

Image Compression based on Non-Parametric Sampling in Noisy Environments (Compression using DWT Only)

Authors: 19G01: Kishan Narotam (717 931) & Nitesh Nana (720 064)

Discrete Wavelet Transform

```
clear all;
close all;

% Read in an image file
% We are going to load the image into MATLAB, asking for the name of the
% file. The user input is typed in and converted into a string. When the
% file is read into MATLAB, it's read in as a 3 dimensional matrix. We
% display the image for manual debugging.
fileName = uigetfile('*.');
uploadedImage = imread(fileName);
image = uint8(double(uploadedImage));

% Original image
% Display the uploaded image. The axis do not show, so we set the
% visibility to be on in order to see the pixels
subplot(2,2,1)
imshow(image);
title('Original Image')
axis = gca;
axis.Visible = 'On';
imwrite(image,'originalImage.tif');
imfinfo('originalImage.tif')

% A wavelet decomposition of the image
wname = 'haar';
N = 2;
% wavedec2 is a 2-D wavelet decomposition and returns the wavelet
% decomposition of image using the wavelet type specified by wname and
% the level of decomposition N.
% The wavelet decomposition vector is stored in variable C.
[C,S] = wavedec2(image,N,wname);

alpha = 1.5;
% Birgé-Massart strategy to determine the thresholds for 2-D wavelet
% THR stores the threshold values and NKEEP stores the number of
% coefficients to be kept for compression.
% Typically alpha = 1.5 for compression
[THR,NKEEP] = wdcbm2(C,S,alpha,prod(S(1,:)));

% Compression
% Wavelet compression
% XC stores the compressed version of the data
% PERF0 is the compression ratio achieved in percentage
[XC,CXC,LXC,PERF0,PERFL2] = wdencomp('lvd',C,S,wname,N,THR,'h');
fprintf('Compression Ratio: ');
disp(PERF0)
```

```

% Compressed image
% Display the Compressed image. The axis do not show, so we set the
% visibility to be on in order to see the pixels
subplot(2,2,2)
imshow(XC);
title('Compressed Image')
axis = gca;
axis.Visible = 'On';
imwrite(XC, 'compressedImage.tif');
imfinfo('compressedImage.tif')

% Decompression
% 2-D wavelet reconstruction
% waverec2 performs wavelet reconstruction on the decomposed structure
X = waverec2(C,S, 'haar');
decompressedImage = uint8(X);

% Decompressed image
% Display the Decompressed image. The axis do not show, so we set the
% visibility to be on in order to see the pixels
subplot(2,2,3)
imshow(decompressedImage);
title('Decompressed and Reconstructed Image');
axis = gca;
axis.Visible = 'On';
imwrite(decompressedImage, 'decompressedImage.tif');
imfinfo('decompressedImage.tif')

```