Haar Wavelet Image Compression

1. Preliminaries

Haar wavelet compression is an efficient way to perform both lossless and lossy image compression. It relies on averaging and differencing values in an image matrix to produce a matrix which is sparse or nearly sparse. A **sparse matrix** is a matrix in which a large portion of its entries are 0. A sparse matrix can be stored in an efficient manner, leading to smaller file sizes.

In these notes we will concentrate on grayscale images; however, rgb images can be handled by compressing each of the color layers separately. The basic method is to start with an image A, which can be regarded as an $m \times n$ matrix with values 0 to 255. In Matlab, this would be a matrix with unsigned 8-bit integer values. We then subdivide this image into 8×8 blocks, padding as necessary. It is these 8×8 blocks that we work with.

2. An Example

Below is a 512×512 pixel grayscale image of the flying buttresses of the Notre Dame Cathedral in Paris:

Figure 1. Flying buttresses of Notre Dame de Paris

And the following is the upper left 8×8 section of our image:

$$A = \begin{pmatrix} 88 & 88 & 89 & 90 & 92 & 94 & 96 & 97 \\ 90 & 90 & 91 & 92 & 93 & 95 & 97 & 97 \\ 92 & 92 & 93 & 94 & 95 & 96 & 97 & 97 \\ 93 & 93 & 94 & 95 & 96 & 96 & 96 & 96 \\ 92 & 93 & 95 & 96 & 96 & 96 & 96 & 95 \\ 92 & 94 & 96 & 98 & 99 & 99 & 98 & 97 \\ 94 & 96 & 99 & 101 & 103 & 103 & 102 & 101 \\ 95 & 97 & 101 & 104 & 106 & 106 & 105 & 105 \end{pmatrix}$$

We will concentrate on the first row:

$$r_1 = \begin{pmatrix} 88 & 88 & 89 & 90 & 92 & 94 & 96 & 97 \end{pmatrix}$$

Our transformation process will occur in three steps. The first step is to group all of the columns in pairs:

$$[88, 88], [89, 90], [92, 94], [96, 97]$$

We replace the first 4 columns of r_1 with the average of these pairs and replace the last 4 columns of r_1 with $\frac{1}{2}$ of the difference of these pairs. We will denote this new row as $r_1 h_1$:

$$r_1 h_1 = \begin{pmatrix} 88 & 89.5 & 93 & 96.5 & 0 & -0.5 & -1 & -0.5 \end{pmatrix}$$

The first 4 entries are called the **approximation coefficients** and the last 4 are called **detail coefficients**. Next, we group the first 4 columns of this new row:

$$[88, 89.5], [93, 96.5]$$

and replace the first 2 columns of $r_1 h_1$ with the average of the pairs and the next 2 columns of $r_1 h_1$ with $\frac{1}{2}$ of the difference of these pairs. We leave the last 4 rows of $r_1 h_1$ unchanged. We will denote this second new row as $r_1 h_1 h_2$:

$$r_1 h_1 h_2 = \begin{pmatrix} 88.75 & 94.75 & -0.75 & -1.75 & 0 & -0.5 & -1 & -0.5 \end{pmatrix}$$

Finally, our last step is to group the first 2 entries of $r_1 h_1 h_2$ together:

$$[88.75, 94.75]$$

and replace the first column of $r_1 h_1 h_2$ with the average of the pairs and the second column of $r_1 h_1 h_2$ with $\frac{1}{2}$ of the difference of these pairs. We leave the last 6 rows of $r_1 h_1 h_2$ unchanged. We will denote this last new row as $r_1 h_1 h_2 h_3$:

$$r_1 h_1 h_2 h_3 = \begin{pmatrix} 91.75 & -3 & -0.75 & -1.75 & 0 & -0.5 & -1 & -0.5 \end{pmatrix}$$

