



# Image Compression

Greg Ames

Dec 07, 2002

## Abstract

Digital images require large amounts of memory to store and, when retrieved from the internet, can take a considerable amount of time to download. The Haar wavelet transform provides a method of compressing image data so that it takes up less memory. This paper will discuss the implementation of the Haar wavelet transform and some of its applications.

## 1. Introduction

In recent years there has been an astronomical increase in the usage of computers for a variety of tasks. With the advent of digital cameras, one of the most common uses has been the storage, manipulation, and transfer of digital images. The files that comprise these images, however, can be quite large and can quickly take up precious memory space on the computer's hard drive. A gray scale image that is 256 x 256 pixels has 65,536 elements to store, and a a typical 640 x 480 color image has nearly a million! The size of these files can also make downloading from the internet a lengthy process. The Haar wavelet transform provides a means by which we can compress the image so

[Introduction](#)

[How Images are Stored](#)

[How the Wavelet...](#)

[The Power of Linear...](#)

[Approximation and...](#)

[Applications of...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



[Page 1 of 27](#)

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet...](#)

[The Power of Linear...](#)

[Approximation and...](#)

[Applications of...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



[Page 2 of 27](#)

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

that it takes up much less storage space, and therefore electronically transmits faster and in progressive levels of detail.

## 2. How Images are Stored

Before we can understand how to manipulate an image, it is important to understand exactly how the computer stores the image. For a 256 x 256 pixel gray scale image, the image is stored as a 256 x 256 matrix, with each element of the matrix being a number ranging from zero (for black) to some positive whole number (for white). We can use this matrix, and some linear algebra to maximize compression while maintaining a suitable level of detail.

## 3. How the Wavelet Transform Works

In order to transform a matrix representing an image using the Haar wavelet transform, we will first discuss the method of transforming vectors called averaging and differencing. First, we start out with an arbitrary vector representing one row of an 8 x 8 image matrix:

$$\mathbf{y} = ( 448 \quad 768 \quad 704 \quad 640 \quad 1280 \quad 1408 \quad 1600 \quad 1600 )$$

Because the data string has length  $8 = 2^3$  there will be three steps to the transform process. If the string were  $2^k$  long, there would be  $k$  steps in the process. For the first step our data string becomes:

$$\mathbf{y}_1 = ( 608 \quad 672 \quad 1344 \quad 1600 \quad -160 \quad 32 \quad -64 \quad 0 )$$

We get this by first thinking of our original data as being four pairs of numbers (448 & 768, 704 & 640, etc...). We average each of these pairs, and the results become the

[Introduction](#)[How Images are Stored](#)[How the Wavelet ...](#)[The Power of Linear ...](#)[Approximation and ...](#)[Applications of ...](#)[Examples](#)[Analysis](#)[The Compressomatic](#)[Home Page](#)[Title Page](#)[Page 3 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

first four entries of our modified string  $\mathbf{y}_1$ . These numbers are known as *approximation coefficients*. Next we subtract these averages from the first member of each pair. These answers become the last four entries of  $\mathbf{y}_1$  and are known as *detail coefficients*, which are shown in bold. The detail coefficients are repeated in each subsequent transformation of this data string. The visual interpretation of approximation and detail coefficients will be discussed in Section 5. For now we will proceed with the second step which changes our data string to:

$$\mathbf{y}_2 = ( 640 \quad 1472 \quad -\mathbf{32} \quad -\mathbf{128} \quad -\mathbf{160} \quad \mathbf{32} \quad -\mathbf{64} \quad 0 )$$

We get this by treating approximation coefficients from the first step as pairs (608 & 672, 1344 & 1600). Once again we average the pairs and the results become the first two entries of  $\mathbf{y}_2$ , and are our new approximation coefficients. We then subtract the averages from the first element of each pair. The results become the third and fourth elements of  $\mathbf{y}_2$  (These are also detail coefficients and are repeated in the next step). The last four elements in  $\mathbf{y}_2$  are identical to  $\mathbf{y}_1$ . For the last step our data string becomes:

$$\mathbf{y}_3 = ( 1056 \quad -\mathbf{416} \quad -\mathbf{32} \quad -\mathbf{128} \quad -\mathbf{160} \quad \mathbf{32} \quad -\mathbf{64} \quad 0 )$$

This time we obtain the first entry of  $\mathbf{y}_3$  by averaging the two approximation coefficients of  $\mathbf{y}_2$  (640 & 1472). We obtain the second element by subtracting the average from the first element of the pair. This is the final detail coefficient and is followed by the detail coefficients from  $\mathbf{y}_2$ .

