

# Daubechies Wavelet Based Compression and Denoising Techniques for Biometric Images

Ajay Krishnan N. and Anil Kishore

**Abstract**—Finger print readers and retinal eye scanners have been an integral aspect of research and space organisations, forensic tests and so on. But to use it on a daily basis, Image compression is a necessary aspect considering the huge population of our country. Wavelet transform provides a platform for this process. In this paper, a novel idea to implement wavelet transform based on Daubechies wavelets has been discussed. The paper also discusses the concepts of linear transform, quantization, dequantization and image denoising. The image is first transformed using discrete wavelet transform to obtain the wavelet coefficients. The process of image compression is achieved by quantizing and encoding the coefficients. Uniform scalar quantization is done on the basis function coefficients in order to reduce their precision by reducing the number of bits required to store the transformed coefficients. The resulting quantized coefficients are then encoded using Huffman entropy encoding. In order to ensure that the image is free from disturbances like noise signals, denoising technique has been implemented using the wavelet coefficients. A paper of such an application has been proposed to tackle the global phenomenon of terrorism through the use of biometric devices including finger print reader and retinal eye scanner.

**Index Terms**—DWT, WSQ, encoder, quantization, denoising, huffmann encoding.

## I. INTRODUCTION

The worrying factor for all countries in recent times is the raise in Terrorism. Even though the intelligence network is much stronger in our country still the terrorists are attacking. One of the methods is to keep a regular check on the number of people entering and leaving the country i.e. traveling through airplane, ships or railways. Hence, a census of the normal people and the suspects can help in combating terror by taking actions at the right time. The various scanning and tracking equipments include finger print technology, thermal sensors, retinal eye scanners and even camera lens images. Retina scanners work using the reflection of infrared light off the retina. As infrared light is directed into the eye, some of the light is absorbed, while other light is reflected back to the scanner. The blood vessels on the retina absorb more light, reflecting less light than the tissue surrounding the vessels. The retinal scanner measures the amount and intensity of the light that is reflected back off the eye in 320 places. It then assigns an intensity grade between zero and 4,095. The resulting numbers are compressed into an 80-byte computer code. This code can then be compared with patterns that have already been entered into the computer's data base. Each time the eye is scanned, the code is

compared to other codes in the database to verify identity. Retina scanning is one of the most accurate biometric security measures available, because the retina patterns are difficult to fake and even the retinas of a dead person will change soon after they have died. Retina scans are so accurate in discovering identity that the identity error rate is estimated to be only one in every million.

The

Important security equipment is the finger print recorder. On an average, if the finger print instruments are used in places of employment, airports, railway stations, shipyards, etc, it would amount to nearly 1000 million cards of fingerprints, of which a few percent belongs to the usual suspects (bad guys tend to get fingerprinted more than once). And to make matters worse, fingerprint data continues to accumulate at a rate of 30,000-50,000 new cards PER DAY. India's fingerprint database can be digitized at 500 dots per inch with 8 bits of grayscale resolution. At this rate, a single fingerprint card turns into about 10 MB of data. When this gets multiplied by the 1000 million people, it comes to around 10,000 terabytes. Similarly, a single image of a retinal eye image turns into approximately 5 MB of data.

Hence, there is a need for some form of data compression. A wavelet-based image coding algorithm can be adopted as a national standard for digitized fingerprint records. The WSQ (Wavelet/Scalar Quantization) utilized for this purpose involves:

- 2-dimensional discrete wavelet transform DWT
- Uniform scalar quantization
- Huffman entropy coding

## II. THEORY OF WAVELETS

A wave is an oscillating function of time or space and is periodic. In contrast, wavelets are localized waves. They have their energy concentrated in time or space and are suited to analysis of transient signals. The transform of a signal is just another form of representing the signal. It does not change the information content present in the signal. The Wavelet Transform provides a time-frequency representation of the signal. The Wavelet Transform, at high frequencies, gives good time resolution and poor frequency resolution, while at low frequencies; the Wavelet Transform gives good frequency resolution and poor time resolution.