We then repeat this process for the remaining rows of A. After this, we repeat this same process to columns of A, grouping rows in the same manner as columns. The resulting matrix is:

$$
\begin{pmatrix}
96 & -2.0312 & -1.5312 & -0.2188 & -0.4375 & -0.75 & -0.3125 & 0.125 \\
-2.4375 & -0.0312 & 0.7812 & -0.7812 & 0.4375 & 0.25 & -0.3125 & -0.25 \\
-1.125 & -0.625 & 0 & -0.625 & 0 & 0 & -0.375 & -0.125 \\
-2.6875 & 0.75 & 0.5625 & -0.0625 & 0.125 & 0.25 & 0 & 0.125 \\
-0.6875 & -0.3125 & 0 & -0.125 & 0 & 0 & 0 & -0.25 \\
-0.1875 & -0.3125 & 0 & -0.375 & 0 & 0 & -0.25 & 0 \\
-0.875 & 0.375 & 0.25 & -0.25 & 0.25 & 0.25 & 0 & 0 \\
-1.25 & 0.375 & 0.375 & 0.125 & 0 & 0.25 & 0 & 0.25
\end{pmatrix}
$$

Notice that this resulting matrix has several 0 entries and most of the remaining entries are close to 0. This is a result of the differencing and the fact that adjacent pixels in an image generally do no differ by much.

We will now discuss how to implement this process using matrix multiplication below.

3. Haar Wavelet Transformation Matrix

If we let

$$
H_1 =
\begin{pmatrix}
\frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\
0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\
0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\
0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 \\
0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\
0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2}
\end{pmatrix}
$$

then AH_1 is equivalent to the first step above applied to all of the rows of A. In particular,

$$
AH_1 =
\begin{pmatrix}
88 & 89.5 & 93 & 96.5 & 0 & -0.5 & -1 & -0.5 \\
90 & 91.5 & 94 & 97 & 0 & -0.5 & -1 & 0 \\
92 & 93.5 & 95.5 & 97 & 0 & -0.5 & -0.5 & 0 \\
93 & 94.5 & 96 & 96 & 0 & -0.5 & 0 & 0 \\
92.5 & 95.5 & 96 & 95.5 & -0.5 & -0.5 & 0 & 0.5 \\
93 & 97 & 99 & 97.5 & -1 & -1 & 0 & 0.5 \\
95 & 100 & 103 & 101.5 & -1 & -1 & 0 & 0.5 \\
96 & 102.5 & 106 & 105 & -1 & -1.5 & 0 & 0
\end{pmatrix}
$$

Similarly, by defining H_2 by:

3

$$H_2 = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

then then AH_1H_2 is equivalent to the two steps above applied to all of the rows of A. In particular,

$$AH_1H_2 = \begin{pmatrix} 88.75 & 94.75 & -0.75 & -1.75 & 0 & -0.5 & -1 & -0.5 \\ 90.75 & 95.5 & -0.75 & -1.5 & 0 & -0.5 & -1 & 0 \\ 92.75 & 96.25 & -0.75 & -0.75 & 0 & -0.5 & -0.5 & 0 \\ 93.75 & 96 & -0.75 & 0 & 0 & -0.5 & 0 & 0 \\ 94 & 95.75 & -1.5 & 0.25 & -0.5 & -0.5 & 0 & 0.5 \\ 95 & 98.25 & -2 & 0.75 & -1 & -1 & 0 & 0.5 \\ 97.5 & 102.25 & -2.5 & 0.75 & -1 & -1 & 0 & 0.5 \\ 99.25 & 105.5 & -3.25 & 0.5 & -1 & -1.5 & 0 & 0 \end{pmatrix}$$