The Haar wavelet does this transformation to each row of the image matrix, and then again to every column in the matrix. The resulting matrix is known as the Haar wavelet transform of the original matrix. It is important to note at this point that this process is entirely reversible. It is this fact that makes it possible to retrieve the original image from the Haar wavelet transform of the matrix.



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 4 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## 4. The Power of Linear Algebra

The averaging and differencing method that we just discussed is very effective, but, as you can imagine, the calculations can quickly become quite tedious. We can, however, use matrix multiplication to do the grunt work for us. In our previous example, we can describe the transformation of  $\mathbf{y}$  to  $\mathbf{y}_1$  as:

$$\mathbf{y}_1 = \mathbf{y} A_1$$

Where  $A_1$  is the matrix

$$A_1 = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 \end{pmatrix}$$

It can next be shown that the transformation from  $\mathbf{y}_1$  to  $\mathbf{y}_2$  can be written as  $\mathbf{y}_2 = \mathbf{y}_1 A_2$ . Where  $A_2$  is the matrix



$$A_2 = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

And lastly we can show that  $\mathbf{y}_3 = \mathbf{y}_2 A_3$ , where  $A_3$  is the matrix

$$A_3 = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This whole transformation can be done in one step and stated by the equation  $\mathbf{y}_3 = \mathbf{y} W$ , where the transformation matrix  $W = A_1 A_2 A_3$ .

[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 5 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



$$W = \begin{pmatrix} 1/8 & 1/8 & 1/4 & 0 & 1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & 1/4 & 0 & -1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & 1/2 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & -1/2 & 0 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & 1/2 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & -1/2 & 0 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & 1/2 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & -1/2 \end{pmatrix}$$

It is also important to note that since every column of the individual matrices that comprise  $W$  is orthogonal to every other column, the matrices are invertible. Thus:

$$W^{-1} = A_3^{-1} A_2^{-1} A_1^{-1}$$

This is what allows us to reconstitute our image from the compressed form. To return to our original data we use the equation

$$\mathbf{y} = \mathbf{y}_3 W^{-1}$$

In general we can say that  $Q = IW$ , where  $Q$  is the row transformed matrix and  $I$  is the original image matrix. But, as stated before, the Haar wavelet transformation does

[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 6 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



Introduction

How Images are Stored

How the Wavelet ...

The Power of Linear ...

Approximation and ...

Applications of ...

Examples

Analysis

The Compressomatic

Home Page

Title Page



Page 7 of 27

Go Back

Full Screen

Close

Quit

these transformations to each column of the image matrix, *and then repeats them on each column* of the matrix. This is done by multiplying  $I$  on the left by the transpose of  $W$ . This gives us our final equation for the *row-and-column* transformed matrix  $T$ :

$$T = ((IW)^T W)^T = W^T IW \quad (1)$$

And from this we can easily derive the equation to return our row-and-column transformed matrix to the original image matrix:

$$I = ((T^T W^{-1})^T W^{-1}) = (W^{-1})^T T W^{-1} \quad (2)$$

## 4.1. An Example

In order to help the reader visualize this process I will provide an example. Suppose we start with an 8 x 8 image represented by matrix  $P$ :

$$P = \begin{pmatrix} 576 & 704 & 1152 & 1280 & 1344 & 1472 & 1536 & 1536 \\ 704 & 640 & 1156 & 1088 & 1344 & 1408 & 1536 & 1600 \\ 768 & 832 & 1216 & 1472 & 1472 & 1536 & 1600 & 1600 \\ 832 & 832 & 960 & 1344 & 1536 & 1536 & 1600 & 1536 \\ 832 & 832 & 960 & 1216 & 1536 & 1600 & 1536 & 1536 \\ 960 & 896 & 896 & 1088 & 1600 & 1600 & 1600 & 1536 \\ 768 & 768 & 832 & 832 & 1280 & 1472 & 1600 & 1600 \\ 448 & 768 & 704 & 640 & 1280 & 1408 & 1600 & 1600 \end{pmatrix}$$



Introduction

How Images are Stored

How the Wavelet ...

The Power of Linear ...

Approximation and ...

Applications of ...

