IMAGE COMPRESSION USING THE HAAR WAVELET

Ho Anh Tuy

Hanoi University of Technology

Nguyen Vinh An

Hanoi Open University

Abstract. The wavelet transform is a new arrival on the mathematical scene. It is widely applied in the area of engineering, image processing. In this paper, we present the main concepts of wavelet, the averaging and differencing technique related to a wavelet decomposition, a linear algebra implementation of the Haar wavelet transform.

1. What is wavelet?

A wavelet is a function that has finite energy and has an average of zero.

Wavelets are used for sub-band coding, signal and image processing, denoising noisy data, detecting self similarity in a time series and compression. Wavelets are also to be used in many other fields.

Wavelet give us the ability to cut up the data into different frequency components, which can be matched to the scale or size of the function.

Sine and cosine functions extend in either direction to infinity, however, wavelets are just for a small portion of time, and are called local. We can approximate functions that have spikes, irregularities or are choppy.

Some examples of mother wavelets:

a) Haar Wavelet

b) Daubechie's 4 Wavelet

c) Daubechie's 6 Wavelet

2. Why use Wavelets?

The Haar Wavelet transform provide a method of imaging compression so that it takes up less memory and therefore transmits faster. Discrete Cosine Transform (DCT) introduces block wide based noise while wavelet transforms for image porcessing tends to throw away noise, so wavelet tries to make the image easy to look at.

To use wavelets for compression, a mother wavelet, such as a Haar or Daubechie's 4 or 6 is selected. Here we investigate the use of Haar for image compression. The Haar wavelet algorithm has the advantage of being simple to compute and easier to understand. The Daubechies D4 algorithm has a slightly

higher computational overhead and is conceptually more complex. There is overlap between iterations in the Daubechies D4 transform step. This overlap allows the Daubechies D4 algorithm to pick up detail that is missed by the Haar wavelet algorithm..

3. Haar wavelets

3.1 Haar Wavelet basis functions

The Haar transform uses square pulses to approximate the original function. The basis functions for Haar wavelets at some level all look like a unit pulse, shifted along the x-axis. The basis functions are called *scales* and are usually denoted as functions $\Phi(t)$, where t denotes time.

The Haar scales are all of the unit pulses

$$\phi(t) = \begin{cases} 1 & -t \le 0 \text{ and } t-1 < 0 \\ 0 & \text{otherwise} \end{cases}$$

The functions $\Phi(t-s)$ are the shifted pulses, shifted by s units to the right.

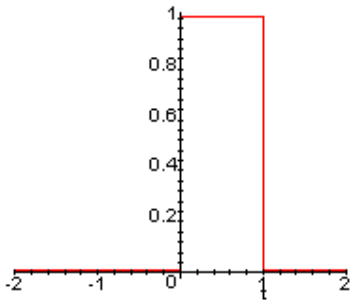


Figure 1: Plot of the unshifted Haar scale function.

To move to a higher level of detail, we consider the functions $\Phi(2t-s)$, which are the same bars halved in width. Suppose we have a step function q whose value is 5 on the interval [0, 0.5) and whose value is 3 on the interval [0.5, 1), and 0 otherwise. Then we can represent q as the function $5\Phi(2t) + 3\Phi(2t-1)$. The closest representation in the lower resolution would be $4\Phi(t)$, which is determined by averaging the previous coefficients (clearly, the average of 5 and 3 is 4). The question is how to get from the lower resolution space back to the higher resolution space? The answer is to know the *wavelets,* or the functions that span the space that contains the information we have discarded in moving from one resolution to another. The wavelet can be expressed as a linear combination of the basis vectors for the higher detail space, since it lies inside the higher resolution space which is spanned by those basis functions. For the Haar case, there is one wavelet $\psi := t \rightarrow \Phi(2t) + \Phi(2t-1)$, which looks like a zig-zag square wave:

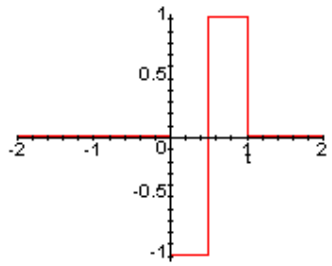


Figure 2: The Haar Wavelet

To reconstruct the original function, we can easily see that $q := t \to 4\ \Phi(t) - \psi(t)$. In general, we can take the two coefficients of the high resolution basis and re-express them as coefficients of the lower resolution basis plus the wavelets - all we are really doing is a change of basis, which we might usually write as a matrix multiplication!

