 **Project Title:  Image Processing Fundamentals**

**Project Number**: Project 5

**Course Number**: CEG 7850-01

**Student's Name**: Alex Reigle

**Date Due**: October 18, 2021

**Date Submitted**:  October 18, 2021

**Declaration Statement:**

I hereby declare that this Report and the Matlab codes were written/prepared entirely by me based on my own work, and I have not used any material from another Project at another department/ university/college anywhere else, including Wright State. I also declare that I did not seek or receive assistance from any other person and I did not help any other person to prepare their reports or code.  The report mentions explicitly all sources of information in the reference list. I am aware of the fact that violation of these clauses is regarded as cheating and can result in invalidation of the paper with zero grade. Cheating or attempted cheating or assistance in cheating is reportable to the appropriate authority and may result in the expulsion of the student, in accordance with the University and College Policies.

**Abstract:**

In this project I show that a for a given function, the second requirement of multiresolution analysis is not satisfied. I have also developed a one and two-dimensional Haar wavelet transform function (haarWavelet1D and haarWavelet2D). These functions allowed me to examine the similarities and differences across varying scales of wavelet decomposition and reconstruction. In 1D I compared the J-scale discrete wavelet transform (DWT) for J = [1,2,3,4] to verify that while the DWT results differ, the inverse discrete wavelet transform (iDWT) remained constant for the given function. In 2D I computed the J-scale DWT and iDWT as well as the approximate, horizontal, vertical, and diagonal coefficient matrices to recreate results of figure 7.30 from the textbook. I also observed the effects of removing various combinations of coefficient matrices on the reconstruction of the initial image.

**Technical Discussion:**

The first problem of the project uses the definition of nested scales defined as

V-ꚙ С … С V-1 С V0 С V1 С V2 С … С Vꚙ (7-123)

to show that for the piecewise scaling function (defined below) does not satisfy the second fundamental requirement of multiresolution analysis.

A picture containing text

Description automatically generated

Additionally, the definition of a scaling function, with father scaling function ϕ(x) was used. This definition is as follows:

Φj,k(x) = 2j/2 ϕ(2jx-k) (7-121)

The project code makes use of the MatLab built in 1D and 2D multilevel wavelet transform (wavedec and wavedec2) as well as the corresponding built in inverse wavelet transform functions (waverec and waverec2). Additionally, the MatLab built in 2D single level inverse wavelet transform function (idwt2) is used.

**Discussion and Results:**

Problem 1 was solved visually by showing that the function ϕ0,0(x), defined above, is not nested in the complete function space spanned by the next higher scale, J = 1. This is shown in Fig. 1, below.

Diagram

Description automatically generated

Figure 1: Row 1 shows the complete function space for the defined ϕ(x) with J=0, denoted as V0. Row 2 shows the next higher function space for the defined ϕ(x) with J=1, denoted a V1. Row 3 depicts the superposition of V0 and V1 to show that, since ϕ1,0(x) and ϕ1,1(x) do not share a boundary, function space for V0 is not nested within V1.

Furthermore, Fig. 1 shows that the two defined function spaces, V0 and V1, do not satisfy the nesting condition of (eqn. 7-123) in the text. Otherwise stated as

Diagram

Description automatically generated (1)

Thus confirming that the function does not satisfy the second fundamental requirement of a multiresolution analysis.

Problem 2 tasked me with computing the J-scale DWT and iDWT for the function

f(x) = {1, 4, -3, 0} (2)

to recreate the results of example 7.19 (on page 512, 4th ed.) from the textbook. I do this using the haarWavelet1D(f,J) function I developed, where f is the input function (2) and J is the scale. Fig. 2, below, shows that the results are as expected.

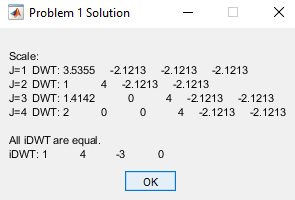


Figure 2: The 1D DWT and iDWT results when preformed on (eqn. 2) with a J-scale of J = [1,2,3,4]. As stated in the figure, the iDWT of all J-scales were equivalent.

In this case, the iDWT for each scale tested results in the exact array input to the DWT. This confirms that the wavelet transform process was done correctly for all scales.

Problem 3 asked for me to implement a 2D multilevel Haar DWT and iDWT function which returns the approximate, verticle, horizontal, and diagonal coefficient matrices of the J-scale DWT and computes the iDWT. This is done using the haarWavelet2D(image, J) function I wrote, where image is the input matrix and J is the scale desired. The scale of J = [1,2,3] was used in this problem to recreate and expand upon the results shown in Figure 7.30 (on page 523) from the textbook.

Graphical user interface

Description automatically generated with medium confidence

Figure 3: **(a)** The original image. **(b)** The J=1 scale 2D Haar DWT. Upper-left is the approximation coefficient matrix. Upper-right is the horizontal coefficient matrix. Lower-left is the vertical coefficient matrix. Lower-right is the diagonal coefficient matrix. **(c)** The J=2 scale 2D Haar DWT with the same approximation/horizontal/vertical/diagonal configuration in the place of the approximation matrix from (b). **(d)** The J=3 scale 2D Haar DWT with the same approximation/horizontal/vertical/diagonal configuration in the place of the approximation matrix from (c). The configuration implemented is done so to mirror the textbook results.

