EE7150 Computer Project V
Wavelets & Multiresolution (Chapter 6)
Due Date: Oct 18, 2021

Note: Problems 2 and 3 describe the steps for building up 1-D and 2-D Haar Wavelet transform based decomposition from scratch. These steps (optional) are given for those students who wish to develop their own codes (not mandatory). *Instead, students may use matlab's built-in wavelet toolbox for the same purpose*. The relevant wavelet routines in matlab for 1D and 2D wavelet decomposition and reconstruction are,

Multilevel: 'wavedec', 'waverec', 'wavedec2', 'waverec2'

Single level: 'dwt', 'idwt', 'dwt2', 'idwt2'

Note also that in matlab, the "db1" wavelet is the same as Haar wavelet. More information about the usage of these matlab functions can be obtained using matlab 'doc' command. Further details and usage information on the haar wavelet can be found using,

```
>> waveinfo('haar')
```

1. Solve Problem 6.34 from the Textbook.

34 Show that scaling function

$$\varphi(x) = \begin{cases} 1 & 0.25 \le x < 0.75 \\ 0 & \text{elsewhere} \end{cases}$$

does not satisfy the second requirement of a multiresolution analysis.

2. **One-Dimensional Discrete Wavelet Transforms (Optional)**

The purpose of this project is to build a rudimentary wavelet transform package using Haar wavelets that can be used in projects that follow. You will use an "averaging and differencing" approach that is unique to Haar basis functions. As an introduction to the method, consider the function in Example 7.8. The necessary "averaging and differencing" operations are:

Step 1: Compute two-point sums and differences across the function vector and divide the results by the square root of 2. Since f(x) = {1, 4, -3, 0}, we get

{1 + 4, -3 + 0, 1 - 4, -3 - 0} / 1.414

{5, -3, -3, -3} / 1.414

Note that the sums are positioned consecutively at the beginning of the intermediate result and followed by the corresponding differences.

Step 2: Repeat the process over the sums computed in the first step to get

{[5 + (-3)] / 1.414, [5 - (-3)] / 1.414, -3, -3} / 1.414

{2 / 1.414, 8 / 1.414, -3, -3} / 1.414

{1, 4, -2.121, -2.121}

The coefficients of the final vector match those in Example 6.19 (Example 7.8 of 3rd Edition). The two-step computation generates a two-scale DWT with respect to Haar wavelets. It can be generalized to higher scales and functions with more than 4 points. Moreover, an inverse DWT can be computed by reversing the process.

a) Write a program to compute *j*-scale DWTs with respect to Haar wavelets. Let scale be an input parameter and assume a 2*M* point discrete one-dimensional function. Use the averaging and differencing approach described above.
b) Write a program to compute the inverse DWT of a *j*-scale DWT based on Haar wavelets.
c) Test your programs using the function in Example 6.19 (Example 7.8 of 3rd Edition).

3. **Two-dimensional Discrete Wavelet Transforms**
   a) Use the routines developed in Part-1 to write a program that computes *j*-scale two-dimensional DWTs with Haar wavelets. Base your routine on the discussion of separable wavelets and two-dimensional wavelet transforms in Section 6.10 (Section 7.5 of 3rd Edition).
   b) Download the image in Fig. 6.30(a) [Fig. 7.1of 3rd Edition] from the course Pilot and use it in your program to generate the one-scale and two-scale DWT (or Haar transform) results shown in Fig. 6.30 (b) and 6.30 (c) (Fig. 7.10 (a) of 3rd Edition). In addition, generate the three-scale DWT result also. Label the various detail and approximation coefficients that make up the transform and indicate their scales.
   c) Write a program to compute the inverse two-dimensional DWT with respect to Haar wavelets and use it to reconstruct the original image from the wavelet decomposition in (b).
   d) Write a program to scale the detail coefficients of the DWT in (b) so that the underlying structure is more visible. Use scaling described in page-75 of 4th Ed (pages 79-80 of 3rd Edition). The approximation coefficients do not need to be scaled.

4. **Wavelet Transform Modifications**

Download the image Fig. 4.40(a) from Pilot, reduce its size in half by row-column deletion (decimation)~~, and pad it with 0s~~ to obtain a 512 x 512 array. [Note for students with 3rd Edition: Fig. 4.41(a) has been replaced with Fig. 4.40(a) in 4th Ed.]. Use the two-dimensional DWT program developed in Part-2 to compute the transform of the padded image at a variety of scales between 1 and 9.

a) Zero the approximation coefficients of the generated transforms and record your observations regarding subsequently reconstructed images. That is, reconstruct by computing the inverse transforms of the decompositions after the approximation coefficients have been zeroed and record the impact on the transform modifications.
b) Repeat the process in (a) but zero the horizontal detail coefficients instead.
c) Repeat the process in (a) but zero the vertical detail coefficients instead.
d) Repeat the process in (a) but zero both the horizontal and vertical detail coefficients.

Project Report Guidelines

Page 1. Cover Page. Typed or printed neatly.

Project title
Project number
Course number
Student's name
Date due
Date handed in

Abstract (not to exceed 1/2 page) **Page 2.** Technical discussion. One to two pages (max). This section should include the techniques used and the principal equations (if any) implemented.
Page 3 (or 4). Discussion of results. One to two pages (max). A discussion of results should include major findings in terms of the project objectives, and make clear reference to any images generated.

Results. Includes all the images generated in the project. Number images individually so they can be referenced in the preceding discussions. Include titles, labels and legends in the figures, as appropriate.

Figures: Number the Figures sequentially as Figure 1, Figure 2, etc. and add captions/titles/axis markers to the figures, as appropriate.

Appendix. Program listings. Includes listings of all programs written by the student. Standard routines and other material obtained from other sources should be acknowledged by name, but their listings need not be included.

A note on program implementation: The objective of the computer programs used in the projects is to teach the student how to manipulate images. There are numerous packages that perform some of the functions required to implement the projects. However, the use of "canned" routines as the only method to implement an entire project is discouraged. For example, if a student is using MATLAB and the Image Processing Toolbox, a balanced approach is to use MATLAB's programming environment to write M functions to implement the projects, using some of MATLAB's own functions in the process. A good example is the implementation of the 2-D Fourier Fast Transform. The student should use the MATLAB function that computes the 2-D FFT directly, but write functions for operations such as centering the transform, multiplying it by a filter function, and obtaining the spectrum.