Image Compression

Mike Oremus

Ottopaskal Rice
ottomanbuski@hotmail.com

December 5, 2000

Abstract

People are always concerned with computer memory. It seems that every new computer boasts high memory and multiple possibilities for memory upgrades. Images, such as photos, take up large amounts of memory and tend to max out a computers storage space. Unfortunately people like images a lot and they fill their computers to the brim. In this article a method of reducing the massive amounts of memory that images take up will be outlined.

1. Interpreting Images Digitally

Images consist of numerous tiny dots known as "pixels". These pixels vary in color and are set up geometrically so that they make up an image. In other words, an image is really a whole bunch of variously colored tiny dots. For example, consider the image of two cows in Figure 1. It appears to be a continuous image, but it is really composed of a

Figure 1: Image of two cows.

discrete number of pixels ("picture elements") that are so tiny they are almost invisible to the human eye.

In order to make the mathematics a bit simpler, this article will concentrate on black and white images. Thus, the image in Figure 1 is changed to grayscale and resized so that it is exactly 256×256 pixels in dimension. This image is shown in Figure 2.

In Figure 3, the pixels become visible when an image is magnified. Notice that the pixels in Figure 3 form rows and columns. In linear algebra, matrices also form rows and columns. If each pixel is assigned a number based on its color, then a matrix is constructed. This is how images can be represented by matrices.

$$\begin{bmatrix} 12 & 34 & 21 & 12 \\ 9 & 23 & 45 & 12 \\ 8 & 13 & 14 & 23 \\ 21 & 45 & 32 & 11 \end{bmatrix}$$

In Figure 4, the cows of Figure 2 are shown again, but with a shade bar on the right to reinforce the idea of placing numerical values on shade/color. The shade bar to the right of the image identifies the numerical values for different shades of gray. Light shades are

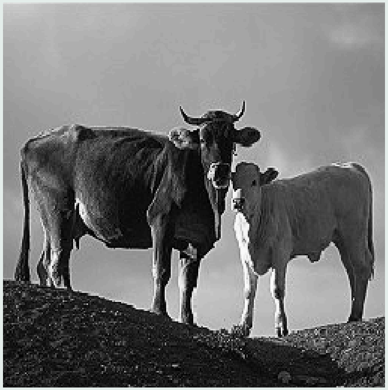

Figure 2: Image of two cows.

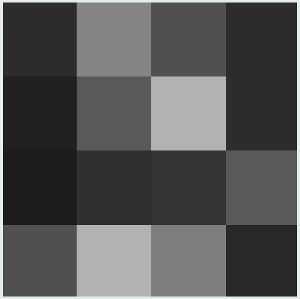

Figure 3: Magnified Image Showing Individual Pixels

Figure 4: Cow Image With Numerical Shade Comparison

represented by higher numbers and dark shades are represented by lower numbers.

The image of the cows in Figure 2, having dimensions 256×256 pixels, can be represented by a 256×256 matrix. A 256×256 matrix has 65,536 different entries. That's 65,536 different pixels represented in a single matrix. In order to understand how to apply linear algebra to these matrices, start small with an 8×8 sample matrix.

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

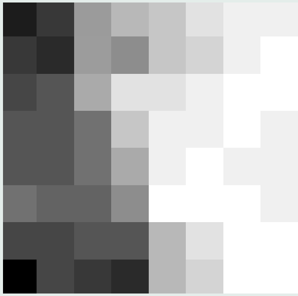

Figure 5: eight by eight image

This 8×8 matrix represents an image having dimensions 8×8 pixels, for a total of 64 pixels. Each pixel is a different shade of grey. Again, dark colors are represented by low numbers (such as 576 in the top left corner), and light colors are represented by high numbers (such as 1600 in the lower right corner). Note the correlation between numbers in this 8×8 matrix and pixel shades in Figure 5.