Examples

Analysis

The Compressomatic

Home Page

Title Page



Page 8 of 27

Go Back

Full Screen

Close

Quit

Notice that the last row is our vector  $\mathbf{y}$  from the section on averaging and differencing. First we want to average and difference the rows of matrix  $P$ . In order to get the row averaged matrix  $Q$  we simply multiply  $P$  on the right by matrix  $W$ , our transformation matrix. This yields:

$$Q = \begin{pmatrix} 1200 & -272 & -288 & -64 & -64 & -64 & -64 & 0 \\ 1185 & -288 & -225 & -96 & 32 & 34 & -32 & -32 \\ 1312 & -240 & -272 & -48 & -32 & -128 & -32 & 0 \\ 1272 & -280 & -160 & -16 & 0 & -192 & 0 & 32 \\ 1256 & -296 & -128 & 16 & 0 & -128 & -32 & 0 \\ 1272 & -312 & -32 & 16 & 32 & -96 & 0 & 32 \\ 1144 & -344 & -32 & -112 & 0 & 0 & -96 & 0 \\ 1056 & -416 & -32 & -128 & -160 & 32 & -64 & 0 \end{pmatrix}$$

Notice that the last row in the matrix is identical to  $\mathbf{y}_3$ . We can now average and difference the columns of  $Q$  to get our row-and-column transformed matrix  $T$  by multiplying  $Q$  on the left by  $W^T$ . This yields:

$$T = \begin{pmatrix} 1212 & -306 & -146 & -54 & -24 & -68 & -40 & 4 \\ 30 & 36 & -90 & -2 & 8 & -20 & 8 & -4 \\ -50 & -10 & -20 & -24 & 0 & 72 & -16 & -16 \\ 82 & 38 & -24 & 68 & 48 & -64 & 32 & 8 \\ 8 & 8 & -32 & 16 & -48 & -48 & -16 & 16 \\ 20 & 20 & -56 & -16 & -16 & 32 & -16 & -16 \\ -8 & 8 & -48 & 0 & -16 & -16 & -16 & -16 \\ 44 & 36 & 0 & 8 & 80 & -16 & -16 & 0 \end{pmatrix}$$



[Introduction](#)[How Images are Stored](#)[How the Wavelet ...](#)[The Power of Linear ...](#)[Approximation and ...](#)[Applications of ...](#)[Examples](#)[Analysis](#)[The Compressomatic](#)[Home Page](#)[Title Page](#)[Page 9 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

Now matrix  $T$  is ready to be compressed. Compression will be discussed in Section 6.1.

## 4.2. Even More Power!

Another way to make these transformation matrices more powerful is to *normalize* the  $A_i$  matrices that comprise our transformation matrix  $W$ . By this I mean that we divide each column of each of the  $A_i$  matrices by its length. This results in  $A_i$  matrices that are *orthogonal*, meaning that each column of the matrix is length one, and orthogonal to each other column of the matrix. This also results in the transformation matrix  $M$  being orthogonal as well. This is called a *normalized wavelet transform*. Having an orthogonal transformation matrix has two main benefits. First, because the inverse of an orthogonal matrix is equal to its transpose, this greatly increases the speed at which we can perform the calculations that reconstitute our image matrix. Second, as you can see from Figure 1 normalized transformations tend to be closer to the original image. This is due to some powerful properties of orthogonal matrices. If  $Q$  is an orthogonal matrix, and  $\mathbf{x}$  and  $\mathbf{y}$  are two arbitrary vectors, then

$$\begin{aligned}\|Q\mathbf{x}\|^2 &= (Q\mathbf{x})^T(Q\mathbf{x}) \\ &= \mathbf{x}^T Q^T Q \mathbf{x} \\ &= \mathbf{x}^T I \mathbf{x} \\ &= \mathbf{x}^T \mathbf{x} \\ &= \|\mathbf{x}\|^2,\end{aligned}$$

which, in turn, implies that

$$\|Q\mathbf{x}\| = \|\mathbf{x}\|.$$

Essentially, when a vector is multiplied by an orthogonal matrix, *magnitude is preserved*.



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



*Page 10 of 27*

[Go Back](#)

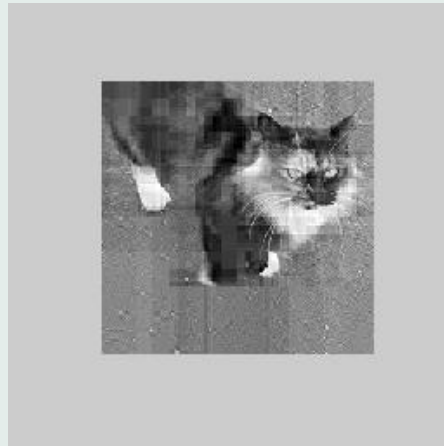
[Full Screen](#)

[Close](#)

[Quit](#)



(a) Normalized transformation matrix



(b) Non-Normalized transformation matrix

Figure 1: Both images compresses to 5.5 : 1 compression ratio.