### A. The Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT), which is based on sub-band coding, is found to yield a fast computation of Wavelet Transform.

In CWT, the signals are analyzed using a set of basis functions which relate to each other by simple scaling and translation. In the case of DWT, a time-scale representation

of the Odigital signal is obtained using digital filtering techniques. The signal to be analyzed is passed through filters with different cutoff frequencies at different scales.

### III. WAVELET TRANSFORMATION USING DAUBECHIES WAVELETS

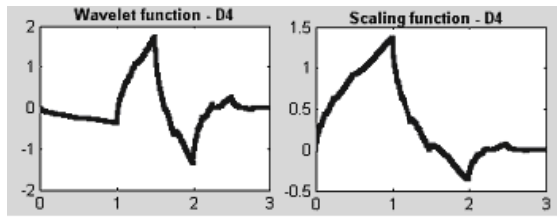


Fig. 1. Shows the daubechies wavelet and scaling functions

The objective is to perform the discrete Daubechies wavelet transformation on an image for the purpose of compression. The Daubechies wavelet transformation is composed of a sequence of low-pass and high-pass filters, known as a filter bank. These filter sequences can be applied in the same way as a discrete FIR filter in the DSP.

Daubechies D4 scaling coefficients

$$c(1)= 0.4830, c(2)= 0.8365, c(3)= 0.2241, c(4)= -0.1294$$

Daubechies D4 low pass filter coefficients

$$\text{reverse}(\sqrt{2}) \times (c(1), c(2), \dots, c(n))$$

$$c(1)= 0.4830, c(2)= 0.8365, c(3)= 0.2241, c(4)= -0.1294$$

High Pass Filter Coefficients are obtained by reversing the low pass filter coefficients and then negating the odd coefficients.

$$d=\{d(0),d(1),d(2),d(3)\}=\{c(3),-c(2),c(1),-c(0)\}$$

$$T = \underbrace{W_N A W_N^T}$$

First half: Applying 1D Transformation to Rows

Second half: Applying 1D Transformation to Columns

where A is the matrix representing the 2D image pixels, T is the Daubechies wavelet transformation of the image, and

The result of the complete transformation, T, is composed of 4 new sub-images, which correspond to the blurred image, and the vertical, diagonal, and horizontal differences between the original image and the blurred image. The blurred representation of the image removes the details (high frequency components), which are represented separately in the other three images, in a manner that produces a sparser representation overall, making it easier to store and transmit.

The two filters combined form a so-called *filter bank*. The idea is to use the following example as a test example for the code you will be asked to write and use later on in this lab. Since our filter now has length 4, we need to decide what to do when our filter encounters a boundary point. This example will periodically extend the signal.

Consider a single row of a 16×16 image whose 8 bits are represented as follows

$$f=\{f(0),f(1)\dots f(7)\} = (2, 5, 8, 9, 7, 4,-1,1)$$

We wish to expand this signal in the basis defined by the filter bank above. In practice, we do this by performing the following steps: Step 1: Extend the signal periodically. We could also do an even extension of the signal or pad with some constant. What we choose to do at the boundary depends on the application and the signal. If we expect our signal to come from some “periodic process”, a periodic extension makes sense. We denote the extended signal with a tilde f.

$$f=\{f(6), f(7), f(0), f(1), f(2), f(3), f(4), f(5), f(6), f(7)\}=(1,1,2,5,8,9,7,4,-1,1)$$

