Journal of Engineering and Development, Vol. 10, No. 4, December (2006) ISSN 1813-7822

Digital Image Compression Using Fourier Transform and Wavelet Technique

Asst. Lect. Mothana Amaar Hassan
Laser & Optoelectronic Engineering Dept.
University of Technology, Baghdad, Iraq

Asst. Lect. Razi G. Al- Azawi
Laser & Optoelectronic Engineering Dept.
University of Technology, Baghdad, Iraq

Asst. Lect. Sabah Abdul-Hassan Gittaffa
Electrical and Electronic Engineering Department
University of Technology, Baghdad, Iraq

Abstract

Fourier analysis and wavelet analysis have often been used in time series analysis. Fourier analysis can be used to detect periodic components that have sinusoidal shape. However, it might be misleading when the periodic components are not sinusoidal. The resulting Fourier analysis is more difficult to interpret compared with classical Fourier analysis. Wavelet analysis is very useful in analyzing and describing time series with gradual frequency changes. Wavelet analysis also has a shortcoming by giving no exact meaning to the concept of frequency because wavelets are not periodic functions. In addition, the two analysis methods above require equally-spaced time series observations.

In this paper, by using a sequence of periodic step functions, a new analysis method, adaptive Fourier analysis, and its theory are developed. These can be applied to time series data where patterns may take general periodic shapes that include sinusoids as special cases. Most importantly, the resulting adaptive Fourier analysis does not require equally-spaced time series observations. To do statistic test for periodic components, the adaptive Fourier analysis are needs. People can approximate a function by polynomials, sinusoid, step functions, and we used wavelets and apply the approximate to nonparametric regression, nonparametric regression by the step function.

In this work we used adaptive Fourier analysis would be applied for many image to compared data observation will be take fewer degree of foredoom.

الخـلاصـــــة

إن تحليل فورير وتحليل تقنية المويجة في اغلب الأحيان تستعمل في تحليل دالة زمنية . أن تحليل فورير يمكن أن يستعمل لاكتشاف المكونات الدورية التي لها الشكل الجيبي، ولكن عندما تكون المكونات غير جيبية فأن الناتج يكون مظلل. إن تحليل فورير كلاسيكي وأن تحليل تقنية المويجة مفيد جدا في وصف المتوالية الزمنية مع التردد التدريجي يكون متغير غير أن له عيبا بوجود أخطاء غير مضبوطة. ويقصد بمفهوم التردد لأن في تقنية المويجة تكون غير دورية، بالإضافة إلى أن الطريقتين تتطلب ملاحظات المتوالية الزمنية المتباعدة إلى حد سواء.

في هذا البحث سيتم تعريف نظرية جديدة سميت (بتحليل فورير المتطور) حيث يمكن أن يقدم بيانات زمنية متوالية في أنماط قد تتخذ أشكال دورية عامة التي تتضمن موجات جيبية للحالات الخاصة. والاهم من ذلك أن النتائج المحللة بطريقة فورير لا تتطلب فاصل زمني متتالي لملاحظتها، وكما أن فحص النماذج يتم بهذه الطريقة.

أن كثير من الطرق المعروفة من التحليل يكون اقل كفاءة من طريقة انحسار الأجزاء في تطبيق ***(Adaptive Fourier Analysis)*** *والتي طبقت على كثير من الصور في هذا العمل والتي كانت نتائجها أكثر دقة ودرجة اقل من الحرية وفاصل زمني قليل للمعالجة.*