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet...](#)

[The Power of Linear...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 11 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Also recall that

$$\begin{aligned}(Q \mathbf{x}) \cdot (Q \mathbf{y}) &= (Q \mathbf{x})^T (Q \mathbf{y}) \\ &= \mathbf{x}^T Q^T Q \mathbf{y} \\ &= \mathbf{x}^T I \mathbf{y} \\ &= \mathbf{x}^T \mathbf{y} \\ &= \mathbf{x} \cdot \mathbf{y},\end{aligned}$$

which means that the dot product of two vectors is unchanged when they are multiplied by an orthogonal matrix. These two facts, along with the fact that

$$\cos \theta = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|},$$

implies that when vectors are multiplied by an orthogonal matrix, angle is preserved as well. Since both magnitude and angle are preserved there is significantly less distortion produced in the rebuilt image when an orthogonal transformation matrix is used.

## 5. Approximation and Detail Coefficients

At this point it also worth further discussing approximation and detail coefficients. Once again I will use matrix  $P$  as an example. If we were to average and difference each row and column only once ( $T_1 = A_1^T P A_1$ ), we would essentially divide our 8 x 8 matrix into four 4 x 4 sub-matrices:



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 12 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

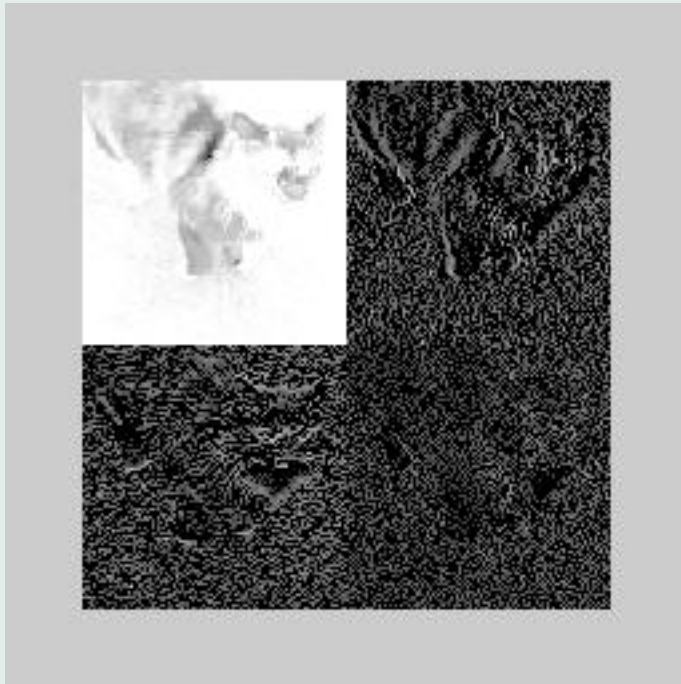


Figure 2: Figure 5 averaged and differenced once.



$$T_1 = \begin{pmatrix} 656 & 1169 & 1392 & 1552 & -16 & -15 & -48 & -16 \\ 816 & 1248 & 1520 & 1584 & -16 & -160 & -16 & 16 \\ 880 & 1040 & 1584 & 1552 & 16 & -112 & -16 & 16 \\ 688 & 752 & 1360 & 1600 & -80 & 16 & -80 & 0 \\ -16 & 47 & 16 & -16 & -48 & -49 & -16 & 16 \\ -16 & 96 & -16 & 16 & -16 & 32 & -16 & -16 \\ -48 & 48 & -16 & -16 & -16 & -16 & -16 & -16 \\ 80 & 80 & 16 & 0 & 80 & -16 & -16 & 0 \end{pmatrix}$$

The 4 x 4 matrix in the upper left-hand corner is comprised of approximation coefficients, and the other three are made of the detail coefficients. For a good visual example we take Figure 5, and average and difference it one time. We then display the image described by the resulting matrix (Figure 2). The image has been brightened in order to bring out the detail. The image in the upper left-hand corner is comprised of the approximation coefficient matrix. The images in the other three corners come from the detail coefficients. The white outline of the cat in these images correspond to large numbers in the detail coefficients. These occur at areas of the image where there is high contrast, and therefore a large difference between neighboring elements in the image matrix. When they are averaged and differenced the resulting detail coefficients are relatively large, and therefore show up as white pixels when we display the matrix as an image.