Now move the low and high pass filters along this vector, two steps at a time (shifting by two steps comes from the fact that we are down sampling the signal)

$$f_1 = \begin{pmatrix} c(0)f(6) + c(1)f(7) + c(2)f(0) + c(3)f(1) \\ c(0)f(0) + c(1)f(1) + c(2)f(2) + c(3)f(3) \\ c(0)f(2) + c(1)f(3) + c(2)f(4) + c(3)f(5) \\ c(0)f(4) + c(1)f(5) + c(2)f(6) + c(3)f(7) \\ d(0)f(6) + d(1)f(7) + d(2)f(0) + d(3)f(1) \\ d(0)f(0) + d(1)f(1) + d(2)f(2) + d(3)f(3) \\ d(0)f(2) + d(1)f(3) + d(2)f(4) + d(3)f(5) \\ d(0)f(4) + d(1)f(5) + d(2)f(6) + d(3)f(7) \end{pmatrix}$$

$$= (0.155, 5.78, 12.4, 6.37, -0.837, 0.966, 0.871, -3.12)$$

Step 2: For the next step, we keep the last half of the vector f<sub>1</sub> fixed while we low- and high pass filter the first half of the vector. In order to do this, it is necessary to periodically extend the first half of the vector f<sub>1</sub>. = (12.4, 6.37, 0.155, 5.78, 12.4, 6.37,-0.837, 0.966, 0.871,-3.12). Now move the low and high pass filters along the first six elements of this vector, two steps at a time:

$$\tilde{f}_1 = (f_1(2), f_1(3), f_1(0), f_1(1), f_1(2), f_1(3), f_1(4), f_1(5), f_1(6), f_1(7))$$

$$f_2 = \begin{pmatrix} c(0)f_1(2) + c(1)f_1(3) + c(2)f_1(0) + c(3)f_1(1) \\ c(0)f_1(0) + c(1)f_1(1) + c(2)f_1(2) + c(3)f_1(3) \\ d(0)f_1(2) + d(1)f_1(3) + d(2)f_1(0) + d(3)f_1(1) \\ d(0)f_1(0) + d(1)f_1(1) + d(2)f_1(2) + d(3)f_1(3) \\ f_1(4) \\ f_1(5) \\ f_1(6) \\ f_1(7) \end{pmatrix}$$

$$= (10.6, 6.87, -5.70, 6.02, -0.837, 0.966, 0.871, -3.12)$$

Note that we didn't touch the last four elements! Step 3: In the last step we only low- and high pass filter the first quarter of the vector f<sub>2</sub>. First we extend the first two elements periodically:

$$f_3 = \begin{pmatrix} c(0)f_2(0) + c(1) + c(2)f_2(0) + c(3)f_2(1) \\ f_2(2) \\ f_2(3) \\ f_2(4) \\ f_2(5) \\ f_2(6) \\ f_2(7) \end{pmatrix}$$

$$= (12.4, 2.65, -5.70, 6.02, -0.837, 0.966, 0.871, -3.12)$$

$$f_2=\{f_2(0), f_2(1), f_2(0), f_2(1), f_2(2), f_2(3), f_2(4), f_2(5), f_2(6), f_2(7)\} = (10.6,6.87,10.6,6.87,-5.70,6.02,-0.837,0.966,0.871,-$$

3.12). As the final step we act with the filter on the first four elements of this vector:

We find that the recovered sequence is approximately equal to the original sequence.

$$F_3 = (12.4, 2.7, -5.7, 6, -0.8, 0.97, 0.88, -3.1)$$

The inverse transformation can be applied to  $T$ , resulting in lossless compression. Lossy compression can be implemented by manually setting to zero the elements below a certain threshold in  $T$ . The equation of the inverse transformation is:

$$\tilde{A} = W_N^{-1} T W_N^{-T} \quad (\tilde{A} = A \text{ after lossless compression})$$

In assembly, we implemented the matrix multiplication using a filter representation of the process. We first applied the filter  $G$  and  $H$  over the rows and then the columns as an FIR filter. While performing this step we use a technique called down sampling to prevent redundancies. To perform this process, we throw away every other result and keep only the ones obtained from successive pairs of elements.