1. Introduction

The digital cameras, scanners and camera phones have made the capture, display, storage and transmission of images, a routine experience. In addition, imaging is extensively used in medicine, law enforcement, Internet gaming and [1] data collected by satellites. Despite rapid improvements in data storage processing speeds, and digital communication system performance, this proliferation of digital media often outstrips the amount of data storage and transmission capacities. Thus, the compression of such signals has assumed great importance in the use, storage and transmission of digital images. For still images, the JPEG and the GIF standards have been the prevailing norms for lossy and lossless compression. Recently, wavelet-based lossy compression schemes have been gaining popularity over discrete cosine transform (DCT) due to their lower complexity and better image quality vis-a-vis compression ratio. For image compression applications, it is vital that a non-expansive (i.e. the total number of input samples is equal to the total number of wavelet coefficients at any point during the decomposition process) discrete wavelet transform (DWT) be employed; If orthogonal wavelets were able to employ symmetric extension, then perhaps their unique advantages (energy preserving, decorrelating, simple inverse) would outweigh advantages of the biorthogonal wavelets (linear phase). Thus, some unresolved questions may now be addressed. This paper presents and compares different DWT implementation techniques as well as compares the performance of orthogonal and biorthogonal wavelets with symmetric extension [2].

2. Theory of Data Analysis Method

2-1 Fourier Analysis

The basic idea of a Fourier series is that any function $x(t) \in L^2[0.T]$ can be decomposed into an infinite sum of cosine and sine functions:

$$x(t) = \sum_{k=0}^{\infty} [a_k \cos\frac{2\pi kt}{T} + b_k \sin\frac{2\pi kt}{T}] \quad \text{for all } t \quad \dots\dots\dots\dots (1)$$

where:

$$a_k = \frac{1}{T}\int_0^T x(t)\cos\frac{2\pi kt}{T}dt \quad \dots\dots\dots\dots (2)$$

$$b_k = \frac{1}{T}\int_0^T x(t)\sin\frac{2\pi kt}{T}dt \quad \dots\dots\dots\dots (3)$$

This is due to the fact that { $1, \cos\frac{2\pi kt}{T}, \sin\frac{2\pi kt}{T}, k = 1,2,3,...$ } form a basis for the space L^2 [0,T]. The summation in (1) is up to infinity, but *x*(t) can be well approximated in the L^2 sense by a finite sum with K cosine and sine functions:

$$x(t) = \sum_{k=0}^{\infty} [a_k \cos\frac{2\pi kt}{T} + b_k \sin\frac{2\pi kt}{T}] \quad \dots\dots\dots\dots (4)$$

This decomposition shows that *x(t)* can be approximated by a sum of sinusoidal shapes at frequencies $\lambda_k = 2\pi k / T, k = 0,1,..., k$. In addition, the variability in *x(t)* as measured by $\int_0^t |x(t)|^2 dt$ can be approximately partitioned into the sum of the variability of the sinusoidal shapes:

$$\int_0^T |x(t)|^2 dt = \int_0^T [\sum_{k=0}^{\infty} [a_k \cos\frac{2\pi kt}{T} + b_k \sin\frac{2\pi kt}{T}]^2 dt \equiv \sum_{k=0}^{k} |\gamma_k|^2 \quad \dots\dots\dots\dots (5)$$

A standard technique of time series analysis is to treat the partition in equation (5) as an analysis of variance (ANOVA) for identifying sinusoidal periodicities in a time series data set $\{ x(t), 0 < t \leq T\}$. When *x(t)* has sharp discontinuities or a non-sinusoidal waveform, such as a rectangular waveform, then we would require a very large number, K, of terms in its Fourier series in order to get an adequate approximation [3].

2-2 Discrete Fourier Transform (DFT)

For an arbitrary time series data set, $x = (x(t_1), x(t_2), \ldots, x(t_N))$, if the observation times are equally spaced, at interval Δt, then the data set x can be simply written as $x = (x(1), x(2), \ldots, x(N))$ by taking Δt =1 and $t_j = j$; and there is an orthogonal system $\{e^{-itw_k} : -\frac{N}{2}+1 \le k \le \frac{N}{2}$ if n is even and $-\frac{N-1}{2} \le k \le \frac{N-1}{2}$ if n is odd$\}$, so that the Discrete Fourier Transform of x can be defined by:

