Report on
Image Processing and Interpretation
Project 5

Submitted to
Dr. Bebis
for
CS474

December 9th, 2015

By:
Aaron Whitehead
and
Jeremiah Berns

This report examines several introductory computer processing practice programs. The programs cover such things as 2D Haar wavelet transformations, and Haar wavelet partial coefficient reconstructions.

Technical Discussion

As the overall structure of the project is broken up into three sections, the technical writeup will also be broken into three sections.

Problem one:

The first problem in this project is to take a image reduce it to its two dimensional Haar wavelet representation. This is done by performing the following algorithm on every row:

1. Take the row of data and break it up into pairs of two data points each.
2. Average each pair, with the average being on the left and a new data point on the right, which is the difference between the two original points and the found average.
3. Repeat this until all pairs have been done, which leaves you with N/2 averages and N/2 details (the differences from the average)
4. Recursively do this on the averages until you are left with 1 single average point, and N-1 details next to it.

Once this has all been done, you are left with a single column that has all averages, which you will treat the same as a row from the algorithm above. This leaves one single average point in the top left hand corner and all other points being details.

Program two:

The second problem in this project is a simple inverse action from part one. This is done by the following algorithm:

1. Take the first column and by taking the average point and the first detail, create two average points.
2. Repeat this system until the entire first column is now just averages.
3. Next, perform this system on each row, by taking the first average in each row, and expanding it using the details on the right repeatedly until all of the points are averages and there are no details left.

This system allows the user to convert the image from 2D Haar wavelet system back to a visually appealing pgm system.

Program three:

The third problem is to take in a 2D Haar wavelet representation and remove all but the top X data points. This is also done using a discrete Fourier transform in order to compare the differences. The algorithm for this problem is:

1. Take all of the data points from the Haar 2D representation and load them into a vector with its data and coordinates.
2. Blank out the image by setting all data points (except the single average at 0,0) to 0.
3. Sort the data points stored in the vector by comparing the absolute values of each point.
4. Loop over the top x points of the vector and place each of the found data points in its proper coordinate location.

After doing all of this, it was then necessary to do the same thing for DFT. The algorithm is essentially the same, except the function used to sort the vector is different, as instead of using the absolute value of the point, it instead uses the magnitude of the complex pair.

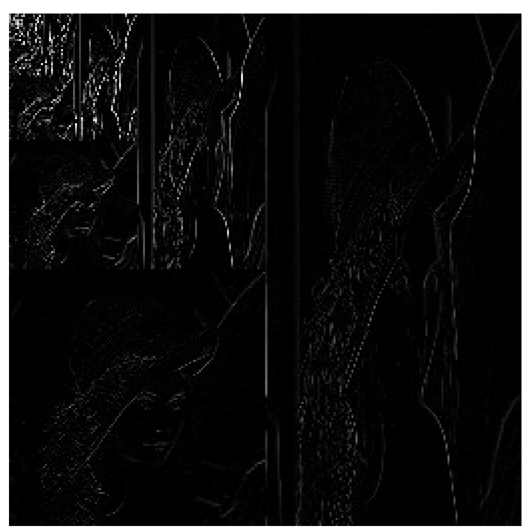
Results and Discussion

Once again, the results section, like the technical discussion section, will be broken into three core parts.

Problem one:

Problem one was to convert the image into a 2D Haar wavelet representation. In order to visualize the changes that were made, the program prints out the converted image for visual comparison reasons. There is not too much to actually say about this, below are several examples.

Given Lenna



2D Haar Lenna

Given Boat

2D Boat Haar

As can be seen, the Haar representation is both dark and skewed. The image may not be easily seen once printed so please check the pdf of this document found at www.github.com/bigmacstorm/imageProcessing in the projFive folder. The Haar image contains both very large and very small numbers, along with matching negative values, meaning that it is somewhat hard to visually represent.

Problem two:

Problem two is a continuation of problem one, in that it is to convert the Haar image back to the original, along with calculating the Mean Square Error. Below are examples of converting Haar images back to regular images.