To perform the transformation in DSP we loaded the pixel values for the image into DSP memory, process the image in the DSP assembly program, and output the transformed image, or transformed-compressed image, to a memory location below the original image. This data would be processed again to recover the original image. To be more specific, our original image was loaded at location 0x1800, the intermediate result, which is the first half of the transformation is stored immediately below the original image at location 0x1C00. Finally, we store the final output directly below this at address 0x2000. Below is a memory map showing the addresses used in the memory for the intermediate result and final result of our output:

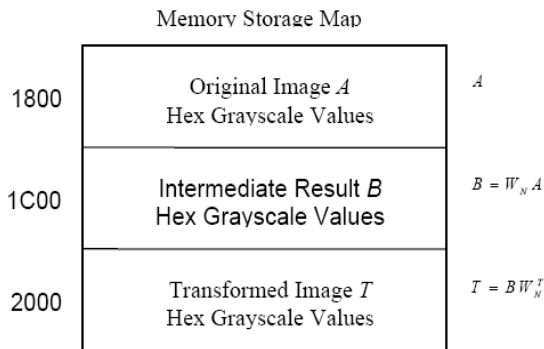


Fig. 4. Shows the memory storage map

#### IV. STEPS INVOLVED IN IMAGE COMPRESSION

Two fundamental components of compression are redundancy

Redundancy which aims at removing duplication from the signal source (image/video) and Irrelevancy reduction which omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS).

Image compression steps

A source encoder performs linear transform to decorrelate the original image data. This process is a lossless one.

Quantization of basis functions coefficients. This process is a lossy one.

Entropy coding of the resulting quantized values. This process is a lossless one.

A compressed image of the original image is obtained.

The compressed image is then subjected to the processes of decoding, dequantization and inverse transform to obtain a reconstructed image.

##### A. Linear Transformation

The coordinate system in which a signal is represented is changed in order to make it much better suited for processing (compression). All the useful signal features and important phenomena should be represented in as compact manner as possible. It is important to compact the bulk of the coefficients.

##### B. Process of Image Compression

Compression is achieved by quantizing and encoding coefficients

A quantizer simply reduces the number of bits needed to store the transformed coefficients by reducing the precision of those values. Since this is a many-to-one mapping, it is a lossy process and is the main source of compression in an encoder. Quantization can be performed on each individual coefficient, which is known as Scalar Quantization (SQ).

Coefficients that correspond to smooth parts of data become small. (Indeed, their difference, and therefore their associated wavelet coefficient, will be zero, or very close to it). So we can throw away these coefficients without significantly distorting the image. We can then encode the remaining coefficients and transmit them along with the overall average value.

At the encoder, the scalar quantization operation maps a given signal value to a quantizer index, which is then encoded as part of the compressed bit stream.

At the decoder, the quantizer index is decoded and converted into corresponding quantized value. This process is sometimes referred to as Dequantization.

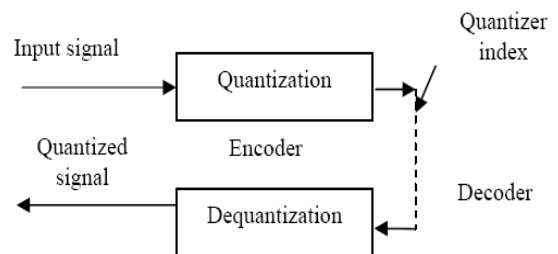


Fig. 5. Shows a block diagram representation of signal transfer

Once the quantization process is completed, the last encoding step is to use entropy coding to achieve the entropy rate of quantizer. An entropy encoder further compresses the quantized values losslessly to give better overall compression. It uses a model to accurately determine the probabilities for each quantized value and produces an appropriate code based on these probabilities so that the resultant output code stream will be smaller than the input stream. The most commonly used entropy encoders are the Huffman encoder and the arithmetic encoder.

##### C. Huffman Coding Algorithm

- Create a vertical table for the symbols, the best ordering being in decreasing order of probability.
- Form a binary tree to the right of the table. Build the tree by merging the two lowest probability symbols at each level, making the probability of the node equal to the sum of the merged nodes' probabilities.

- At each node, label each of the emanating branches with a binary number. The bit sequence obtained from passing from the tree's root to the symbol is its Huffman code.