$$X^*(k) = \frac{1}{N}\sum_{t=1}^{N} x(t)e^{-itw_k} \quad \text{(6)}$$

where: the frequencies $w_k = 2\pi k/N$, k=0,1, …, [N/2], are called the Fourier frequencies; and [r] is the largest integer no larger than r. The Fourier series of $x(t)$ can be written as:

$$x(t) = \sum_{k=-[N(N-1)/2]}^{[N/2]} X^*(k)e^{-itw_k} \quad \text{(7)}$$

Corresponding to (5), we have the ANOVA

$$\sum_{t=1}^{N} x^2(t) = N \sum_{k=-[N(N-1)/2]}^{[N/2]} |X^*(k)|^2 \quad \text{(8)}$$

This representation provides an ANOVA for revealing how well the periodicities in x may be described by the sinusoidal shapes $X^*(-k)e^{itw_{-k}} + X^*(k)e^{itw_k}$. The ANOVA decomposition in (8) holds only if the DFT, $X^*(t)$, is evaluated only at a fixed set of N/2 equally-spaced frequencies w_k, and the data set must be equally-spaced.

2-3 Image Compression Using Fast Fourier Transform (FFT)

In the JPEG image compression algorithm, the input image is divided into 8-by-8 or 16-by-16 blocks, and the two-dimensional DCT is computed for each block. The DCT coefficients are then quantized, coded, and transmitted. The JPEG receiver (or JPEG file reader) decodes the quantized DCT coefficients, computes the inverse two-dimensional DCT of each block, and then puts the blocks back together into a single image. For typical images, many of the DCT coefficients have values close to zero; these coefficients can be discarded without seriously affecting the quality of the reconstructed image. The computes of two-dimensional DCT of 8-by-8 blocks in the input image is done by discards (sets to zero) all but 10 of the 64 DCT coefficients in each block; and then reconstructs the image using the two-dimensional inverse DCT of each block. Although there is some loss of quality in the reconstructed image, it is clearly recognizable, even though almost 85% of the DCT coefficients were discarded. To experiment with discarding more or fewer coefficients, and to apply this technique to other images [1]. The image compression is shown in **Fig.(1)**.

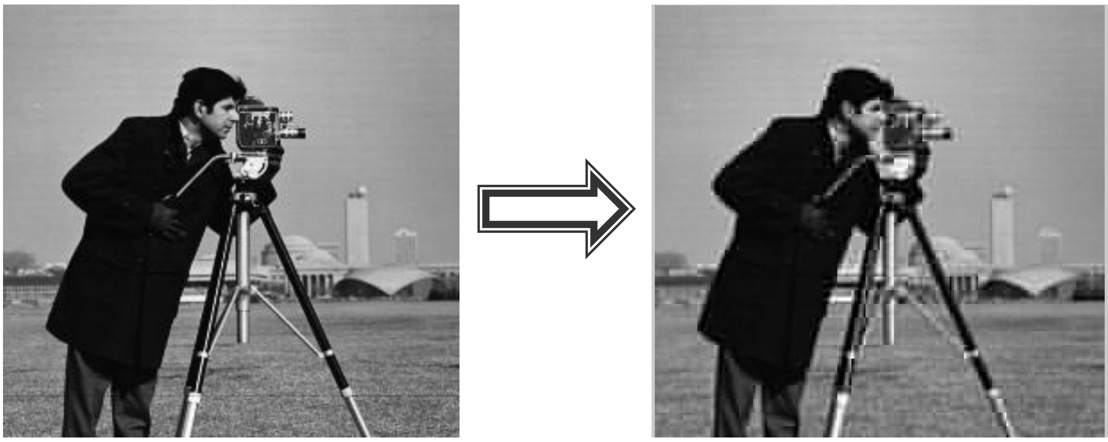

Image before compression **Image after compression**

Figure (1) The image before and after compression

2-4 Wavelet Analysis

Wavelet analysis is to decompose a given function x(t) $\in$ L2 into a sum of wavelet functions. It involves a mother wavelet ψ(t), which may be any real or complex continuous function that satisfies certain conditions, such as $\int_{-\infty}^{\infty} \psi(t)dt = 0$ and $\int_{-\infty}^{\infty} |\psi(t)|^2 \le \infty$. Wavelets are themselves derived from their mother wavelet ψ(t) by translations and dilations. The Haar function can be a mother wavelet defined by:

$$\psi(t)=\begin{cases}1 & \text{for } 0 \le t < 1/2 \\ -1 & \text{for } 1/2 \le t < 1 \\ 0, & \text{otherwise}\end{cases} \quad \text{.. (9)}$$

Another commonly used wavelet is Morlet wavelet defined as:

$$\psi(t)=e^{-t^2}\cos(\pi t\sqrt{2/\ln 2}) \cong e^{-t^2}\cos(2.885\pi t) \quad \text{................................ (10)}$$

Four different mother wavelets: Haar, Daublet, Symmlet, and Coiflet are shown in **Fig.(2)**, where the first letter of the wavelet indicates the name: d for Daublet, s for Symmlet, and c for Coiflet; the number of the wavelet indicates its width and smoothness [2]. Given a mother wavelet ψ(t), an infinite sequence of wavelets can be constructed by varying translations *b* and dilations *a* as below:

$$\psi_{a,b}(t) = |a|^{-1/2}\,\psi\left(\frac{t-b}{a}\right) \quad \text{.......... (11)}$$

By defining the continuous wavelet transform W(a,b) as:

$$W(a,b) = \langle x(t), \psi_{a,b}(t) \rangle = \int_{-\infty}^{\infty} x(t)\,\psi_{a,b}(t)\,dt \quad \text{.......... (12)}$$

We can represent *x(t)* as:

$$x(t) = \frac{1}{C_1}\int_{0}^{\infty}\int_{-\infty}^{\infty} a^{-2}\,W(a,b)\,\psi_{a,b}(t)\,da\,db \quad \text{.......... (13)}$$

where:

$$C_1 = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{\omega}\,dt \qquad \text{and} \qquad \Psi(\omega) = \int_{-\infty}^{\infty} \psi(t)e^{-i\omega t}\,dt$$