If we take our original matrix  $P$  and perform two rounds of averaging and differencing on it ( $T_2 = A_2^T A_1^T P A_1 A_2$ ), we end up with the following matrix:

[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 13 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



$$T_2 = \begin{pmatrix} 972 & 1512 & -236 & -56 & -16 & -88 & -32 & 0 \\ 840 & 1524 & -56 & -52 & -32 & -48 & -48 & 8 \\ -60 & -40 & -20 & -24 & 0 & 73 & -16 & -16 \\ 120 & 44 & -24 & 68 & 48 & -64 & 32 & 8 \\ 16 & 0 & -32 & 16 & -48 & -49 & -16 & 16 \\ 40 & 0 & -56 & -16 & -16 & 32 & -16 & -16 \\ 0 & -16 & -48 & 0 & -16 & -16 & -16 & -16 \\ 80 & 8 & 0 & 8 & 80 & -16 & -16 & 0 \end{pmatrix}$$

Only the  $2 \times 2$  that makes the upper left corner of  $T_2$  is now approximation coefficients. The remaining  $2 \times 2$  matrices that make up the  $4 \times 4$  matrix in the upper left corner, along with the  $4 \times 4$ 's that comprise the rest of the matrix are the all detail coefficients. This can be seen clearly if we take the image in Figure 5 and perform two rounds of averaging and differencing to the rows and columns. This results in the image in Figure 3. Again the image has been brightened in order to bring out the detail.

Finally, if we do one more round of averaging and differencing the rows and columns, we end up with with the row-and-column transformed matrix  $T$  from the last section (See Section 4.1). Only the element  $T_{1,1}$  is now a approximation coefficient. When we reach the fully row-and-column transformed matrix, this approximation coefficient is known as the overall approximation coefficient. We can get a feeling for what this means visually if we perform three rounds of averaging and differencing to Figure 5, and display the image in Figure 4. Once again the image has been brightened in order to bring out the detail.

Figures 2, 3, and 4 clearly visually illustrate the meaning of the detail coefficients. Note that all of these figures, along with Figure 5, are actually  $256 \times 256$ . Since  $256 = 2^8$ , there would actually be eight rounds of averaging and differencing to reach

Introduction

How Images are Stored

How the Wavelet...

The Power of Linear...

Approximation and...

Applications of...

Examples

Analysis

The Compressomatic

Home Page

Title Page



Page 14 of 27

Go Back

Full Screen

Close

Quit



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 15 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

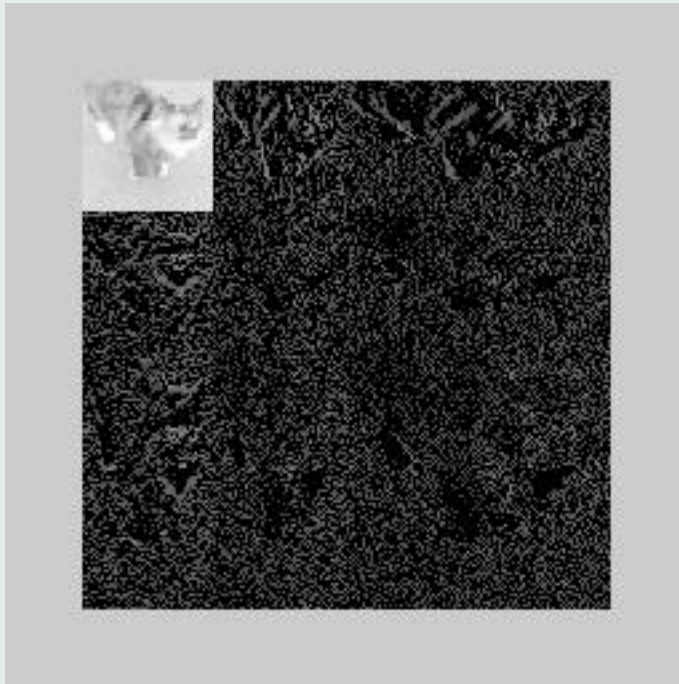


Figure 3: Figure 5 averaged and differenced twice.



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 16 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

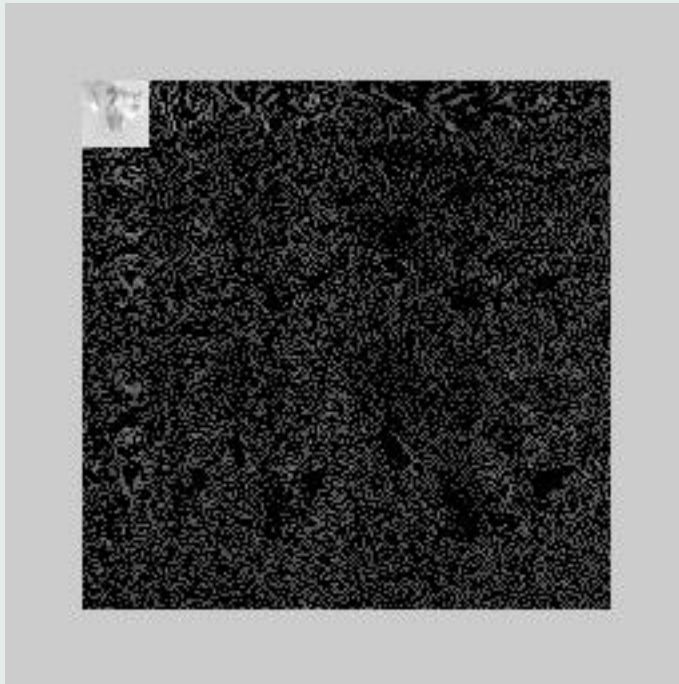


Figure 4: Figure 5 averaged and differenced three times.