D. Wavelet Denoising

- DWT of the image is calculated
- Resultant coefficients are passed through threshold testing
- The coefficients  $<$  threshold are removed, others shrunk
- Resultant coefficients are used for image reconstruction with IWT.
- The intuition behind this approach is that the neighboring pixels exhibit high correlation, which translates to only a few large wavelet coefficients. On the other hand, the noise is evenly distributed among the coefficients and is generally small.

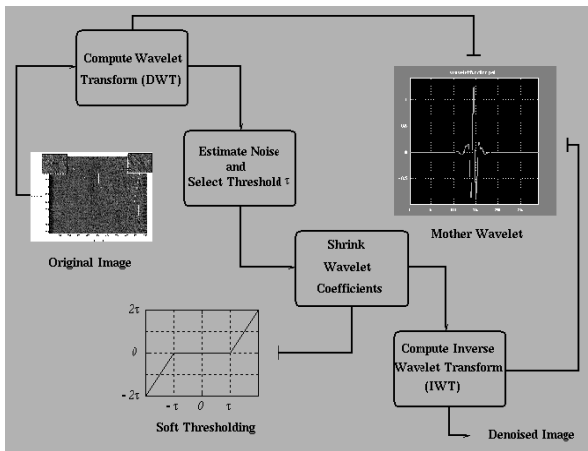


Fig. 6. Illustrates the concept of wavelet denoising

V. BLOCK DIAGRAM REPRESENTATION OF THE ENTIRE PROCESS

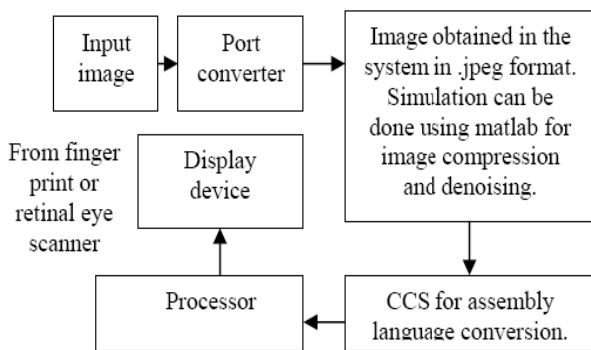


Fig. 7. Shows a block diagram representation of the entire process

The input image is obtained in a system using a port converter from a finger print reader or retinal eye scanner. The image is present in jpeg format. Simulation can be done in MATLAB to ascertain the process of image compression using wavelets in its various stages of encoding, quantization and dequantization and decoding. Coding is done using discrete wavelet transform coefficients. The MATLAB coding is then converted to assembly language using code composer studio. The image is then passed through the processor to perform the process of image compression. The compressed image can be viewed in a monitor. The image obtained will be compressed and will have good resolution.

VI. OBSERVATION

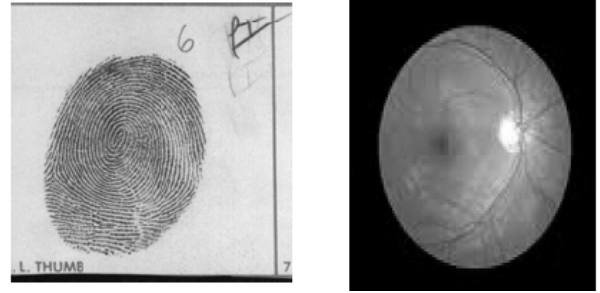


Fig. 8. Shows an Original finger print image (541832 bites 768x768) and a retinal eye scanner image

WSQ Vs JPEG

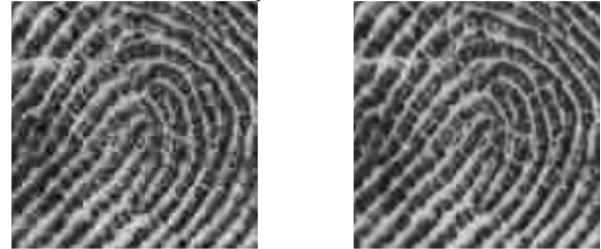


Fig. 9. Shows a WSQ image (file size 30987 bytes,

Compression ratio 19.0.) on the left and a JPEG image( file size 30081 bytes, compression ratio 19.6 ) on the right.