2. "Averaging and Differencing" (The Harr Wavelet Transform)

Now apply "Averaging and Differencing" (otherwise known as "The Harr Wavelet Transform"), by Colm Mulcahy, Ph.D, to the 8×8. To understand "Averaging and Differencing" strip off the first row of the 8×8 matrix. Now form a new row by averaging each pair of numbers in the original row. This will yield a new row only half the length of the original row. Fill the remaining positions by subtracting the averages from the corresponding first element of each pair. Continue this process until all the original numbers are averaged down into one number. The remaining numbers will be subtraction differences also called

Home Page
Title Page
Page 5 of 17
Go Back
Full Screen
Close
Quit

"detail coefficients."

$$\begin{bmatrix} 576 & 704 & 1152 & 1280 & 1344 & 1472 & 1536 & 1536 \\ 640 & 1216 & 1408 & 1536 & -64 & -64 & -64 & 0 \\ 928 & 1472 & -288 & -64 & -64 & -64 & -64 & 0 \\ 1200 & -272 & -288 & -64 & -64 & -64 & -64 & 0 \end{bmatrix}$$

Notice that with this 1×8 row, three steps are needed to complete the process.

This is the idea of "Averaging and Differencing." To complete this process on the 8×8 matrix, though, the process must be applied to every row and then to every column of the new matrix. This would require repeating the previous operations 15 times. This is a lot of work, and of course linear algebra simplifies the process greatly.

Imagine an 8×8 matrix that could perform these operations for us. The following matrix will actually complete the first step of our process for each row.

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

Refer to the original matrix as P, and the new matrix as A_1. By multiplying matrix P on the right by matrix A_1 the first step is completed for each row. Notice that multiplying our original first row by the the matrix A_1 yields the same results as shown before.

$$\begin{aligned} &(576, 704, 1152, 1280, 1344, 1472, 1536, 1536)A_1 \\ &\qquad = (640, 1216, 1408, 1536, -64, -64, -64, 0) \end{aligned}$$

A similar 8×8 matrix will perform the second step to each row. It will take the averages and differences of the left side of the rows and leave the right sides (detail coefficients) unchanged. Thinking in terms of block multiplication, a new matrix is easily constructed.

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

Note the similarity between matrix A_2 and matrix A_1. Also notice the differences. Particularly the identity matrix that is found in lower right. This is the portion of the matrix that leaves the detail coefficients unchanged. Carrying on from our previous example this point is illustrated:

$$(640, 1216, 1408, 1536, -64, -64, -64, 0)A_2 \\ = (928, 1472, -288, -64, -64, -64, -64, 0)$$

A third and last 8×8 matrix will complete the averaging and differencing process for the rows from the original matrix P. This last matrix, A_3, will take the average and difference

of the remaining two entries and leave the detail coefficients unchanged.

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

Again, note the size of the identity matrix in the lower right. The larger size makes sense because there are more elements in the rows are to be left unchanged. Again carrying through with the example the point is illustrated.

$$(928, 1472, -288, -64, -64, -64, -64, 0)A_3 \\ = (1200, -272, -288, -64, -64, -64, -64, 0)$$

The Averaging and Differencing will be complete when the original matrix P is multiplied on the right by A_1, A_2, and A_3. Repeat the process on the columns of the resulting matrix by multiplying on the left by A_1^T, A_2^T, and A_3^T. This process, although quicker than the original, still involves a lot of plugging and chugging. Here again linear algebra simplifies the mathematics.

Home Page

Title Page

◀◀ ▶▶

◀ ▶

Page 8 of 17

Go Back

Full Screen

Close

Quit

By multiplying A_1, A_2, and A_3 together, a new matrix W is created.

$$W = A_1 A_2 A_3 = \begin{bmatrix} 1/8 & 1/8 & 1/4 & 0 & 1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & 1/4 & 0 & -1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & 1/2 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & -1/2 & 0 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & 1/2 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & -1/2 & 0 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & 1/2 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & -1/2 \end{bmatrix}$$

The matrix W will perform the same operations as A_1, A_2, and A_3, but will greatly simplify this process. Similarly, the transpose of matrix W will be equal to the product of A_1^T, A_2^T, and A_3^T. So, by multiplying the original matrix P by W on the right and W^T on the left the Averaging and Differencing process is completed and a new matrix T is created.

$$T = W^T P W \tag{1}$$

Applying this process to matrix P produces the new transformed matrix T:

$$T = W^T P W = \begin{bmatrix} 1212 & -306 & -146 & -54 & -24 & -68 & -40 & 4 \\ 30 & 36 & -90 & -2 & 8 & -20 & 8 & -4 \\ -50 & -10 & -20 & -24 & 0 & 73 & -16 & -16 \\ 82 & 38 & -24 & 68 & 48 & -64 & 32 & 8 \\ 8 & 8 & -32 & 16 & -48 & -49 & -16 & 16 \\ 20 & 20 & -56 & -16 & -16 & 32 & -16 & -16 \\ -8 & 8 & -48 & 0 & -16 & -16 & -16 & -16 \\ 44 & 36 & 0 & 8 & 80 & -16 & -16 & 0 \end{bmatrix}$$

Notice that the top left entry represents an overall average, and the other entries are all detail coefficients.