3.2 The Haar Wavelet transform

3.2.1 The Haar Forward Transform

Each step in the forward Haar transform calculates a set of wavelet coefficients and a set of averages. If a data set S_0, S_1, S_{N-1} contains N elements, there will be $N/2$ averages and $N/2$ coefficients values. The averages are stored in the lower half of the N element array and the coefficients are stored in the upper half. The averages become the input for the next step in the calculation, where $N_{i+1} = N_i/2$. The recursive iterations continue until a single average and a single coefficient are calculated. This replaces the original data set of N elements with an average, followed by a set of coefficients whose size is an increasing power of two (e.g., 2^0, 2^1, 2^2)

The Haar equations to calculate an average (a_i) and a wavelet coefficient (c_i) from a data set are shown below:

$$a_i = \frac{S_i + S_{i+1}}{2}; \qquad c_i = \frac{S_i + S_{i+1}}{2}$$

Let's start with a simple example. Suppose we have a one-dimensional 'image' with resolution of four pixels: v = [19 13 7 3]

We can represent this in the *Haar basis* by computing a wavelet transform. In general, if the data string has length equal to 2^k, then the transformation process will consist of k steps. In this case, there will be 2 steps since $4 = 2^2$.

We perform the following operations on the entries of the vector v:

Step 1:

a. Divide the entries of v into two pairs: (19, 13), (7,3).

b. Form the average of each of these pairs:

$$\frac{19+13}{2}=16; \qquad \frac{7+3}{2}=5$$

To recover the original four pixel values from two averaged values, we calculate some *detail coefficients*. The first detail coeficient is 3 since the average is 3 less than 19 and 3 more than 13. This single number allows us to recover the first two pixels of our four-pixel image. Similarly, the second detail coefficient is 2, since 5 + 2 = 7 and 5 – 2 = 3.

Thus, we have decomposed the original image into a lower resolution (two pixel) and a pair of detail coefficients: v1 = [16 5 3 2]

Step 2:

Repeating this process recursively on the averages give the full decomposition:

Resolution	Average	Detail coefficients
4	[19 13 7 3]	
2	[16 5]	[3 2]
1	[10.5]	[5.5]

Thus, the wavelet transform of original four pixel image is given by

[10.5 5.5 3 2]

The way we computed the wavelet transform by recursively averaging and differencing coefficients is called *filter bank*. Given the transform, we can reconstruct the image to any resolution by recursively adding and substructing the detail coefficients from the lower resolution versions.

3.2.2 The inverse transform

The data input to the forward transform can be perfectly reconstructed using the following equations:

$$s_i = a_i + c_i$$
$$s_{i+1} = a_i - c_i$$

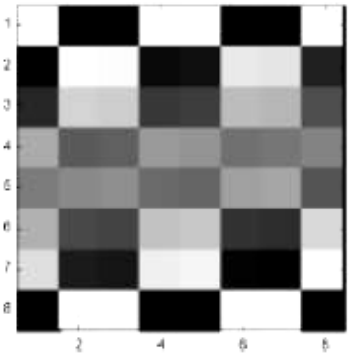
Storing the image's wavelet transform is advantage over the image itself because a large number of the detail coefficients turn out to be very small in magnitude. We can remove these small coefficients and which introduces only small errors in the reconstructed image ("Lossy" image compression).