VII. COMPARISON BETWEEN HAAR AND DAUB 4 TRANSFORM

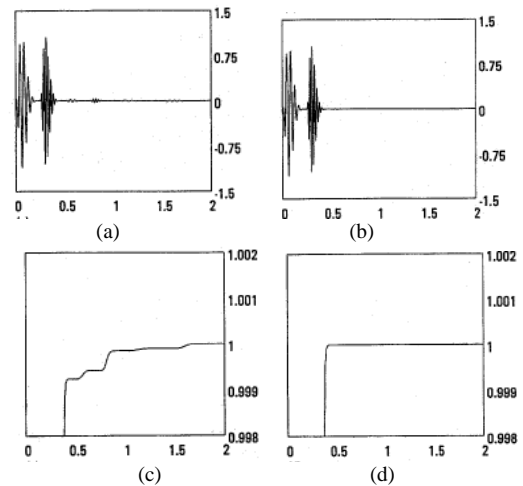


Fig. 10. (a) Graph of 2-level Haar transform of signal (b) Graph of 2-level Daub transform of signal (c) & (d) Cumulative energy profiles of the transforms

VIII. CONCLUSION

Wavelets provide an easier and better approach to image processing. Its use in the prevention of terrorism through the finger print technology and retinal eye scanners can be extended into a package containing thermal sensors, iris eye scanners and even camera lens images.

REFERENCES

[1] I. Daubechies, *Ten lectures on wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.  
 [2] Y. Nievergelt, *Wavelets Made Easy*, Birkhäuser, Boston, 1999.

- [3] Z. Xiong and K. Ramchandran. Wavelet Image Compression. [Online]. Available: <http://lena.tamu.edu/~zx/>
- [4] Daubechies, "Orthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 45, 1988.
- [5] S. Mallat, *A wavelet tour of signals processing*, Academic Press, 1998.
- [6] S. Betz, N. Bhagat, P. Murhy, and M. Stengler, "Wavelet Based Image Compression: Image Compression Theory," ELEC 301 Final Project.
- [7] S. Gergic, K. Kers, and M. Grgic, "Image Compression Using Wavelets," *ISIE99*, vol. 48, no. 3, pp. 682-695, June 2001.



**Ajay Krishnan Nilakantan** was born in Neyveli, Tamil Nadu, India on 14<sup>th</sup> November 1989. He completed his under graduation in Electrical and Electronics Engineering from St. Joseph's College of Engineering, Chennai, Tamil Nadu, India in the year 2011 with a distinction. He is currently working in Larsen & Toubro Limited-construction Division, Manapakkam, Chennai, Tamil Nadu, India. He has attended several Workshops on DSP. His under graduation project focused on "Analysis of Harmonics in Power Systems using Wavelets. He has presented a paper titled "Haar Wavelet Based Compression and Denoising Techniques for Biometric images" at an International Conference hosted by Nirma University, Ahmedabad, Gujarat, India on December 10<sup>th</sup>, 2010. ". His prime area of Focus is the use of Wavelets in diverse fields of Engineering like Images, Power Systems.



**Anil Kishore** was born in Thiruvananthapuram, Kerala, India on 23<sup>rd</sup> March, 1989. He completed his under graduation in Electrical and Electronics Engineering from St. Joseph's College of Engineering, Chennai, Tamil Nadu, India in the year He is currently pursuing his Masters degree in Business Administration from the University of Toledo, Ohio. He has presented papers in various National and International Conferences, a prime example of which is his presentation at the International Conference in Singapore (ICSAP) where he was applauded for his creative and a thought provoking style of presentation at such a young age. Mr. Kishore is a member of IE and is a team player, dedicated, versatile, object oriented and has an eye for quality.