[Introduction](#)[How Images are Stored](#)[How the Wavelet...](#)[The Power of Linear...](#)[Approximation and...](#)[Applications of...](#)[Examples](#)[Analysis](#)[The Compressomatic](#)[Home Page](#)[Title Page](#)[Page 17 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

the row-and-columned transformed matrix. But I think the reader can see the visual pattern that each round of averaging and differencing would produce.

## 6. Applications of Wavelet Transforms

Wavelets have broad applications in fields such as signal processing and medical imaging. Due to time and space constraints, I will only be discussing two in this paper. The two applications most applicable to this paper are wavelet image compression and the progressive transmission of image files over the internet.

### 6.1. Wavelet Compression

So what is the point to all of this? The point of doing the Haar wavelet transform is that areas of the original matrix that contain little variation will end up as small or zero elements in the Haar transform matrix. A matrix is considered *sparse* if it has a high proportion of zero entries. Matrices that are sparse take much less memory to store. Because we cannot expect the transformed matrices always to be sparse, we must consider wavelet compression

To perform wavelet compression we first decide on a non-negative threshold value known as  $\epsilon$ . We next let any value in the Haar wavelet transformed matrix whose magnitude is less than  $\epsilon$  be reset to zero. Our hope is that this will leave us with a relatively sparse matrix. If  $\epsilon$  is equal to zero we will not modify any of the elements and therefore we will not lose any information. This is known as *lossless compression*. *Lossy compression* occurs when  $\epsilon$  is greater than zero. Because some of the elements are reset to zero, some of our original data is lost. In the case of lossless compression we are able to reverse our operations and get our original image back. With lossy compression we are only able to build an approximation of our original image.

[Introduction](#)[How Images are Stored](#)[How the Wavelet ...](#)[The Power of Linear ...](#)[Approximation and ...](#)[Applications of ...](#)[Examples](#)[Analysis](#)[The Compressomatic](#)[Home Page](#)[Title Page](#)[Page 18 of 27](#)[Go Back](#)[Full Screen](#)[Close](#)[Quit](#)

## 6.2. Progressive Transmission

Many people frequently download images from the internet, a few of which are not even pornographic. Wavelet transforms speed up this process considerably. When a person clicks on an image to download it, the source computer recalls the wave transformed matrix from memory. It first sends the overall approximation coefficient and larger detail coefficients, and then the progressively smaller detail coefficients. As your computer receives this information it begins to reconstruct the image in progressively greater detail until the original image is fully reconstructed. This process can be interrupted at any time if the user decides that s/he does not want the image. Otherwise an user would only be able to see an image after the entire image file had been downloaded. Because a compressed image file is significantly smaller it takes far less time to download.

## 6.3. Summary of Compression

Image compression using the Haar wavelet transform can be summed up in a few simple steps.

1. Convert the image into a matrix format( $I$ ).
2. Calculate the row-and-column transformed matrix ( $T$ ) using Equation 1. The transformed matrix should be relatively sparse.
3. Select a threshold value  $\epsilon$ , and replace any entry in  $T$  whose magnitude is less than  $\epsilon$  with a zero. This will result in a sparse matrix denoted as  $S$ .
4. To get our reconstructed matrix  $R$  from matrix  $S$ , we use an equation similar to equation 2, but, because the inverse of an orthogonal matrix is equal to its



transpose, we modify the equation as follows:

$$R = W S W^{-1} \quad (3)$$

If you choose  $\epsilon$  equal to zero, then  $S = T$  and therefore  $R = I$ . This is what we referred to previously as lossless compression. Because  $\epsilon$  is equal to zero, no element of  $T$  is changed, and therefore no data is lost.

If you choose  $\epsilon$  greater than 0, then elements of  $T$  are reset to zero, and therefore some original data is lost. The reconstituted image will contain distortions. This is what we referred to as lossy compression. The secret of compression is to choose your  $\epsilon$  value so that compression is maximized while distortions in the reconstituted image are minimized.