Lenna Haar

Lenna Reconstructed

Haar Boat

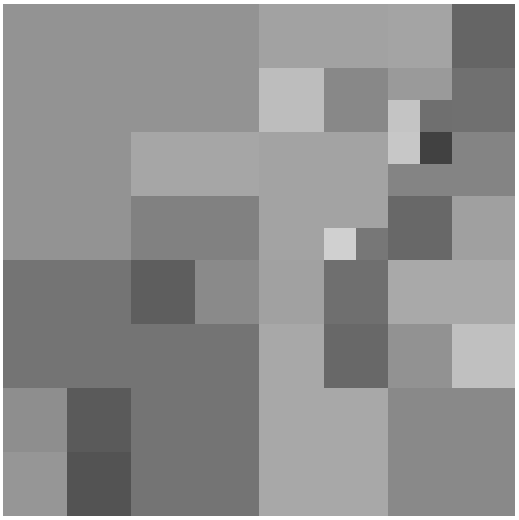
Reconstructed Boat

As can be seen, the reconstruction works very well. As far as the eye can tell, the given and reconstructed image are identical. Further more the Mean Square Error also verifies this. The MSE for the Lenna image was 0.975, which is very good as that is cumulative across the entire image. The MSE for the boat reconstructed image is 0.997 which is once again, super low.

Problem three:

Problem three is where most of the meat comes from in this project, as the results are left to be analyzed and compared with different strategies. Below is a series of given images, the image reconstructed from Haar representation with all but the top x removed, and then the reconstructed image from the DFT with all but the top x complex pairs removed.