Finally, if we define

$$H_3 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

then then $AH_1H_2H_3$ is equivalent to the all three steps above applied to all of the rows of A. In particular,

$$AH_1H_2H_3 = \begin{pmatrix} 91.75 & -3 & -0.75 & -1.75 & 0 & -0.5 & -10 & -0.5 \\ 93.125 & -2.375 & -0.75 & -1.5 & 0 & -0.5 & -10 & 0 \\ 94.5 & -1.75 & -0.75 & -0.75 & 0 & -0.5 & -0.5 & 0 \\ 94.875 & -1.125 & -0.75 & 0 & 0 & -0.5 & 0 & 0 \\ 94.875 & -0.875 & -1.5 & 0.25 & -0.5 & -0.5 & 0 & 0.5 \\ 96.625 & -1.625 & -20 & 0.75 & -10 & -10 & 0 & 0.5 \\ 99.875 & -2.375 & -2.5 & 0.75 & -10 & -10 & 0 & 0.5 \\ 102.375 & -3.125 & -3.25 & 0.5 & -10 & -1.5 & 0 & 0 \end{pmatrix}$$

If we let H be the product of these three matrices, then

$$H = H_1 H_2 H_3 = \begin{pmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{4} & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & -\frac{1}{4} & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{8} & \frac{1}{8} & -\frac{1}{4} & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ \frac{1}{8} & -\frac{1}{8} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{8} & -\frac{1}{8} & 0 & \frac{1}{4} & 0 & 0 & -\frac{1}{2} & 0 \\ \frac{1}{8} & -\frac{1}{8} & 0 & -\frac{1}{4} & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{8} & -\frac{1}{8} & 0 & -\frac{1}{4} & 0 & 0 & 0 & -\frac{1}{2} \end{pmatrix}$$

To apply the procedure to the columns, we just multiply A on the left by H^T. So the resulting matrix is

$$H^T A H = \begin{pmatrix} 96 & -2.0312 & -1.5312 & -0.2188 & -0.4375 & -0.75 & -0.3125 & 0.125 \\ -2.4375 & -0.0312 & 0.7812 & -0.7812 & 0.4375 & 0.25 & -0.3125 & -0.25 \\ -1.125 & -0.625 & 0 & -0.625 & 0 & 0 & -0.375 & -0.125 \\ -2.6875 & 0.75 & 0.5625 & -0.0625 & 0.125 & 0.25 & 0 & 0.125 \\ -0.6875 & -0.3125 & 0 & -0.125 & 0 & 0 & 0 & -0.25 \\ -0.1875 & -0.3125 & 0 & -0.375 & 0 & 0 & -0.25 & 0 \\ -0.875 & 0.375 & 0.25 & -0.25 & 0.25 & 0.25 & 0 & 0 \\ -1.25 & 0.375 & 0.375 & 0.125 & 0 & 0.25 & 0 & 0.25 \end{pmatrix}$$

The important part of this process is that it is reversible. In particular, H is an invertible matrix. If

$$B = H^T A H$$

then

$$A = (H^T)^{-1} M H^{-1}$$

B represents the compressed image of A (in the sense that it is sparse). Moreover, since H is invertible, this compression is lossless.

4. Lossy Compression

The Haar wavelet transform can be used to perform lossy compression so that the compressed image retains its quality. First, the **compression ratio** of an image is the ratio of the non-zero elements in the original to the non-zero elements in the compressed image.

Let A be one of our 8×8 blocks and $H^T A H$ be the Haar wavelet compressed image of A. $H^T A H$ contains many detail coefficients which are close to 0. We will pick a number $\epsilon > 0$ and set all of the entries of $H^T A H$ with absolute value at most ϵ to 0. This matrix now has more 0 values and represents a more compressed image. When we decompress this image using the inverse Haar wavelet transform, we end up with an image which is close to the original.

For example, with our matrix A from the example, if we choose $\epsilon = 0.25$ then we end up with the matrix:

$$\begin{pmatrix} 96 & -2.0312 & -1.5312 & 0 & -0.4375 & -0.75 & -0.3125 & 0 \\ -2.4375 & 0 & 0.7812 & -0.7812 & 0.4375 & 0 & -0.3125 & 0 \\ -1.125 & -0.625 & 0 & -0.625 & 0 & 0 & -0.375 & 0 \\ -2.6875 & 0.75 & 0.5625 & 0 & 0 & 0 & 0 & 0 \\ -0.6875 & -0.3125 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.3125 & 0 & -0.375 & 0 & 0 & 0 & 0 \\ -0.875 & 0.375 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.25 & 0.375 & 0.375 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

This matrix represents a compression ratio of $\frac{48}{27} = 1.7$.

