**Final Project CECS 627 Fall 2019**

**Two-level discrete 2-D wavelet transform**

The purpose of the project is to implement and compare of two image compression routine:

1. Lossless two-level 2-DWLT
2. Lossy two-level 2-DWLT

Image: lenna.jpg

Wavelet Name: db1, db2, db4, db8, db16, db32

Deliverables:

1. Plot Wavelet Name vs. Compression Ratio for Lossless system
2. Plot Wavelet Name vs. Compression Ratio for Lossy System
3. Plot Wavelet Name vs. SNR in dBs for Lossy System

Here is an example of compressing the input image using wavelet name ‘coif5’ which is not one

Of the assigned wavelets

>> % the four filters of wname=’coif5’ could be explored by the following instruction

>> [h0 h1 f0 f1]=wfilters('coif5');

>> size(h0)

ans =

1. 30

>>% Each one of them has 30 coeffecients. This will increase the size of the wavelet transformation 4 outputs

>> %let’s get the image

>> f=mat2gray(imread('lenna.jpg'));

>> sz1=size(f)

sz1 =

512 512

>> % perform Single-level 2D WLT

>> [LL LH HL HH]=dwt2(f,'coif5');

>> sz2=size(LL)

sz2 =

270 270

>> % each one of the 4 quadrant is little bit more than the expected size of 256x256. This is

>> % because convolution increases the size of the output samples. However, the reconstructed

>> % will go back to the same size.

>> % next we do Two Level 2D WLT by transforming the quadrant LL

>> [LL1 LH1 HL1 HH1]=dwt2(LL,'coif5');

>> sz3=size(LL1)

sz3 =

149 149

>>% each one of the 1/16 of the image is more than the expected 128x128 even more than ¼ of >>%270x270 or 135x135. This is because of convolution too

>>% therefore we can’t concatenate them in one picture without trimming the pixels at the

>> % boundaries

>>% that is the reason this project didn’t ask to show the compressed image. However, the input

>>% image f, is transformed to LL1, LH1, HL1, HH1, LH, HL, and HH. The number of

>>% data points at the input is 512x512 = 262144 and the number of data at the output is

>>% 4\*149\*149+3\*270\*270 = 307504

>>%Let’s reconstruct and analyze the lossless image

>> LL2=idwt2(LL1,LH1,HL1,HH1,'coif5',sz2);

>> g=idwt2(LL2,LH,HL,HH,'coif5',sz1);

>> SNR(g,f)

ans =

175.2546

>>%the computed SNR is too high because g is lossless. Now, let’s compute the compr. Ratio

>>%this requires the computation of the entropy of the 7 output sub images

>>% LL1, LH1, HL1, HH1, LH, HL, and HH. Since we didn’t put them in one image

>>%we can’t use SummaryStat.m function. However, we will compute the histogram

>>%of each one of the 7 sub images. Then add the 7 histograms.

>> h1=hist(reshape(LL1,size(LL1,1)\*size(LL1,2),1),256);

>> h2=hist(reshape(LH1,size(LH1,1)\*size(LH1,2),1),256);

>> h3=hist(reshape(HL1,size(HL1,1)\*size(HL1,2),1),256);

>> h4=hist(reshape(HH1,size(HH1,1)\*size(HH1,2),1),256);

>> h5=hist(reshape(LH,size(LH,1)\*size(LH,2),1),256);

>> h6=hist(reshape(HL,size(HL,1)\*size(HL,2),1),256);

>> h7=hist(reshape(HH,size(HH,1)\*size(HH,2),1),256);

>> h=h1+h2+h3+h4+h5+h6+h7;

>> sum(h)

ans =

307504

>>% this the number of the output data points. Let’s compute the entropy

>> h=h/sum(h);e=0;L2=log(2);for i=1:256 if h(i) > 0 e=e-h(i)\*log(h(i))/L2;end;end;

>> e

e =

5.7805

>> Comp\_Ratio = 8\*512\*512/(e\*307504)

Comp\_Ratio =

1.1798

>>%Let’s reconstruct and analyze the Lossy image. Here the image be reconstructed using LL1 >>%only. The rest will be zeros

>> z1=zeros(sz3);

>> z2=zeros(sz2);

>> LL2=idwt2(LL1,z1,z1,z1,'coif5',sz2);

>> g1=idwt2(LL2,z2,z2,z2,'coif5',sz1);

>> SNR(g1,f)

ans =

22.9830

>> this is not bad SNR in decibels. We need to compute the entropy of LL1. Its histogram is h1

>> sum(h1)

ans =

22201

>> %This is 149x149

>> 149\*149

ans =

22201

>> h1=h1/sum(h1);e=0;L2=log(2);for i=1:256 if h1(i) > 0 e=e-h1(i)\*log(h1(i))/L2;end;end;

>> e

e =

7.5681

>> Comp\_Ratio = 8\*512\*512/(e\*22201)

Comp\_Ratio =

12.4815

% This is very high compression ratio