Image Compression Using the Harr Wavelet Transform

LiHong Huang Herman

December 13, 2001

Abstract

We encode the information of a picture into a big matrix, so that we can transfer the information to the users. If the information is so big that it requires a large storage of the memory system, we can transfer the main information to the user, so that they can understand what the picture is like, by using a technique called Haar wavelet transform. The clearness of the picture is based on the approximation of the quantity of the information we transfer. That is what we call the levels of detail. We will focus on the black-and-white pictures.

1. Digitalize the information

The PC's cannot see what we see from the pictures. We have to digitalize the pictures by using a scanner or a digital camera, so that they will understand what we understand. Each image is presented mathematically by a matrix of numbers. Digitalizing a image from 0 (presenting black) to some positive whole number, such as 256 or 65536 (presenting white), the image has been transformed into the computer language.

For example, Figure 1 is presented mathematically by a 128×128 matrix(array) of numbers.

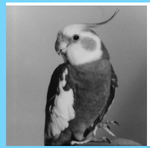
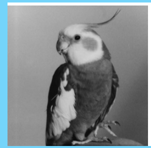


Figure 1:

2. Treat the digit

There is a method for transforming a string of data. It is called averaging and differencing. For example, consider a 8×8 matrix A extracted from a matrix P that we are going to treat. P is the matrix presented mathematical for a picture. By using MATLAB, such as,

```
A=floor(64*rand(8))
pcolor(A)
colormap(gray(64))
```



we can easily form a 8 × 8 matrix A,

$$A = \begin{bmatrix} 60 & 52 & 59 & 8 & 28 & 53 & 19 & 24 \\ 14 & 28 & 58 & 12 & 59 & 1 & 12 & 55 \\ 38 & 39 & 26 & 12 & 29 & 43 & 12 & 54 \\ 31 & 50 & 57 & 38 & 26 & 24 & 43 & 37 \\ 57 & 58 & 3 & 17 & 54 & 53 & 19 & 31 \\ 48 & 47 & 22 & 12 & 33 & 32 & 34 & 57 \\ 29 & 11 & 52 & 0 & 12 & 45 & 9 & 52 \\ 1 & 25 & 0 & 47 & 43 & 27 & 44 & 41 \end{bmatrix}$$

Figure 2 stands for this matrix.

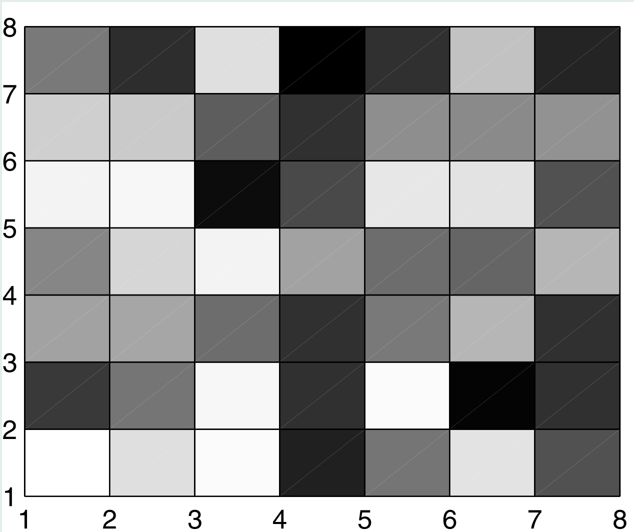
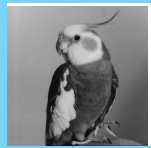


Figure 2:

That is our original data. Extract the first row of A, and name it A^*.

$$A^* = \begin{bmatrix} 60 & 52 & 59 & 8 & 28 & 53 & 19 & 24 \end{bmatrix}$$



Digitalize the information
Treat the digit
Linear Algebra is good
Compression–helps us . . .
The pictures
Home Page
Title Page
Page 3 of 16
Go Back
Full Screen
Close
Quit

We can think of it as four pairs of numbers. They are 60 and 52, 59 and 8, 28 and 53, and 19 and 24.

