Final Project: Two-level Discrete 2-D Wavelet Transform

Jacob Taylor Cassady

CECS 627: Digital Image Processing

Fall 2019

# Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Wavelet Name** | **Wavelet Enum** | **Lossless SNR** | **Lossless Compression Ratio** | **Lossy SNR** | **Lossy Compression Ratio** |
| coif5 | 1 | 403.5385 | 1.1798 | 52.9706 | 12.4815 |
| db1 | 2 | 701.1841 | 1.3949 | 45.9911 | 16.5504 |
| db2 | 3 | 582.5093 | 1.3084 | 50.0388 | 16.1003 |
| db4 | 4 | 563.2154 | 1.4411 | 52.2531 | 15.5909 |
| db8 | 5 | 544.6446 | 1.2623 | 52.4808 | 14.4907 |
| db16 | 6 | 547.9617 | 1.1801 | 53.1195 | 12.4438 |
| db32 | 7 | 339.8129 | 1.0014 | 53.1832 | 9.3185 |

# Deliverables

## Plot Wavelet Name vs. Compression Ratio for the Lossless System

Please note: the wavelet enumeration numbers correspond to wavelet names shown in the results table.

## Plot Wavelet name v.s Compression Ratio for Lossy System

Please note: the wavelet enumeration numbers correspond to wavelet names shown in the results table.

## Plot Wavelet name vs. SNR in dBs for Lossy System

I was unable to find the SNR function you used in your example. I found an equation online from [here](https://www.mathworks.com/matlabcentral/answers/71609-how-to-get-snr-for-2-images) to compute SNR that gave analogous results to your example. Decibels were calculated by the following equation: snr\_db = 20\*log10(snr).

|  |  |  |
| --- | --- | --- |
| **Wavelet Name** | **Lossy Compression Ratio** | **Lossy SNR DBs** |
| coif5 | 12.4815 | 21.92533562 |
| db1 | 16.5504 | 24.37616989 |
| db2 | 16.1003 | 24.13667937 |
| db4 | 15.5909 | 23.85742372 |
| db8 | 14.4907 | 23.22178731 |
| db16 | 12.4438 | 21.89906045 |
| db32 | 9.3185 | 19.38692019 |

# Appendix

## Matlab Script

wavelet\_names = {'coif5', 'db1', 'db2', 'db4', 'db8', 'db16', 'db32'}

for wavelet\_index = 1:length(wavelet\_names)

wavelet\_name = wavelet\_names{wavelet\_index}

[h0 h1 f0 f1] = wfilters(wavelet\_name);

% Get the image

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

sz1=size(f);

% Perform Single-level 2D WLT

[LL LH HL HH] = dwt2(f, wavelet\_name);

sz2=size(LL);

% Perform Two Level 2d WLT by transmofring quandrant LL

[LL1 LH1 HL1 HH1] = dwt2(LL, wavelet\_name);

sz3 = size(LL1);

% Reconstruct Lossless Image

LL2 = idwt2(LL1, LH1, HL1, HH1, wavelet\_name, sz2);

lossless\_g = idwt2(LL2, LH, HL, HH, wavelet\_name, sz1);

lossless\_snr = 20\*log(norm(f,'fro')/norm(f-lossless\_g,'fro'))

% Sum the histograms across the 7 output sub images

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;

h\_sum = sum(h);

% Compute the entropy of the lossless image

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

% Compute the Compression Ratio

lossless\_compression\_ratio = 8\*512\*512/(e\*h\_sum)

% Reconstruct the lossy Image

z1 = zeros(sz3);

z2 = zeros(sz2);

LL2 = idwt2(LL1, z1, z1, z1, wavelet\_name, sz2);

lossy\_g = idwt2(LL2, z2, z2, z2, wavelet\_name, sz1);

lossy\_snr = 20\*log(norm(f,'fro')/norm(f-lossy\_g,'fro'))

% Compute the entropy of the lossy image

h1\_sum = sum(h1);

h1 = h1 / h1\_sum; e = 0; l2 = log(2); for i = 1:256 if h1(i) > 0 e = e - h1(i)\*log(h1(i))/L2; end; end;

lossy\_compression\_ratio = 8\*512\*512/(e\*h1\_sum)

end

## MATLAB Output

>> FinalProject

wavelet\_names =

1×7 cell array

{'coif5'} {'db1'} {'db2'} {'db4'} {'db8'} {'db16'} {'db32'}

wavelet\_name =

'coif5'

lossless\_snr =

403.5385

lossless\_compression\_ratio =

1.1798

lossy\_snr =

52.9706

lossy\_compression\_ratio =

12.4815

wavelet\_name =

'db1'

lossless\_snr =

701.1841

lossless\_compression\_ratio =

1.3949

lossy\_snr =

45.9911

lossy\_compression\_ratio =

16.5504

wavelet\_name =

'db2'

lossless\_snr =

582.5093

lossless\_compression\_ratio =

1.3084

lossy\_snr =

50.0388

lossy\_compression\_ratio =

16.1003

wavelet\_name =

'db4'

lossless\_snr =

563.2154

lossless\_compression\_ratio =

1.4411

lossy\_snr =

52.2531

lossy\_compression\_ratio =

15.5909

wavelet\_name =

'db8'

lossless\_snr =

544.6446

lossless\_compression\_ratio =

1.2623

lossy\_snr =

52.4808

lossy\_compression\_ratio =

14.4907

wavelet\_name =

'db16'

lossless\_snr =

547.9617

lossless\_compression\_ratio =

1.1801

lossy\_snr =

53.1195

lossy\_compression\_ratio =

12.4438

wavelet\_name =

'db32'

lossless\_snr =

339.8129

lossless\_compression\_ratio =

1.0014

lossy\_snr =

53.1832

lossy\_compression\_ratio =

9.3185