3.2.3 Application to image compression

The basic idea behind this method of compression is to treat a digital image as an array of numbers i.e., a matrix. Each image consists of a fairly large number of little squares called **pixels** (picture elements). The matrix corresponding to a digital image assigns a whole number to each pixel. For example, in the case of a 256x256 pixel gray scale image, the image is stored as a 256x256 matrix, with each

element of the matrix being a whole number ranging from 0 (for black) to 225 (for white).

For example, the original image is represented by the following matrix (A):



$$A = \begin{bmatrix} 64 & 2 & 3 & 61 & 60 & 6 & 7 & 57 \\ 9 & 55 & 54 & 12 & 13 & 51 & 50 & 16 \\ 17 & 47 & 46 & 20 & 21 & 43 & 42 & 24 \\ 40 & 26 & 27 & 37 & 36 & 30 & 31 & 33 \\ 32 & 34 & 35 & 29 & 28 & 38 & 39 & 25 \\ 41 & 23 & 22 & 44 & 45 & 19 & 18 & 48 \\ 49 & 15 & 14 & 52 & 53 & 11 & 10 & 56 \\ 8 & 58 & 59 & 5 & 4 & 62 & 63 & 1 \end{bmatrix}$$

Figure 3: Original image

We can perform a 2 D Haar transform (the averaging and differencing) by first performing a 1 D Haar transform on each row and the on each column. , the results are row averages in the first column and the detail coefficients in the remaining columns of that row.

$$\begin{bmatrix} 32.5 & 0 & .5 & .5 & 31 & -29 & 27 & -25 \\ 32.5 & 0 & -.5 & -.5 & -23 & 21 & -19 & 17 \\ 32.5 & 0 & -.5 & -.5 & -15 & 13 & -11 & 9 \\ 32.5 & 0 & .5 & .5 & 7 & -5 & 3 & -1 \\ 32.5 & 0 & .5 & .5 & -1 & 3 & -5 & 7 \\ 32.5 & 0 & -.5 & -.5 & 9 & -11 & 13 & -15 \\ 32.5 & 0 & -.5 & -.5 & 17 & -19 & 21 & -23 \\ 32.5 & 0 & .5 & .5 & -25 & -27 & -19 & 31 \end{bmatrix}$$

Now if we apply the averaging and differencing to the column, we get the following matrix

$$\begin{bmatrix} 32.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & -4 & 4 & -4 \\ 0 & 0 & 0 & 0 & 4 & -4 & 4 & -4 \\ 0 & 0 & .5 & .5 & 27 & -25 & 23 & -21 \\ 0 & 0 & -.5 & -.5 & -11 & 9 & -7 & 5 \\ 0 & 0 & .5 & .5 & -5 & 7 & -9 & 11 \\ 0 & 0 & -.5 & -.5 & 21 & -23 & 25 & -27 \end{bmatrix}$$

This is matrix that represents our image with one overall average in the upper left-hand corner of the matrix. The remaining components are all detail coefficients that represent the amount of detail in that area of the image.

By doing threshold, we choose a number δ and set all elements with magnitude less than δ to zero, we have this simple matrix:

$$\begin{matrix} 32.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 27 & -25 & 23 & -21 \\ 0 & 0 & 0 & 0 & -11 & 9 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & -9 & 11 \\ 0 & 0 & 0 & 0 & 21 & -23 & 25 & -27 \end{matrix}$$

By choosing $\delta > 0$, some entries of the matrix will be set to zero, therefore some details will be lost and the image will be compressed. The ratio of nonzero entries in the transformed matrix $(S=W^TAW)$ to the number of nonzero entries in the compressed matrix obtained from S by applying the threshold δ is defined as *compression ratio*.

