The Haar Wavelet Transform: Compression and Reconstruction

Damien Adams and Halsey Patterson

December 13, 2006

Have you ever looked at an image on your computer? Of course you have.

- Images today aren't just stored on rolls of film.
- Most images today are stored or compressed using linear algebra.
- What does linear algebra have to do with images?

Images are made up of individual *pixels*.

- Pixels are squares of uniform color.
- Each pixel is represented by a number.
- Lower numbers are darker. Zero is completely black.

Tieing into linear algebra

The numbers are organized in a matrix. A typical image can have a lot of pixels-256x256, 640x480,1024x768..etc.

- We need a way to store images without storing all of that data.
- The answer is compression.
- Images are compressed and then retrieved (reconstructed) using **averaging and differencing**.

Conceptually, this works by **averaging** neighboring values and replacing the two with their average. So, the two numbers are replaced by one. **Differencing** allows us to keep track of the difference between the average and original values.

What it looks like

Here is an 8×8 matrix.

$$A = \begin{bmatrix} 210 & 215 & 204 & 225 & 73 & 111 & 201 & 106 \\ 169 & 145 & 245 & 189 & 120 & 58 & 174 & 78 \\ 87 & 95 & 134 & 35 & 16 & 149 & 118 & 224 \\ 74 & 180 & 226 & 3 & 254 & 195 & 145 & 3 \\ 87 & 140 & 44 & 229 & 149 & 136 & 204 & 197 \\ 137 & 114 & 251 & 51 & 108 & 164 & 15 & 249 \\ 186 & 178 & 69 & 76 & 132 & 53 & 154 & 254 \\ 79 & 159 & 64 & 169 & 85 & 97 & 12 & 202 \end{bmatrix}$$

Averaging

Let's grab an arbitrary row for example:

$$\begin{bmatrix} 45 & 11 & 30 & 24 & 45 & 38 & 0 & 23 \end{bmatrix}$$

Now we begin averaging

$$\begin{matrix} 45 & 11 \\ 30 & 24 \\ 45 & 38 \\ 0 & 23 \end{matrix}$$

These averages are now placed back into the row:

$$\begin{bmatrix} 28 & 27 & 41.5 & 11.5 & x & x & x & x \end{bmatrix}$$

Differencing

Differencing is taking the difference between the value on the left side of each pair and the average of each pair.

Averaged	First Value − Average	Differenced
28	45 − 28	17
27	30 − 27	3
41.5	45 − 41.5	3.5
11.5	0 − 11.5	−11.5

The result is our averaged and differenced row.

$$\begin{bmatrix} 28 & 27 & 41.5 & 11.5 & 17 & 3 & 3.5 & -11.5 \end{bmatrix}$$

More Averaging and Differencing

Damien Adams and Halsey Patterson

The averaging and differencing can be continued until there is just one averaged value (on the far left) and the rest are difference values.

- The difference values are called **detail coefficients**.

But surely isn't there a way to average and difference entire matrices?'

- This process is called **Wavelet Transforming** a matrix.

Naturally, Wavelet Transforming is done with linear algebra-namely block multiplication.

Wavelet Transforming

Let's look at which matrices are used for Wavelet Transforming. We will use the following for our 8×8:

$$W_1 = \begin{bmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 \end{bmatrix}$$

These transforming matrices average and difference a piece of the matrix at a time.

Wavelet Transforming

By computing AW_1, the rows of A are left with four averages and four detail coefficients. So, we need to transform more. Can you guess how?

- Block muliplication is used to get our W_2

$$W_2 = \begin{bmatrix} W_{2\times 2} & 0 \\ 0 & I \end{bmatrix}$$

Wavelet Transforming

We now have W_2.

$$W_2 = \begin{bmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, AW_1W_2 gives us a matrix where each row has two averages and six detail coefficients.

Wavelet Transforming

Similarly, we are able to get W_3.

$$W_3 = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Finally, we have that $W = W_1 W_2 W_3$ in this case, or more generally $W = W_1 W_2 W_3 ... W_n$. Finally, we can get our *wavelet transformed* matrix $T = AW$.

What now?

Okay, but where does the compression come in? We just have a matrix with a few average values and a bunch of detail coefficients!

- Detail coefficients let us know the difference in darkness between neighboring pixels.
- Small detail coefficients mean a small difference in the shade of neighboring pixels.
- Ignoring small differences may not change the big picture.

How small of differences can we ignore? We must set a threshhold, ϵ, for which any detail coefficient smaller is just set to zero. What effect will this have? Our new matrix is called a **sparse** matrix.

Compressed!

Zeroing out some values has resulted in a loss of data. The result has pros and cons, so try out a few ϵ's before sticking with one. For our image of Lena, let's try $\epsilon = 1$. We'll see her image later.

Reconstruction

Remember how viewing images on web pages used to be? That is image reconstruction going on. It turns out that multiplying W^{-1} on the left of our transformed matrix T, we get our reconstructed matrix R. Also, W^{-1} is much easier to calculate if W is orthogonal. Once we've got W^{-1}, we can see how to get R.

$$T = AW_1 W_2 W_3 \ldots W_n = AW \qquad \text{and} \qquad R = W^{-1} T = W^{-1} AW$$

This wont be quite as good as the original—.

Success

Let's see how our reconstructed matrix compares to the original.

Figure: Woman and Her Compression Using $\epsilon = 1$

The fact that the image was essential preserved means that our choice of ϵ was a success.

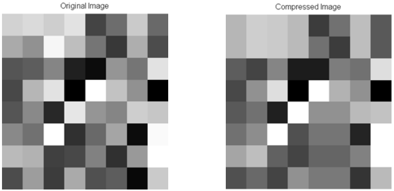

Figure: Previous Matrix A Comparison with $\epsilon = 25$

Conclusion

In summary, you begin with a matrix A that represents an image. Then average and difference to get your transformed matrix. Choose a ϵ value as a threshhold, and get alot of zeros in your matrix. Next, with your transformed matrix T, you reconstruct by multiplying it on the left by W^{-1}. This process is the **Haar Wavelet Transformation**.

The End

Special thanks to Dave Arnold for a lot of help, Colm Mulachy for the great Haar Wavelet Transform paper and the matrices that are used in Matlab to wavelet compress these images, and to Gilbert Strang for providing an excellent textbook and Matlab manual! Fin!