After computation, the results of the J-scale DWT were all scaled using reduced intensity resolution. This affect is done by a combination of scaling using scalePixelValues and intensity level reduction using the quantize function. This process amplifies the underlying structures (most notable in the horizontal and vertical coefficient matrices) making them more visible.

Problem 4 applies a very similar process to the problem 3. After using row and column decimation to reduce the size of the image Figure 4.40(a) from the text, I used the same haarWavelet2D function utalized from problem 3 to compute the approximate, horizontal, vertical, and diagonal coefficient matrices for all scale values from one to nine. I then proceeded to multiply the approximate, horizontal, vertical, and both horizontal and vertical coefficient matrices by zero (respectively) before computing the iDWT of each. This was done to show the effect of each component at various scales on the wavelet reconstruction process.

Qr code

Description automatically generated

Figure 4: The reconstruction on image Fig 4.40a, from the text, after DWT with the approximate coefficient matrix was replaced with zeros for J-scale of 1-9.

Qr code

Description automatically generated

Figure 5: The reconstruction on image Fig 4.40a, from the text, after DWT with the horizontal coefficient matrix was replaced with zeros for J-scale of 1-9.

A picture containing qr code

Description automatically generated

Figure 6: The reconstruction on image Fig 4.40a, from the text, after DWT with the vertical coefficient matrix was replaced with zeros for J-scale of 1-9.

A picture containing qr code

Description automatically generated

Figure 7: The reconstruction on image Fig 4.40a, from the text, after DWT with both the horizontal and vertical coefficient matrices replaced with zeros for J-scale of 1-9.

As expected, replacing the approximation coefficient matrix resulted in the greatest destruction to the image reconstruction, as shown in Fig. 4. The aprroximation coefficient matrix was nearly enough to reconstruct the image without the vertical, horizontal or either coefficient matrices, as shown in Fig. 5-7. However, at higher scales the loss of any information seems to greatly reduce the ability to reconstruct an image.

This is likely due to the use of the built-in MatLab function wavedec2 and waverec2, which are fast wavelet transform functions that only further decompose the approximation coefficient matrix at each higher scale and return only the highest scale approximate, horizontal, vertical, and diagonal coefficient matrices. This would result in larger amounts of data being excluded from the iDWT and at each J+1th level only as much detail is available for reconstruction as the Jth approximation coefficient matrix. This effect is well illistrated below.

A picture containing text, items, different

Description automatically generated

Figure 8: This is the original image used for processing in Problem 3 as well as the corresponding isolated approximate coefficient matrix at each scale.

From Fig. 8, it is obvious that with each DWT of increasing scale, the approximation matrix is of worsening resolution. From this observation and the knowledge of the wavedec2 and waverec2 functions, we can conclude that any high level DWT would need to be implemented using a method that could capture more information than the highest J-level scaling and wavelet functions.

**Figures:**

Diagram

Description automatically generated

Figure 1: Row 1 shows the complete function space for the defined ϕ(x) with J=0, denoted as V0. Row 2 shows the next higher function space for the defined ϕ(x) with J=1, denoted a V1. Row 3 depicts the superposition of V0 and V1 to show that, since ϕ1,0(x) and ϕ1,1(x) do not share a boundary, function space for V0 is not nested within V1.

Text

Description automatically generated

Figure 2: The 1D DWT and iDWT results when preformed on (eqn. 2) with a J-scale of J = [1,2,3,4]. As stated in the figure, the iDWT of all J-scales were equivalent.

Graphical user interface

Description automatically generated with medium confidence

Figure 3: **(a)** The original image. **(b)** The J=1 scale 2D Haar DWT. Upper-left is the approximation coefficient matrix. Upper-right is the horizontal coefficient matrix. Lower-left is the vertical coefficient matrix. Lower-right is the diagonal coefficient matrix. **(c)** The J=2 scale 2D Haar DWT with the same approximation/horizontal/vertical/diagonal configuration in the place of the approximation matrix from (b). **(d)** The J=3 scale 2D Haar DWT with the same approximation/horizontal/vertical/diagonal configuration in the place of the approximation matrix from (c). The configuration implemented is done so to mirror the textbook results.

Qr code

Description automatically generated

Figure 4: The reconstruction on image Fig 4.40a, from the text, after DWT with the approximate coefficient matrix was replaced with zeros for J-scale of 1-9.

Qr code

Description automatically generated

Figure 5: The reconstruction on image Fig 4.40a, from the text, after DWT with the horizontal coefficient matrix was replaced with zeros for J-scale of 1-9.

A picture containing qr code

Description automatically generated

Figure 6: The reconstruction on image Fig 4.40a, from the text, after DWT with the vertical coefficient matrix was replaced with zeros for J-scale of 1-9.

A picture containing qr code

Description automatically generated

Figure 7: The reconstruction on image Fig 4.40a, from the text, after DWT with both the horizontal and vertical coefficient matrices replaced with zeros for J-scale of 1-9.

A picture containing text, items, different

Description automatically generated

Figure 8: This is the original image used for processing in Problem 3 as well as the corresponding isolated approximate coefficient matrix at each scale.

**Appendix:**

**Written Programs:**

* main
* haarWavelet1D
* haarWavelet2D
* scalePixelValues
* quantize

**Utilized Programs:**

* waverec2
* wavedec2
* idwt2
* subplot
* imsho
* cast
* waverec
* wavedec
* appceof2
* max
* min
* abs
* zeros
* size
* num2str
* msgbox
* title
* imread
* pwd
* sqrt