We measure the level of compression by the *compression ratio*. The compression ratio is the ratio of nonzero entries in the transformed matrix  $T$  to the number of nonzero entries in the compressed matrix  $S$ . According to Dr. Colm Mulcahy of Spelman University, a compression ratio of 10 : 1 or greater means that  $S$  is sparse enough to have a significant savings in terms of storage and transmission time[4].

## 7. Examples

The image in Figure 5 is represented by a 256 by 256 matrix. For the sake of simplicity I will use the 8 by 8 matrix that comprises the upper left hand corner of the image. It is described by the following matrix:

[Introduction](#)

[How Images are Stored](#)

[How the Wavelet...](#)

[The Power of Linear...](#)

[Approximation and...](#)

[Applications of...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 19 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



Figure 5: The Original Image



- [Introduction](#)
- [How Images are Stored](#)
- [How the Wavelet ...](#)
- [The Power of Linear ...](#)
- [Approximation and ...](#)
- [Applications of ...](#)
- [Examples](#)
- [Analysis](#)
- [The Compressomatic](#)

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 20 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



$$I = \begin{pmatrix} 100 & 103 & 99 & 97 & 93 & 94 & 78 & 73 \\ 102 & 97 & 100 & 111 & 113 & 104 & 96 & 82 \\ 99 & 109 & 104 & 95 & 93 & 92 & 88 & 76 \\ 114 & 104 & 99 & 102 & 93 & 82 & 74 & 74 \\ 96 & 91 & 91 & 87 & 79 & 78 & 77 & 76 \\ 90 & 88 & 83 & 78 & 77 & 74 & 76 & 76 \\ 92 & 81 & 73 & 72 & 69 & 65 & 66 & 62 \\ 75 & 70 & 69 & 65 & 60 & 55 & 61 & 65 \end{pmatrix}$$

We can see the image represented by matrix  $I$  in figure 6a. I should also note that the colors in the blown up versions may not exactly correspond to the original image because the color scale is adjusted to the max and min values of each matrix.

If we perform the normalized wavelet transformation on  $I$  and choose  $\epsilon$  so as to achieve a compression ratio of approximately 5 : 1, we get the following compressed matrix:

$$S = \begin{pmatrix} 255 & 52 & 15 & 21 & 0 & 0 & 0 & 0 \\ 78 & 0 & 0 & 22 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

[Introduction](#)

[How Images are Stored](#)

[How the Wavelet...](#)

[The Power of Linear...](#)

[Approximation and...](#)

[Applications of...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 21 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



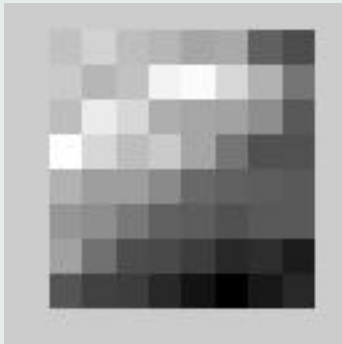
Page 22 of 27

[Go Back](#)

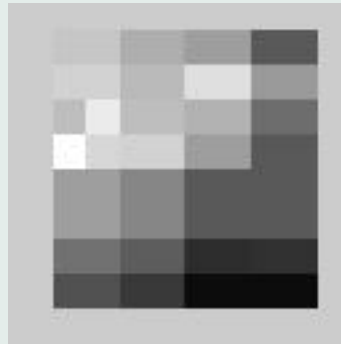
[Full Screen](#)

[Close](#)

[Quit](#)



(a) Original Image



(b) Rebuilt Compressed Image

Figure 6: It is easy to see the distortions present in the rebuilt image.



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 23 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

As you can see, the matrix is comprised mainly of zeros. This takes up much less memory to store. Instead of remembering each and every value in a matrix, the computer need only remember that the matrix is all zeros except in a few specific spots. This is significant when dealing with a 256 x 256 gray scale image that has 65,536 elements, and even more so with a typical 640 x 480 color image that has 921,600 elements.

When we restore the matrix using Equation 3, we end up with the following matrix:

$$R = \begin{pmatrix} 100 & 100 & 95 & 95 & 92 & 92 & 76 & 76 \\ 103 & 103 & 98 & 98 & 106 & 106 & 90 & 90 \\ 99 & 109 & 99 & 99 & 96 & 96 & 81 & 81 \\ 114 & 104 & 104 & 104 & 91 & 91 & 76 & 76 \\ 91 & 91 & 86 & 86 & 76 & 76 & 76 & 76 \\ 91 & 91 & 86 & 86 & 76 & 76 & 76 & 76 \\ 82 & 82 & 76 & 76 & 66 & 66 & 66 & 66 \\ 74 & 74 & 69 & 69 & 58 & 58 & 59 & 59 \end{pmatrix}$$

If you compare matrix  $R$  with matrix  $I$ , it is easy to see that although they are similar, there are small variations. These variations show up as distortions in the reconstructed image. These can be easily seen in figure 6b.