The second step is to average each pair and put them into the first four entries of a row E that has the same size as A^*. Let B be the first four entries of E, that is,

$$B = \begin{bmatrix}(60+52)/2 & (59+8)/2 & (28+53)/2 & (19+24)/2\end{bmatrix} = \begin{bmatrix}56 & 33.5 & 40.5 & 21.5\end{bmatrix}$$

Now, by extracting the first number of each pair from A^*, we form a row C, which is,

$$C = \begin{bmatrix}60 & 59 & 28 & 19\end{bmatrix}$$

Let D be $D = C - B$, and put D into the last four entries of E, to obtain, $E = [B, D]$, which is,

$$E = \begin{bmatrix}56 & 33.5 & 40.5 & 21.5 & 4 & 25.5 & -12.5 & -2.5\end{bmatrix}$$

Apply the same method to B, to obtain a row F, which is,

$$F = \begin{bmatrix}44.75 & 31 & 11.25 & 9.5\end{bmatrix}$$

Replace the first four entries of E by the entries in F, to obtain new row E,

$$E = \begin{bmatrix}44.75 & 31 & 11.25 & 9.5 & 4 & 25.5 & -12.5 & -2.5\end{bmatrix}$$

Applying the same method above, we have

$$E = \begin{bmatrix}37.875 & 6.875 & 11.25 & 9.5 & 4 & 25.5 & -12.5 & -2.5\end{bmatrix}$$

The numbers we obtain from the subtraction are called detail coefficients.(We will discuss the detail coefficient in section 4.) By summarizing each step, we have

60	52	59	8	28	53	19	24
56	33.5	40.5	21.5	4	25.5	-12.5	-2.5
44.75	31	11.25	9.5	4	25.5	-12.5	-2.5
37.875	6.875	11.25	9.5	4	25.5	-12.5	-2.5

Clearly, the process can be generalized to strings of any length: strings of length 2^k require k steps

Home Page
Title Page
◀◀ ▶▶
◀ ▶
Page 4 of 16
Go Back
Full Screen
Close
Quit

of averaging and differencing. We treat each row of the matrix, to obtain a totally different matrix,

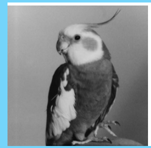
$$\begin{bmatrix} 37.88 & 6.88 & 11.25 & 9.50 & 4.00 & 25.50 & -12.50 & -2.50 \\ 29.88 & -1.88 & -7.00 & -1.75 & -7.00 & 23.00 & 29.00 & -21.50 \\ 31.63 & -2.88 & 9.75 & 1.50 & -0.50 & 7.00 & -7.00 & -21.00 \\ 38.25 & 5.75 & -3.50 & -7.50 & -9.50 & 9.50 & 1.00 & 3.00 \\ 36.50 & -2.75 & 23.75 & 14.25 & -0.50 & -7.00 & 0.50 & -6.00 \\ 35.63 & -3.38 & 15.25 & -6.50 & 0.50 & 5.00 & 0.50 & -11.50 \\ 26.25 & -3.25 & -3.00 & -1.00 & 9.00 & 26.00 & -16.50 & -21.50 \\ 28.50 & -10.25 & -5.25 & -3.75 & -12.00 & -23.50 & 8.00 & 1.50 \end{bmatrix}$$

Not only this, we also treat each column of the matrix the same manner after we treated each row. Let us call this semifinal matrix T, whose rows and columns have be treated.

$$T = \begin{bmatrix} 33.06 & -1.47 & 5.16 & 0.59 & -2.00 & 8.19 & 0.38 & -9.94 \\ 1.34 & 3.44 & -2.53 & -0.16 & -1.25 & 8.06 & 2.25 & -0.56 \\ -0.53 & 0.53 & -0.50 & 3.44 & 1.75 & 8.00 & 5.63 & -1.50 \\ 4.34 & 1.84 & 11.81 & 3.13 & 0.75 & -1.13 & 2.38 & 0.63 \\ 4.00 & 4.38 & 9.13 & 5.63 & 5.50 & 1.25 & -20.75 & 9.50 \\ -3.31 & -4.31 & 6.63 & 4.50 & 4.50 & -1.25 & -4.00 & -12.00 \\ 0.44 & 0.31 & 4.25 & 10.38 & -0.50 & -6.00 & 0.00 & 2.75 \\ -1.13 & 3.50 & 1.13 & 1.38 & 10.50 & 24.75 & -12.25 & -11.50 \end{bmatrix}$$