We can reconstruct the original image by applying the inverse wavelet transform, by doing this we get approximation of the original matrix:

$$\begin{matrix} 59.5 & 5.5 & 7.5 & 57.5 & 55.5 & 9.5 & 11.5 & 53.5 \\ 5.5 & 59.5 & 57.5 & 7.5 & 9.5 & 55.5 & 53.5 & 11.5 \\ 21.5 & 43.5 & 41.5 & 23.5 & 25.5 & 39.5 & 32.5 & 32.5 \\ 43.5 & 21.5 & 23.5 & 41.5 & 39.5 & 25.5 & 32.5 & 32.5 \\ 32.5 & 32.5 & 39.5 & 25.5 & 23.5 & 41.5 & 21.5 & 21.5 \\ 32.5 & 32.5 & 25.5 & 39.5 & 41.5 & 23.5 & 43.5 & 43.5 \\ 53.5 & 11.5 & 9.5 & 55.5 & 57.5 & 7.5 & 5.5 & 59.5 \\ 11.5 & 53.5 & 55.5 & 9.5 & 7.5 & 57.5 & 59.5 & 5.5 \end{matrix}$$

Suppose that A is the matrix corresponding to a certain image. The Haar transform is carried out by performing the above operations on each row of the matrix A and then by repeating the same operations on the columns of the resulting matrix. The row-transformed matrix is AW. Transforming the columns of AW is obtained by multiplying AW on the left by the matrix W^{T} (the transpose of W). Thus, the Haar transform takes the matrix A and stores it as $W^{\mathrm{T}}AW$. Let S denote the transformed matrix:

$$S = W^T AW .$$

Using the properties of inverse matrix, we can retrieve our original matrix:

$$A = (W^T)^{-1} S W^{-1} = (W^{-1})^T S W^{-1} .$$

This allows us to see the original image (decompressing the compressed image).

Using wavelet transformation and inverse wavelet transformation, we have lost some of the detail in the image but it would not be noticeable in most cases.

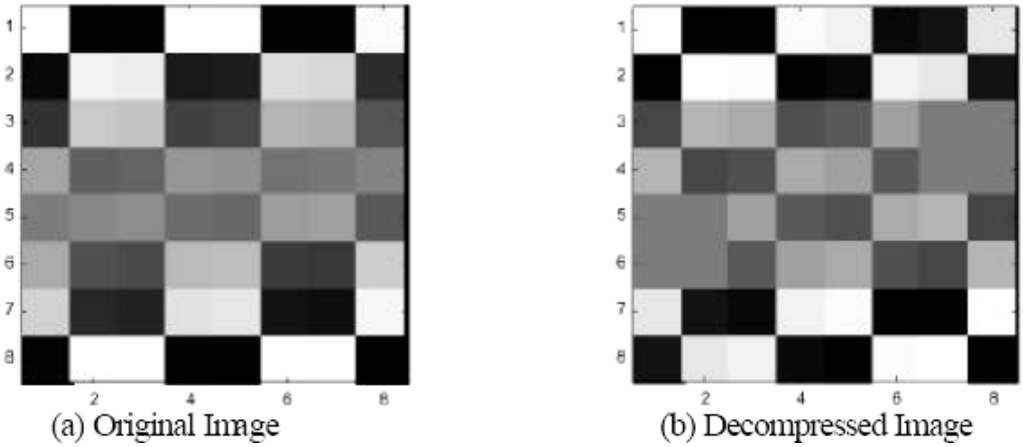


Figure 4. Original image and decompressing the compressed image.

4. Wavelet transform: A linear Algebra view

4.1 The Forward transform

Averaging and differencing can be calculated by matrix multiplication of the signal [S_0, S_1,..., S_N] and the vector of the same size [0.5,0.5,0,0,...0]. This is the scaling vector. The first coefficient is calculated by the inner product of the signal and the vector [0.5,- 0.5,0,0, ...0]. This is the wavelet vector.

The next average and coefficient are calculated by shifting the scaling h_i and wavelet vectors g_i by two and calculating the inner products.

The scaling and wavelet values for the Haar transform are shown in below matrix form

h0	h1	0	0	...
g0	g1	0	0	...
0	0	h0	h1	...
0	0	g0	g1	...
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

The first step of the forward Haar transform for an eight element signal is shown below. Here signal is multiplied by the forward transform matrix

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \Leftarrow \begin{bmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ a_2 \\ c_2 \\ a_3 \\ c_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix}$$

The arrow represents a split operation that reorders the results so that the average values are in the first half of the vector and the coefficients are in the second half.

