Prof. A. Zakhor   Spring 2019
EE225b – Digital Image Processing
Homework 14 – Wavelet Transforms

Overview:

In this assignment, you will implement the digital wavelet transform (DWT) using Haar-based wavelets and reconstruct the image by using partial wavelet information.

Assignment specifics:

1.   One-Dimensional Discrete Wavelet Transforms

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 6.19. 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. 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 2M 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.

2. Two-dimensional Discrete Wavelet Transforms

(a) Use the routines developed in 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.

(b) Download the image in Figure 2.2 and use your program to generate the **three**-scale DWT (or Haar transform) shown in Fig. 6.30. 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. The approximation coefficients do not need to be scaled.

3. Wavelet Transform Modifications

Download the image from Figure 2.3 from the course web site, reduce its size in half by row-column deletion, and pad it with 0s to obtain a 512×512 array. Use the two-dimensional DWT program developed in 2 to compute the transform of the padded image at a variety of scales between 1 and 8.

(a) Zero the approximation coefficients of the generated transforms and record your observations regarding subsequently reconstructed images. That is, compute 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.

4. Wavelet Coding

Download and compress the image of Figure 2.4 using the Haar-based DWT program. Use various scales while truncating the detail coefficients to achieve compression. Quantify the reconstruction error using the program from Homework12 Part I.

Note:

1. For each problem, you need to:

    a. [1] Email your source code (zip it) to ee225bsp19@gmail.com. If the question asks for any implementations.

        i.   Make sure it is executable because I need to run your code to give you a score. Either MATLAB or Python is okay.

        ii. Email title: FirstName_LastName_HW#.

b. [2] Submit a single PDF file (not word or other formats) on Gradescope which contains:

    i. your answer for each problem;

    ii. your source code;

    iii. your output image.

Make sure to prepare your solution to each problem on a separate page. On Gradescope, please select and match each page to the corresponding problems. Please do make sure that you have BOTH your output images AND codes included in the Gradescope submission AND email me your executable codes, otherwise you will lose credits.

Here are some helpful Matlab commands:

`I = uint8(x)`	`I` is a matrix of integers ranging from 0..255
`imshow(I)`	Displays `I` as a grayscale image in the current figure
`I = imread('small.bmp', 'bmp')`	Reads the image file small.bmp and stores it in matrix `I`
`imwrite(I, 'result.bmp', 'bmp')`	Writes the matrix I to the image file result.bmp
`B = conv2(h1, h2, A, mode)`	Performs 2-D convolution on the matrix `A` with impulse responses `h1` along the rows and `h2` along the columns.
`B = padarray(A, m, n)`	Zero-pads the matrix `A` with `m` rows and `n` columns of zeroes