2.1. Implementing Thresholds

Equation (1) creates a new matrix T. Using the following method matrix P is reconstructed from T.

$$\begin{aligned} T &= W^T P W \\ (W^T)^{-1} T W^{-1} &= (W^T)^{-1} W^T P W W^{-1} \\ (W^T)^{-1} T W^{-1} &= I P I \end{aligned}$$

This leads to the following reconstruction of matrix P.

$$(W^T)^{-1} T W^{-1} = P \tag{2}$$

Clearly equation (2) merely un-does the operations done by equation (1). However, this will not achieve the desired results. In lieu of using matrix T in equation (2), replace it with a close approximation matrix, N. This matrix N is constructed by implementing a threshold (replacing every element in T whose absolute value is less than or equal to a specified value with zero) on matrix T. Consider again, matrix T.

$$T = \begin{bmatrix} 1212 & -306 & -146 & -54 & -24 & -68 & -40 & 4 \\ 30 & 36 & -90 & -2 & 8 & -20 & 8 & -4 \\ -50 & -10 & -20 & -24 & 0 & 73 & -16 & -16 \\ 82 & 38 & -24 & 68 & 48 & -64 & 32 & 8 \\ 8 & 8 & -32 & 16 & -48 & -49 & -16 & 16 \\ 20 & 20 & -56 & -16 & -16 & 32 & -16 & -16 \\ -8 & 8 & -48 & 0 & -16 & -16 & -16 & -16 \\ 44 & 36 & 0 & 8 & 80 & -16 & -16 & 0 \end{bmatrix}$$

Implement a threshold of 50 (let 0 replace every number in matrix T whose absolute value

Home Page
Title Page
Page 10 of 17
Go Back
Full Screen
Close
Quit

is less than or equal to 50)

$$N = \begin{bmatrix} 1212 & -306 & -146 & -54 & 0 & -68 & 0 & 0 \\ 0 & 0 & -90 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 72 & 0 & 0 \\ 82 & 0 & 0 & 68 & 0 & -64 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -56 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80 & 0 & 0 & 0 \end{bmatrix}$$

3. Brief Notes on Storage

Outside the scope of this project is the storage procedures of computers. As it turns out computers store images in the form of matrix T and matrix N. When an image is called up on a computer the stored matrix is reconstructed using the process derived in equation (2) to form an image. As previously mentioned these matrices are extremely large (the previous image contained 65,536 entries) and take up a lot memory. The method computers use to store the matrices such as T and N involves special algorithms. The result of this method is that matrices with a lot of zeros entries take up less memory than those containing more non-zero entries. In short: matrices like N take up less memory than matrices like T. Thus, the purpose of implementing a threshold on matrix T is to save memory. By dropping small differences that are not noticeable to the human eye, the image is slightly altered and space is saved. These new, less memory impactive, matrices can be reconstructed to produce close approximations of the original image.

4. Reconstructing An Image

As equation (2) shows, matrix P can be reconstructed very easily. If matrix N is substituted for matrix T a close approximation of matrix P will result. Thus:

$$(W^T)^{-1} N W^{-1} = R \tag{3}$$

The new approximation matrix R:

$$R = \begin{bmatrix} 670 & 670 & 1147 & 1137 & 1464 & 1464 & 1572 & 1572 \\ 670 & 670 & 1147 & 1137 & 1464 & 1464 & 1572 & 1572 \\ 614 & 614 & 1057 & 1339 & 1464 & 1464 & 1572 & 1572 \\ 726 & 726 & 945 & 1227 & 1464 & 1464 & 1572 & 1572 \\ 932 & 932 & 912 & 1176 & 1614 & 1614 & 1586 & 1586 \\ 932 & 932 & 912 & 1176 & 1614 & 1614 & 1586 & 1586 \\ 848 & 688 & 876 & 884 & 1314 & 1314 & 1558 & 1558 \\ 688 & 848 & 876 & 884 & 1314 & 1314 & 1558 & 1558 \end{bmatrix}$$

Although matrix R is an approximation of matrix P, the images are very similar. As mentioned previously, the differences between the reconstructed image and the original image are slight, and barely noticeable to a human eye. Keep in mind that these images are 8×8, a small portion of an actual image.