We call this procedure wavelet transform. The point of the wavelet transform is that regions of little variation in the original data manifest themselves as small or zero element in the wavelet transformed version. We see that, comparing the original matrix and the last matrix, the data has became smaller. We can use MATLAB to see what happened to the original picture. We have to download a program called halfinmat and halfwavmat to create w.

```
w=halfwavmat
T=w'*A*w
pcolor(T)
colormap(gray(64))
```



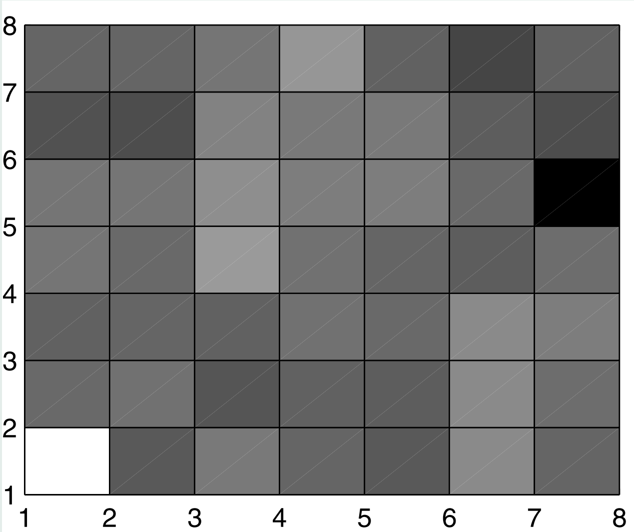


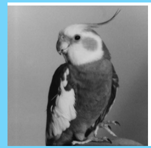
Figure 3:

Since the data has become smaller, it is easy for a PC to transform and store the information. In addition, the most important thing is that the treated information is reversible. To explain this reversing process, we need linear algebra.

3. Linear Algebra is good

Recall the summarization is section 2,

60	52	59	8	28	53	19	24
56	33.5	40.5	21.5	4	25.5	−12.5	−2.5
44.75	31	11.25	9.5	4	25.5	−12.5	−2.5
37.875	6.875	11.25	9.5	4	25.5	−12.5	−2.5



Digitalize the information
Treat the digit
Linear Algebra is good
Compression–helps us . . .
The pictures
Home Page
Title Page
Page 6 of 16
Go Back
Full Screen
Close
Quit

Again, we are going to use the first row for example. There exists a matrix S_1 such that,

$$\begin{bmatrix}56 & 33.5 & 40.5 & 21.5 & 4 & 25.5 & -12.5 & -2.5\end{bmatrix}$$
$$= \begin{bmatrix}60 & 52 & 59 & 8 & 28 & 53 & 19 & 24\end{bmatrix} S_1$$

According to the matrix multiplication rule, multiplying a matrix on the right, think column operation [1]. To obtain B (refers to section 2), we need to average each pair and put them into the first four entries of a row E that has the same size of A^*. To obtain the first column of S_1, s_1, average the first two numbers of A and zeros out the rest of the numbers.The result is,

$$s_1 = \begin{bmatrix}1/2\\1/2\\0\\0\\0\\0\\0\\0\end{bmatrix}$$

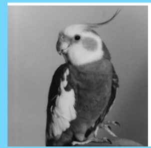
Do the same for the second column of S_1, s_2. This averages the second pair numbers and zeros out the rest of the numbers, so the result is,

$$s_2 = \begin{bmatrix}0\\0\\1/2\\1/2\\0\\0\\0\\0\end{bmatrix}$$

Do the same to obtain s_3 and s_4.