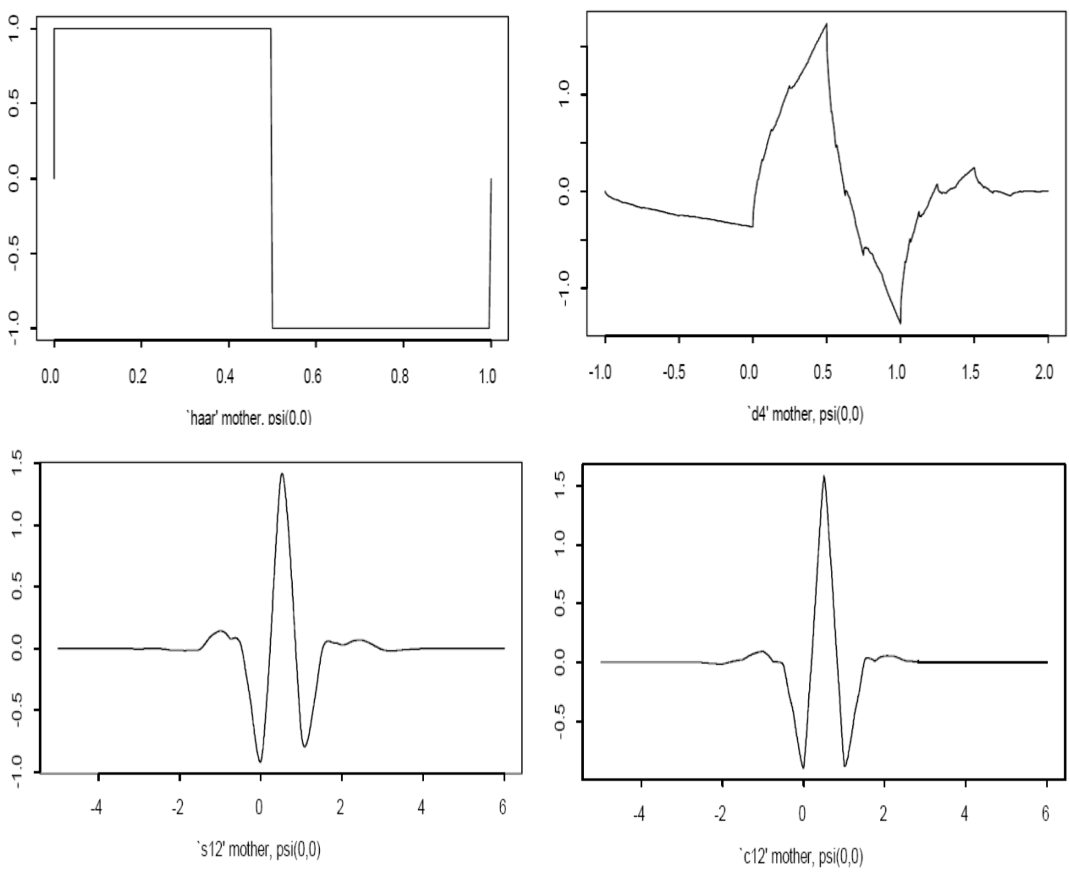

Figure (2) Four different mother wavelets

When *a* and *b* take on discrete sets of values, we can similarly obtain the discrete wavelet transform as:

$$W(m,n)=\langle x(t),\psi_{m,n}(t)\rangle=\int_{-\infty}^{\infty}x(t)\psi_{m,n}(t)\,dt \quad \text{.......................................} \quad (14)$$

and,

$$x(t)=\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}W_{m,n}\,\psi_{m,n}(t) \quad \text{...} \quad (15)$$

For an equally spaced time series data *x =(x(1), x(2), ..., x(N))*, we can take approximate wavelet transforms by replacing (13) by an estimate such as:

$$W(m,n)=\int_{-\infty}^{\infty}x(t)\psi_{m,n}(t)\,dt \approx \sum_{l=1}^{N}x(l)\psi_{m,n}(l) \quad \text{...} \quad (16)$$

It follows that a class of discrete wavelet transform (DWT) for equally spaced time series data can be implemented by using an efficient computational algorithm [2,3].

An example of wavelet is given in **Fig.(2)**. In the example, the signal is ss =10*cos (π*t/15)+3*cos(π*t/10), which is plotted in **Fig.(3)**.

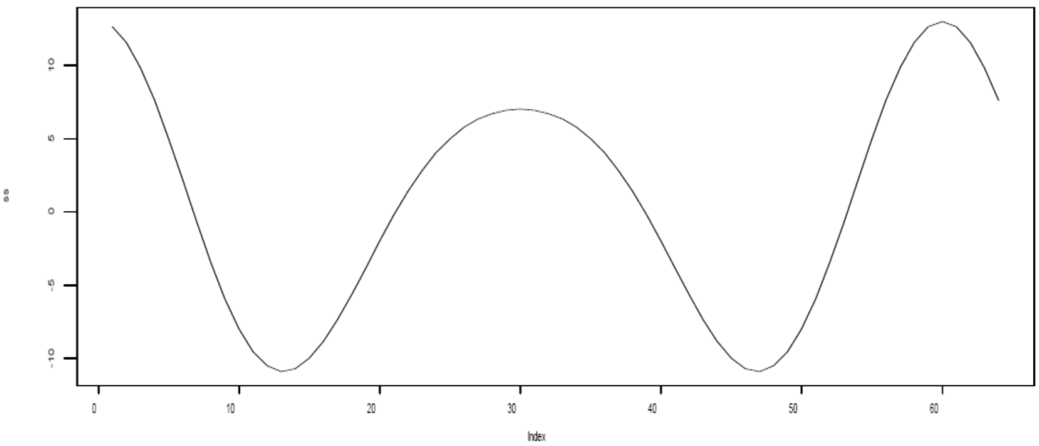

Figure (3) The signal ss =10*cos (π*t/15)+3*cos(π*t/10)