The next step would be multiply the a_i values by a 4 x 4 transform matrix, generating two new averages and two new coefficients which would replace the averages in the first step.

The last step would multiply these new averages by a 2 x 2 matrix generating the final average and the final coefficients.

Figure 5: A small picture of Lena after one pass of the Haar transform

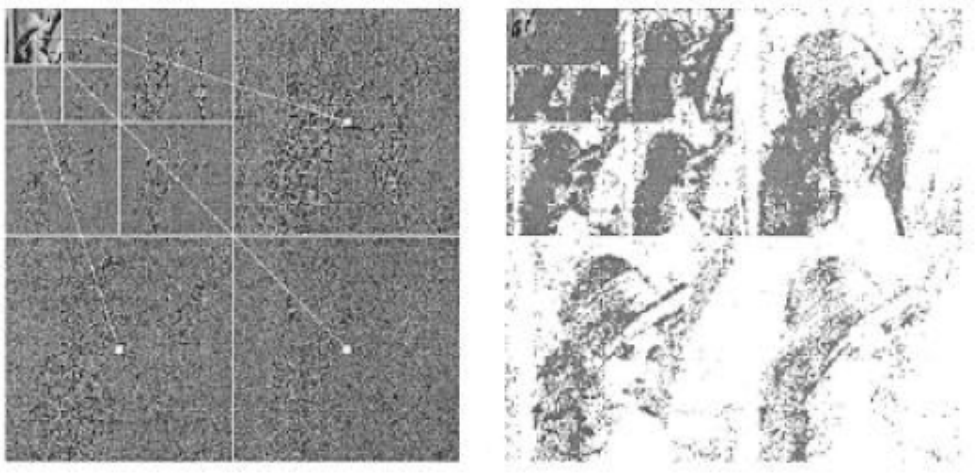


Figure 6. Three level decomposition of Lena image

4.2 The Inverse transform

Like the forward Haar transform, we can use the linear algebra terms to described the inverse transform. The matrix operation to reverse the first step of the Haar transform for an eight element signal is shown below.

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ c_0 \\ a_1 \\ c_1 \\ a_2 \\ c_2 \\ a_3 \\ c_3 \end{bmatrix} \Leftarrow \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

The arrow represents a merge operation that interleaves the averages and the coefficients.

5. Conclusion

We can sum up the Haar wavelet transform by following steps:

Convert the image into matrix format (I).

Calculate the row and column transformed matrix (T) using

$T = W^T I W$. The transform should be relatively sparse

Select the threshold value δ and replace any value of T less than δ with a zero. We will receive a sparse matrix which is denoted as S.

To get our reconstructed matrix R from matrix S, we use equation

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

Because the inverse of an orthogonal matrix is equal to its transpose, we modify the equation as : $R = W S W^{-1}$.

If $\delta = 0$ then $S = T$ therefore $R = I$. This is called *lossless compression.*

If $\delta > 0$, some of components of T is set to zero, some original data to be lost, the reconstructed image is distortion. This is called *lossy compression.*

We should choose δ carefully so that the compression is maximized while distortion of the reconstructed image is minimized.

The compression ratio is measured by the the ratio of nonzero entries in the transformed matrix T to the number of nonzero entries in the compressed matrix S.

References

1. Ian Kaplan, Jannuary *A linear Algebra View of the Wavelet Transform,* 2002.
2. Lewis, A. S and knowleg, G. Image Compression Using the 2-D Wavelet Transform, *IEEE Trans. IP,* vol. 1, no. **2**, April 1992, pp.244-250.
3. Emil Mikulic, *Haar Wavelet Transform,* 2004
4. *Application to Image Compression,* http://aix1.uotta.ca/~jkhoury/haar.htm
5. Alex Nicolaou, *A Wavelet Wading Pool*, 1996.