To obtain the fifth column s_5, subtract the first entry of A^* from the first entry of B to form the fifth entry of E. If we use the example in section 2,

$$60 - \frac{60+52}{2} = \frac{2}{2}60 - \frac{60+52}{2} = \frac{60-52}{2}$$



Therefore, s_5 should be

$$s_5 = \begin{bmatrix} 1/2 \\ -1/2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

By applying the same idea to s_6, s_7 and s_8, we construct S_1,

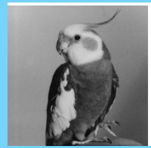
$$S_1 = \begin{bmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 \end{bmatrix}$$

Let S_2 be a matrix, such that,

$$\begin{aligned} &\begin{bmatrix} 44.75 & 31 & 11.25 & 9.5 & 4 & 25.5 & -12.5 & -2.5 \end{bmatrix} \\ &= \begin{bmatrix} 56 & 33.5 & 40.5 & 21.5 & 4 & 25.5 & -12.5 & -2.5 \end{bmatrix} S_2 \end{aligned}$$

Obviously, S_2 should be,

$$S_2 = \begin{bmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



and, for the third step, S_3 should be,

$$S_3 = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that, firstly, the columns have 1 keeps the detail coefficients; secondly, each column is orthogonal. Recall that, if $q_1, q_2, \cdots, q_k$ is an orthogonal collection of nonzero vectors, then they are independent. Since the columns of the S's are evidently orthogonal to each other with respect to the standard dot product, each of these matrices is invertible. Recall that, A^* is the first row extract from matrix A. Therefore, we have this matrix T, which is obtained by from treating only the rows of A,

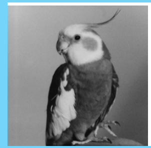
$$T = AS_1S_2S_3$$

Since S's are invertible, we have,

$$A = TS_3^{-1}S_2^{-1}S_3^{-1}$$

That means the treated information is reversible. Furthermore, we treat the columns as the same as the rows of matrix P. Then we will obtain a matrix T which has been averaged and diferenced through rows to columns (we will see the equations at section 4).

However, if we a have huge amount information to treat, we have to have many S's to deal with them. Finding the inverse of S's is easy by using MATLAB, but it will take quite a lot of memory and time. Remember that orthonormal is good[2]. If we can turn each column into orthonormal, we will save a lot of time and memory. Recall that, with an orthogonal matrix, Q, $Q^T = Q^{-1}$. That means we don't have to find the inverse of each S's, if they are orthonormal, and taking the transpose of the S's is enough. If we exchange the $\pm 1/2's$ into $\pm 1/\sqrt{2}'s$, we are done. We have found the orthonormal columns. We call this process normalization modification. Most pictures you will see in the paper are treated by normalization modification method.



4. Compression–helps us to transform easier

Consider the row vector we discussed above

$$A = \begin{bmatrix} 60 & 52 & 59 & 8 & 28 & 53 & 19 & 24 \end{bmatrix}$$

and the row vector

$$E = \begin{bmatrix} 37.875 & 6.875 & 11.25 & 9.5 & 4 & 25.5 & -12.5 & -2.5 \end{bmatrix}$$

If we just transformed E, and do nothing, we just did half of the game. Think about the numbers which are close to zero. You make them to be zero. It will make the storing and transferring process way easier. (Think about a 100×100 matrix with all the components are zero) In addition, we don't need a very precise version of a picture, but a approximated picture, which is good enough to let our perception to be stimulated. Let us compare these pictures (Figure 4 and Figure 5).

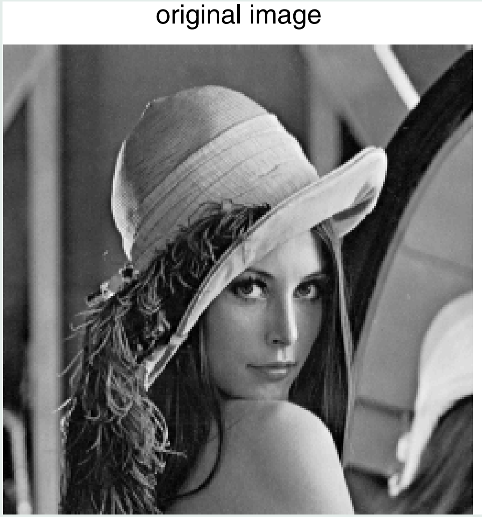


Figure 4:

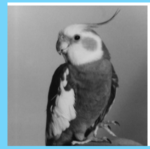
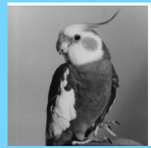


