDB] = dwtQuant2(gray,'db25',stepsize(i));

end

figure;

hold on;

plot(stepsize,nonZeroH/(10^4));

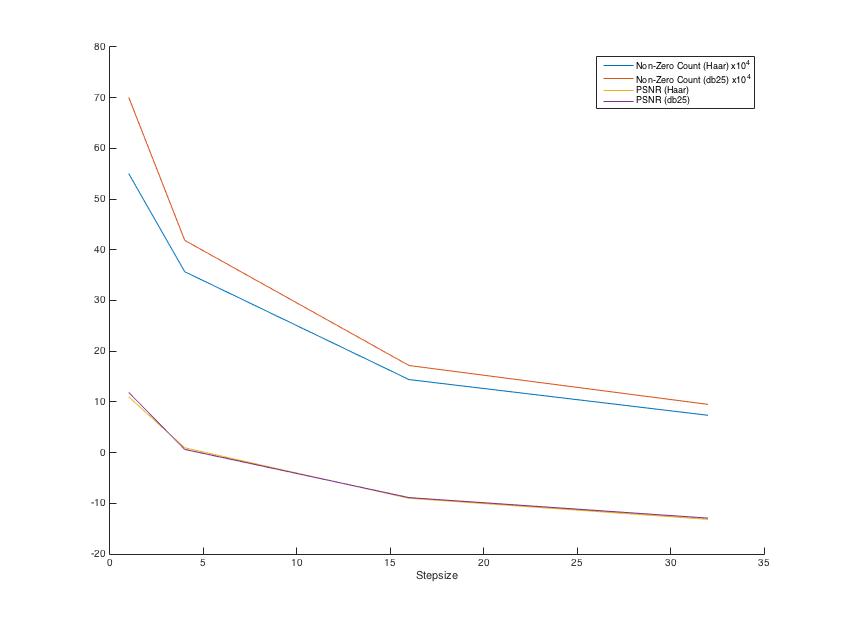
plot(stepsize,nonZeroDB/(10^4));

plot(stepsize,PSNRH);

plot(stepsize,PSNRDB);

xlabel('Stepsize')

legend('Non-Zero Count (Haar) x10^4','Non-Zero Count (db25) x10^4','PSNR (Haar)','PSNR (db25)');



For a given step size, Haar wavelet gives better quality (smaller step size) for the same bitrate.

For a given image quality (in terms of PSNR), db25 takes marginally lower step size, however takes a substantially larger number of bits as it has greater number of non-zero terms.