On the next page are some examples of our beginning image, compressed by various ratios

FIGURE 2. Original Image

FIGURE 3. 10.1 Compression Ratio
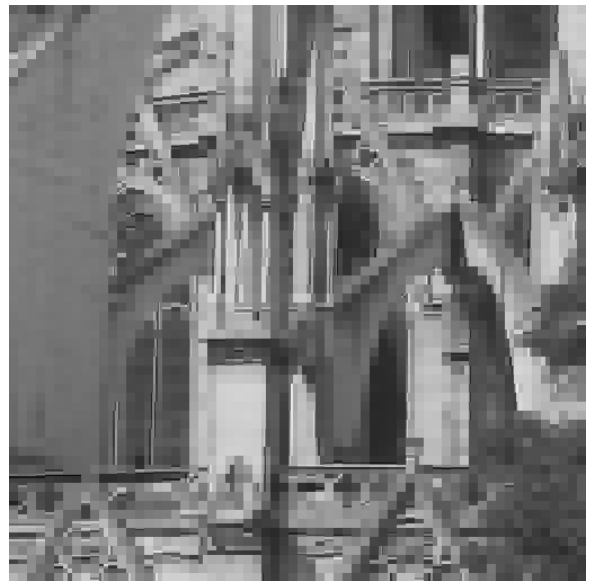

FIGURE 4. 30:1 Compression Ratio

FIGURE 5. 50:1 Compression Ratio

5. Normalization

We can achieve both a speedup in processing and an improvement in image quality by normalizing the transformation above. Notice that H_1, H_2 and H_3 have orthogonal columns. We produce the matrices $\tilde{H}_1$, $\tilde{H}_2$ and $\tilde{H}_3$ from H_1, H_2 and H_3 by normalizing each column of the starting matrix to length 1. Thus, $\tilde{H}_1$, $\tilde{H}_2$ and $\tilde{H}_3$ are orthogonal matrices. If we set $\tilde{H} = \tilde{H}_1\tilde{H}_2\tilde{H}_3$, then $\tilde{H}$ is orthogonal. This means that $\tilde{H}^{-1} = \tilde{H}^T$. Thus, to decompress an image, we only need to transpose $\tilde{H}$ rather than calculate its image. Moreover, because an orthogonal matrix preserves angles and lengths, using $\tilde{H}$ usually results in better image quality when applying lossy compression.

The two images below represent a 10:1 compression using the standard and normalized Haar wavelet transform. Notice that the rounded and angled sections in the normalized image are of a higher quality.

FIGURE 6. Standard, 10:1 Compression

FIGURE 7. Normalized, 10:1 Compression

6. Progressive Transmission

The Haar wavelet transformation can also be used to provide progressive transmission of a compressed image. An easy way to do this is consider the images on page 7. First the 50:1 image in its Haar wavelet compressed state is sent. The receiving computer decompresses this to show the 50:1 image. Next, only the detail coefficients which were zeroed out in the 50:1 image but not in the 30:1 image are sent. This new data along with the 50:1 image allows us to reconstruct the 30:1 image. Similarly, the detail coefficients which were zeroed out to give the 30:1 image but not the 10:1 image are sent. From this we construct the 10:1 image. Finally the remaining detail coefficients are sent, allowing us to reconstruct the original image. We can stop this process at any time to view the compressed image rather than the original.

7. MATLAB Code

The Matlab code and images used in these notes can be found at:

`http://www.math.ohio-state.edu/~husen/teaching/wi2010/572/572.html`