Figure 5:

The first one is the original picture, and the second one has been compressed. The definition of wavelet compression is, fix a nonnegative threshold value ϵ, and decree that any detail coefficient in the wavelet transformed data whose magnitude is less than or equal to ϵ will be reset to zero, then rebuild an approximation of the original data using this doctored version of the wavelet transformed data [3]. The threshold for the second picture is 1. Even though we reset the numbers (they are less than or equal zero) to be zero, which means we lost the original data, the picture is still good. That is because we set an appropriate threshold to zero the lower coefficient out. If we go back to section 2 and section 3, we find the average of a pair of numbers, then subtract the average from the first number of that pair. The result is the detail coefficient. That is, the low detail coefficient means that there is not much different in neighboring pixels, such as pure white gradually change to light yellow. Think about a pair of numbers don't have a big difference, then the coefficient will be very small. High differences mean that neighboring pixels jumped from light to dark, or vice verse. The larger the absolute value of the detail coefficient, the higher the difference, and the shaper the changing of the color. For the previous



matrix T we mentioned,

$$T = \begin{bmatrix} 33.06 & -1.47 & 5.16 & 0.59 & -2.00 & 8.19 & 0.38 & -9.94 \\ 1.34 & 3.44 & -2.53 & -0.16 & -1.25 & 8.06 & 2.25 & -0.56 \\ -0.53 & 0.53 & -0.50 & 3.44 & 1.75 & 8.00 & 5.63 & -1.50 \\ 4.34 & 1.84 & 11.81 & 3.13 & 0.75 & -1.13 & 2.38 & 0.63 \\ 4.00 & 4.38 & 9.13 & 5.63 & 5.50 & 1.25 & -20.75 & 9.50 \\ -3.31 & -4.31 & 6.63 & 4.50 & 4.50 & -1.25 & -4.00 & -12.00 \\ 0.44 & 0.31 & 4.25 & 10.38 & -0.50 & -6.00 & 0.00 & 2.75 \\ -1.13 & 3.50 & 1.13 & 1.38 & 10.50 & 24.75 & -12.25 & -11.50 \end{bmatrix}$$

If we set the threshold to be 5, the matrix T will become a new matrix T_n. The MATLAB code is,

```
myT=T;
k=find(abs(myT)<5);
myT(k)=0;
myT
A_n=inv(w')*(myT)*inv(w)
pcolor(A_n)
colormap(gray(64))
```

myT is the matrix has been compressed.

$$myT = \begin{bmatrix} 33.06 & 0.00 & 5.16 & 0.00 & 0.00 & 8.19 & 0.00 & -9.94 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 8.06 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 8.00 & 5.63 & 0.00 \\ 0.00 & 0.00 & 11.81 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 9.13 & 5.63 & 5.50 & 0.00 & -20.75 & 9.50 \\ 0.00 & 0.00 & 6.63 & 0.00 & 0.00 & 0.00 & 0.00 & -12.00 \\ 0.00 & 0.00 & 0.00 & 10.38 & 0.00 & -6.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 10.50 & 24.75 & -12.25 & -11.50 \end{bmatrix}$$

A_n is the matrix obtained from the reversing steps.

$$A_n = \begin{bmatrix} 52.84 & 41.84 & 43.03 & -5.47 & 23.56 & 53.81 & 27.00 & 27.88 \\ 23.59 & 34.59 & 61.28 & 12.78 & 53.81 & 1.06 & 19.25 & 58.13 \\ 44.84 & 44.84 & 29.53 & 13.03 & 27.44 & 38.69 & 11.13 & 55.00 \\ 31.59 & 31.59 & 42.78 & 26.28 & 27.44 & 38.69 & 35.13 & 31.00 \\ 50.03 & 50.03 & 10.22 & 21.97 & 43.44 & 43.44 & 12.75 & 32.63 \\ 50.03 & 50.03 & 22.22 & 9.97 & 22.69 & 22.69 & 33.50 & 53.38 \\ 36.91 & 15.91 & 64.59 & 14.84 & 20.81 & 45.31 & 11.63 & 54.50 \\ 15.91 & 36.91 & 15.09 & 64.34 & 45.31 & 20.81 & 34.63 & 31.50 \end{bmatrix}$$

Compare Figure 6, the figure after compression, with Figure 1.

