Wavelet Exercise

Digital Processing of Single and Multi-Dimensional Signals 0510-6201
Tel-Aviv University

Submission deadline: 13/06/2019

Write your full name and ID number in the mail body.

You may submit in pairs

In this exercise you will implement a Discrete Wavelet Transform (DWT) filter bank and use it to compress an image. Please read the entire exercise before diving into MATLAB, writing a clear and generic code will help you to reduce the time required to answer all questions. In addition you will be requested to submit parts of your code.

<u>What to submit:</u>

1. PDF file with your answers to all questions.
2. main.m - a script that runs the entire exercise solution and recreates all the results you presented in your PDF file.
3. haar_dwt.m
4. haar_idwt.m
5. remove_coeffs.m
6. haar_image_dwt.m
7. haar_image_idwt.m

Code will be automatically tested so make sure it runs without any errors. If it fails to run the grade on it will be 0. Document your code, failing to do so will reduce your grade.

Wavelets in 1-D:

Note: Squared error is defined as $SE(x,y) = \sum_{i=1}^{N}(x_i - y_i)^2$ for $,y \in \mathbb{R}^{N\times 1}$.

The Haar Wavelets are characterized by:

$$\mathrm{g} = \frac{1}{\sqrt{2}}[1,-1]$$

$$\mathrm{h} = \frac{1}{\sqrt{2}}[1,1]$$

To compute the wavelet transform of a signal f of length N, one needs to successively compute

$(\mathrm{f} * \bar{\mathrm{h}}) \downarrow 2$ or $(\mathrm{f} * \bar{\mathrm{g}}) \downarrow 2$, where $*$ denotes circular convolution (periodic boundary conditions) and

$\bar{x}[n] = \ ^*[-n]$.

A signal f is given as:

$$f = [0;\ 1;\ 2;\ 1;\ 0; -1; -2;\ 1;\ 1;\ 5;\ 5;\ 3;\ 2;\ 3;\ 3;\ 3]$$

1. Plot the original signal as a stem plot.
2. What is the signal length for the approximation and the details at level three, at level two and level one?
3. Compute (by hand) the Discrete Wavelet Transform of f using Haar Wavelets up to level 3. Do this by drawing the binary tree of operations together with all intermediate results. You may combine the successive convolution and downsampling steps into one calculation. Carry the $\sqrt{2}$ through your computations symbolically.
4. Implement (your own) Haar DWT MATLAB function as following:

```
function [ approx,details ] = haar_dwt( signal,levels )

Inputs:
      signal - input signal for analysis.
      levels - analysis depth.
Outputs:
      approx - approximation of the signal at level 0.
      details - cell array with levels cells. Cell j should contain the
                details of level j-1;
```

5. Use your function to calculate the DWT of f and compare your results to the ones from question (3). Plot your results using the stem function and explain.

We will implement the Inverse Discrete Wavelet Transform (IDWT). The algorithm works as follows: select the two leaf nodes in the tree which have greatest tree-depth. Call these leaves a and b, where a has been obtained by applying $\bar{g}$ and b has been obtained by applying $\bar{h}$. Replace the common ancestor of a and b with $c = (a \uparrow 2) * g + (b \uparrow 2) * h$. Repeat until the tree has only a single node left.

6. Implement:

```
function [ signal ] = haar_idwt( approx,details )

        Inputs:
              approx - approximation of the signal at level 0.
              details - cell array with levels cells. Cell j contains the
                        details of level j-1;
        Outputs:
              signal - reconstructed signal using Haar wavelets.
```

7. Use your `haar_dwt` and `haar_idwt` implementations to analyze and synthesize f and make sure you have a perfect reconstruction.
8. Write a function that sets **details** coefficients with an **absolute** value <= thresh to zero:

```
function [ th_details,zero_count ] = remove_coeffs( details,thresh )

        Inputs:
              thresh - threshold to be used for zeroing coefficients.
              details - cell array.
        Outputs:
              th_details - cell array of the same size as details. Each
                           cell contains the values in the corresponding
                           cell in details only thresholded by thresh.

              zero_count - total number of values that were zeroed by the
                           thresholding operation.
```

9. Remove all coefficients that have an absolute value of less than 0.8 and reconstruct the signal from the remaining coefficients. Plot the reconstructed signal. How good is the reconstruction? What is the Squared Error of the reconstruction? How many coefficients were set to 0?
10. Plot a graph of the Squared Error of the reconstruction and the number of coefficients set to zero against the threshold value in the range $[0,10]$.
11. What is the maximal threshold for which the Squared reconstruction error is below (or equal to) 2.25? How many coefficients were set to zero at this threshold?
12. How can this be used to compress the signal?

Wavelets in 2-D:

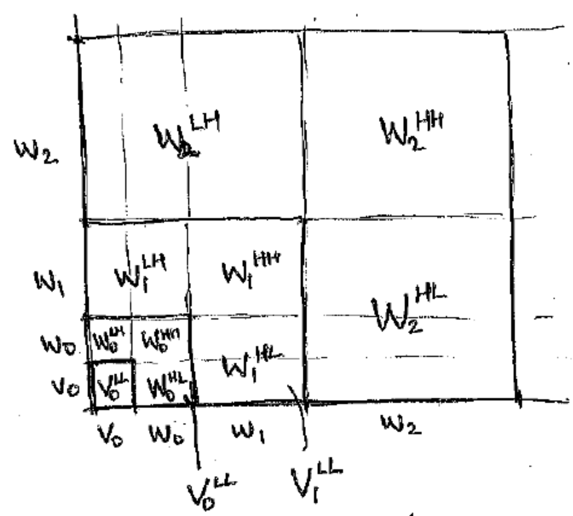
Using the 1D DWT you implemented in the previous section you will now implement an image compression method.

Assuming the Haar DWT is separable one can calculate the DWT of an image by first applying a 1D DWT on the image rows and then applying the same 1D DWT on the columns of the result.

13. Implement a 2D Harr analysis function that accepts an image and the number of wavelet levels and returns the resulting wavelet coefficients.

```
function [ wt ] = haar_image_dwt( image,level )

            Inputs:
                  image - input image. (Assume the image is square and
                          it's size os a power of 2).
                  level - number of wavelet levels for analysis.
            Outputs:
                  wt - wavelet transform of the input image. wt is the
                       same size as image and should be in the
                       following format (you may flip the DWT on the y
                       axis):
```



14. Display your 2D DWT with 3 levels of **lena.png**. Explain your results.
15. Implement the inverse 2D Haar Wavelet synthesis function.

```
function [ synth ] = haar_image_idwt( wt,level )
   Inputs:
         wt - input 2D DWT of an image. (Assume the wt is square and
                 it's size os a power of 2).
         level - number of wavelet levels for analysis.
   Outputs:
         synth - the synthesized image.
```

16. Make sure analyzing and synthesizing the image using your functions results in a perfect reconstruction.

Normalize the input image to the range [0,255]

17. One can compress an image by zeroing out DWT coefficients that have an absolute value below a threshold T. Plot graphs of the compression ratio and the Square Error (of the synthesis) as a function of the threshold T and explain your results. (We define the compression ratio as the percentage of **details** coefficients that were zeroed compared out of the **total** number of coefficients).
18. Show 3 interesting examples of reconstructions of Lena's image at different thresholds and explain.
19. Assuming a user wishes to compress an image and is willing accept an **MSE** of 20 (gray levels)^2 what is the compression ratio this method can offer him?
20. Can you suggest a better method for compression of images using wavelets?