Other figures are included in order for the reader to be able to visualize the distortions that can occur with compression. Figure 7 shows the original image along with the reconstructed image after being compressed at a 10.3 : 1 ratio. Figure 8 shows the original image along with the reconstructed image after being compressed at a 10.2 : 1 ratio. In both of these examples it is easy to see that although the images are still clear, they do contain distortions due to the fact that  $\epsilon$  is greater than zero.



(a) Original Image



(b) Rebuilt Compressed Image

Figure 7: Image with 10.3 : 1 compression ratio.



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



[Page 24 of 27](#)

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)





[Introduction](#)

[How Images are Stored](#)

[How the Wavelet ...](#)

[The Power of Linear ...](#)

[Approximation and ...](#)

[Applications of ...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 25 of 27

[Go Back](#)

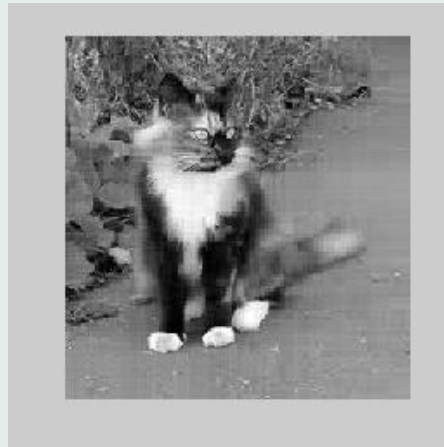
[Full Screen](#)

[Close](#)

[Quit](#)



(a) Original Image



(b) Rebuilt Compressed Image

Figure 8: Image with 10.2 : 1 compression ratio.



[Introduction](#)

[How Images are Stored](#)

[How the Wavelet...](#)

[The Power of Linear...](#)

[Approximation and...](#)

[Applications of...](#)

[Examples](#)

[Analysis](#)

[The Compressomatic](#)

[Home Page](#)

[Title Page](#)



Page 26 of 27

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## 8. Analysis

I think that it is clear to see that the wavelets have their advantages. They greatly increase the speed at which image files are transferred electronically and drastically decrease the amount of valuable memory needed to store them. According to Dr. Mulcahy, the Haar wavelet transform is the simplest, crudest wavelet transform[4]. It does, however, illustrate the power and potential of wavelet transforms. They have broad applications in fields such as signal processing, computer graphics, medical imaging, seismology, as well as the imaging applications previously discussed. After witnessing the power Of the Haar wavelet transform I can only image what is possible with a more powerful transform.

### A. The Compressomatic

In order to produce the compressed images used in this paper it was necessary to use Matlab. The file I produced is based on the three Matlab m-files written by Dr. Colm Mulcahy of Spelman University. They are indmat.m, wavmat.m, and compress.m. These files were subsequently modified and merged into a single m-file named project1.m by LiHong Huang, a fellow student at College of the Redwoods. LiHong's chief contribution was to the effort was to combine the three into a single m-file, and modifying the original program to allow it to read images directly. Dr. Mulcahy's original program required the user to enter the matrix that described an image. LiHong's changes allowed the powerful advantage of allowing the user to reference a image file (JPEG, GIF, etc...) directly. The program, after importing the image in an unsigned 8-bit integer format, converts each element to double precision so that matrix arithmetic is possible in Matlab. This was a huge improvement on the original program.

My main contribution to m-file was to remove unnecessary code, make minor formatting changes, and to comment the code so as to be more user friendly to other users.



Introduction

How Images are Stored

How the Wavelet ...

The Power of Linear ...

Approximation and ...

Applications of ...

Examples

Analysis

The Compressomatic

Home Page

Title Page



Page 27 of 27

Go Back

Full Screen

Close

Quit

I also modified the code so that any size square image could be used. Dr. Mulcahy's original and LiHong Huang's modification required a 256 x 256 pixel image.

You may view my Matlab m-file by [clicking on this link](#).

Dr. Mulcahy's original paper and m-files can be seen on his web site <http://www.spelman.edu/colm/wav.html>.

## References

- [1] Ames, Rebecca *For instruction in Adobe Photoshop*
- [2] Arnold, Dave *College of the Redwoods*
- [3] Huang, LiHong *For his help in constructing the Matlab M-Files*
- [4] Mulcahy, Colm *Image Compression Using the Haar Wavelet Transform*
- [5] Strang, Gilbert *Introduction to Linear Algebra*