5. Applying The Haar Wavelet Transform To Full Size Images

Now that the Haar Wavelet Transform is understood for 8×8 matrices, it's time to apply these ideas to full size images. This is done by first "normalizing" (multiplying by $\sqrt{2}$)

Figure 6: Original Image On Left represented by matrix P, New Image On Right represented by matrix R

matrix A_1, matrix A_2, matrix A_3, and matrix W. The result is quite interesting.

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

By normalizing matrix A_1 a new matrix A_1 is created. This new matrix has the property that its transpose acts as its inverse. This happens because the columns are orthogonal to one another. With denominators of $\sqrt{2}$ the multiplication of A_1^T and A_1, creates an identity matrix. Thus, it may be stated that

$$A^T = A^{-1}.$$

When matrix A_2, matrix A_3, and matrix W are normalized the same properties arise. Therefore,

$$W^T = W^{-1}. \tag{4}$$

Now equation (2) can be simplified knowing that

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

This leads to the following result:

$$W T W^T = P \tag{5}$$

If a threshold is again implemented on matrix T, a new matrix N will again be constructed. Therefore equation (3) can also be re-written:

$$W N W^T = R \tag{6}$$

Matrix N still takes up less memory, and matrix R still is an approximation of matrix P. In order to apply the new matrix W to a full size image it must be as large as the matrix it will be multiplied by. With linear algebra any matrix W is found by creating large matrices similar to A_1 and following similar procedures to find A_2, A_3, A_4, ..., A_n, where the number n is determined by the size of the image. By multiplying these matrices together a new matrix W is created. The following 256×256 pixel images were generated using this procedure. Compare the compressed images to the original image. Pay attention to the change in quality as the threshold increases; when threshold is small–quality is retained, when threshold is large–quality suffers.

References

[1] Mulcahy, Colm, *Image Compression Using The Harr Wavelet Transform*, Spelman Science and Math Journal, pp. 22-30

Home Page
Title Page
◀◀ ▶▶
◀ ▶

Go Back
Full Screen
Close
Quit

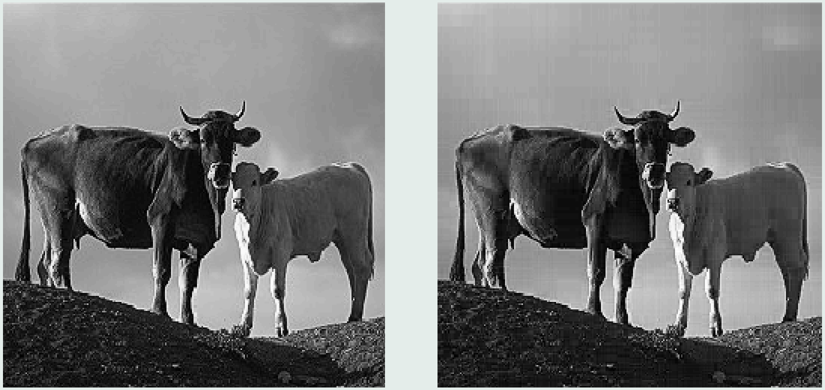

Figure 7: Cow Image With Threshold=20

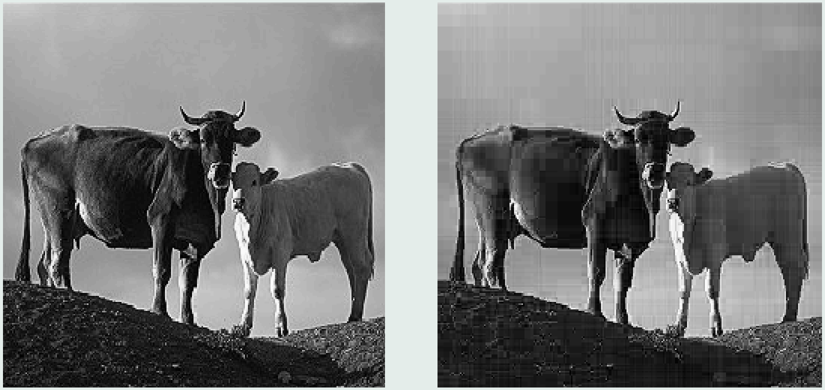

Figure 8: Cow Image With Threshold=40

Figure 9: Cow Image With Threshold=80

Figure 10: Cow Image With Threshold=200