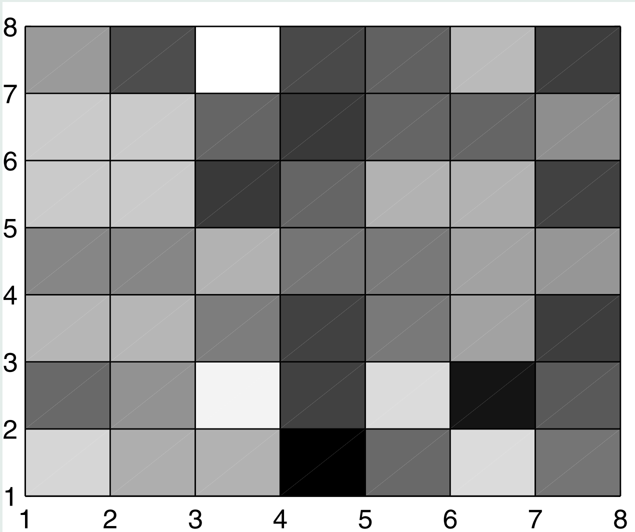
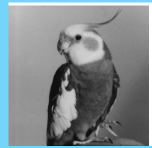


Figure 6:

Now we have new steps. First, we averaging and differencing (the orthogonal matrices are better) the whole matrix.



Second, we wavelet compress the whole matrix.

Third, we obtain the approximated picture by using the inverse of the S's, or using the transposes of the S's (if the S's are orthogonal matrices).

Here is the equation to summarize what we did.

$$S_1 S_2 S_3 = W \tag{1}$$

$$T = ((AW)^T W)^T = W^T AW \tag{2}$$

Remember we are not only treat the rows, but also the columns.

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

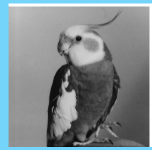
We should realize how important the role play by the orthonormal columns in here. It takes no effort to take the inverse of W, if S's are orthogonal.

If we approximated, or wavelet compressed the matrix, we just need to change the third equation's T into the matrix T^* we got after the compressed, and change matrix A into a approximation matrix A^* we will satisfy with.

5. The pictures

Theory is still theory till we show the power of it.

Figure 7 is the original picture. Figure 8 is a picture has been treated by the normalization modification method and the threshold is 1. We can simply download a program called project1 to have some fun.



Home Page
Title Page
Page 14 of 16
Go Back
Full Screen
Close
Quit

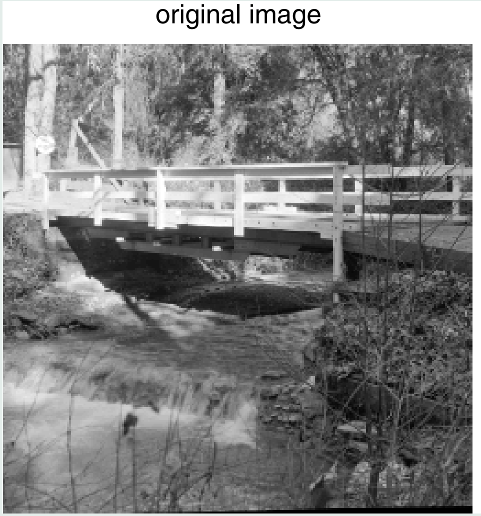
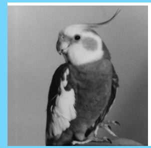


Figure 7:



Digitalize the information
Treat the digit
Linear Algebra is good
Compression–helps us . . .
The pictures

Home Page

Title Page

Page 15 of 16

Go Back

Full Screen

Close

Quit

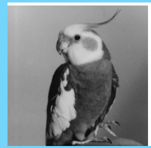
Figure 8:

References

[1] David Arnold *2001 class notes*

[2] Gilbert Strang *1998 Introduction To Linear Algebra*

[3] Colm Mulcahy *1996 Image Compression Using the Harr Wavelet Transform*



Home Page

Title Page

Page 16 of 16

Go Back

Full Screen

Close

Quit