Given Lenna

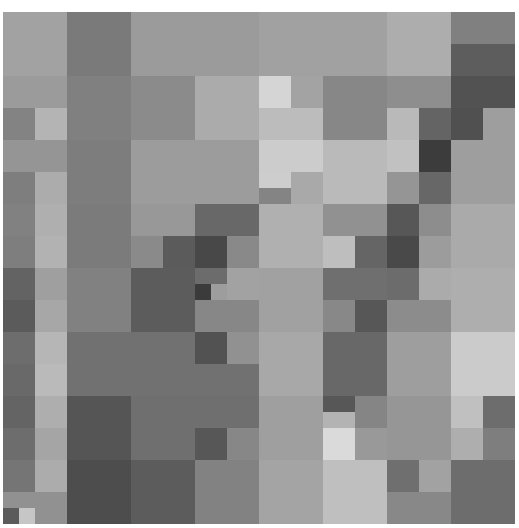


Lenna Haar x=20

Lenna DFT x=20

Lenna Haar x=40

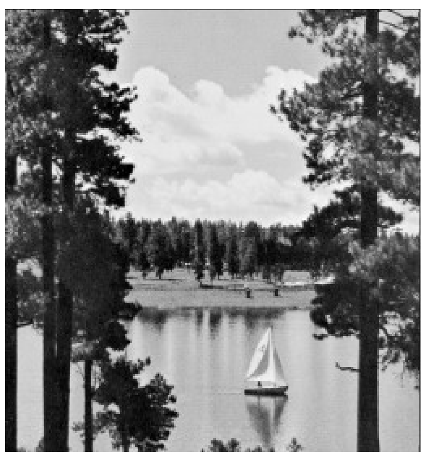
Lenna DFT x=40



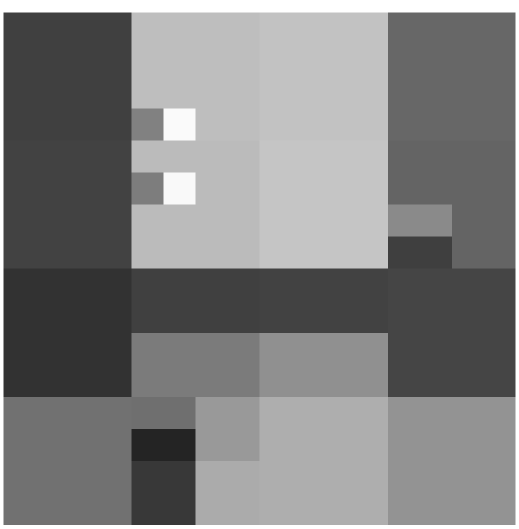
Lenna Haar x=80

Lenna DFT x=80

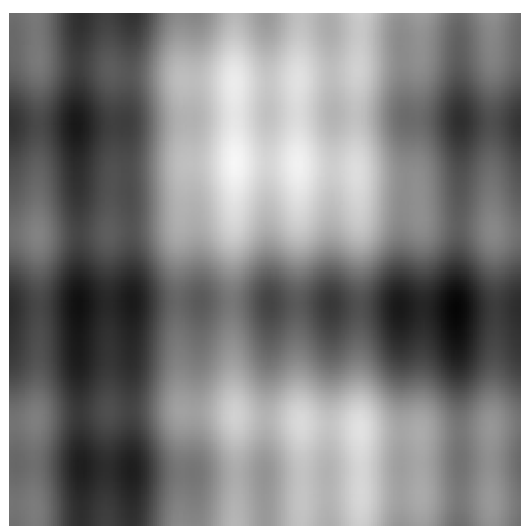
As can be seen by analyzing these images, it is apparent that the DFT and the Haar reconstructions are quite different. The first thing that jumps out is that the Haar reconstruction results in a "blocky" image, which can be understood to be because sections are being averaged together essentially, and now must be stretched across the gaps. The DFT meanwhile is quite blurry, with only very vague correlation to the starting image. As the number of points increases on the Haar reconstruction, the number of blocky sections increases. By the time the program uses 80 data points, it can even be seen to look similar to the given image if one were to squint their eyes. This can also be seen in the MSE of the images, which were 1297.52, 1085.08, and 854.83 respectively, definitely showing a trend moving toward the optimal. As the DFT samples increase, the hazy lines start to take more shape, adding more curves or move color changes between lines, however it is still quite a bit different from the optimal and I personally cannot see much more than a very vague correlation between dark and bright areas. Below, in order to give another example of these things, is the boat image set.



Boat Given

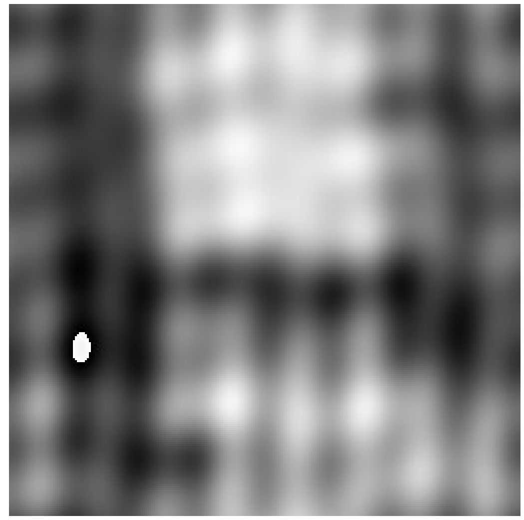


Boat Haar x=20



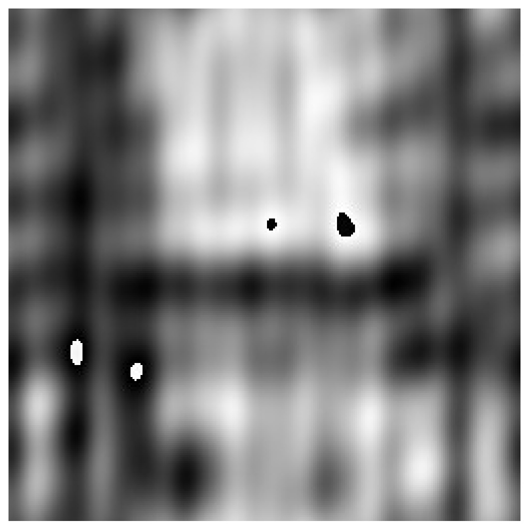
Boat DFT x=20

Boat Haar x=40



Boat DFT x=40

Boat Haar x=80



Boat DFT x=80

Once again, it can be seen that the images slowly improve and approach the optimal as the number of data points left in the image grows.

Division of Labor

It is slightly difficult to say exactly who did what, it can be said that both Aaron and Jeremiah worked equally on the write up, while Aaron took a lead on problem three and Jeremiah took the lead on problems one and two. Using GitHub and voice chat while working means that the labor was very evenly split. If you are interested, you can find the code on my github: http://github.com/BigMacStorm/imageProcessing