2-4-1 Image Compression Using Wavelet Transform

This research is based on the Image Wavelet Compression C code, which is provided by Honeywell Technology Center [4]. The IWC code contains all the routines required for a simple wavelet-based image compression of a 512x512 8-bit pixel image in PGM format. The compression process consists of four basic steps: wavelet transform, quantization, run-length encoding, and entropy coding. This is shown in **Fig.(4)**. The IWC code accepts an optional compression-factor parameter specifying how aggressively the image should be compressed. A compression factor indicates minimal compression and maximum image quality. The compression factor of 255 indicates maximum compression with higher degradation to the image quality. If the compression factor is not indicated, a default compression rate of 128 will be used. The compressed output image file format is specific to this program [4].

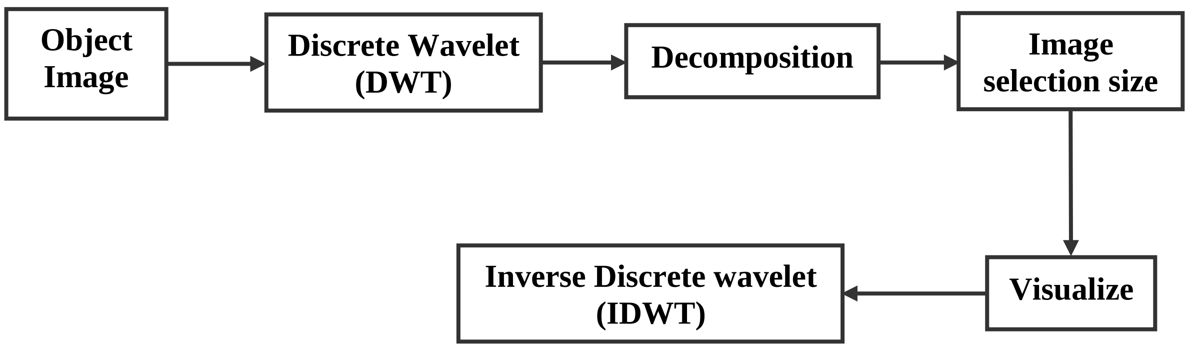

Figure (4) Image wavelet compression

2-4-2 Wavelet Transform Routine

The first step, the wavelet transform routine process, is a modified version of the biorthogonal Cohen-Daubechies-Feuvear wavelet. Wavelet transforms have received significant attention and are widely used for signal and image processing. For example, they are widely used in image coding, image compression, and speech discrimination. The basic concept behind wavelet transform is to hierarchically decompose an input signal into a series of successively lower resolution reference signals and their associated detail signals. At each level, the reference signal and the detail signal contain the information needed to reconstruct the reference signal at the next higher resolution level [5].

2-4-3 One-Dimensional Wavelet Transform

The one-dimensional discrete wavelet transform can be described in terms of a filter band as shown in **Fig.(5)**. An input signal $x[n]$ is applied to the low pass filter $l[n]$ and to the analysis high-pass filter $h[n]$. The odd samples of the outputs of these filters are then discarded, corresponding to a decimation factor of two. The decimated outputs of these filters constitute the reference signal $r[k]$ and the detail signal $d[k]$ for a new-level of decomposition. During reconstruction, interpolation by a factor of two is performed, followed by filtering using the low pass and high-pass synthesis filters *l[n]* and *h[n]*. Finally, the outputs of the two synthesis filters are added together [6].

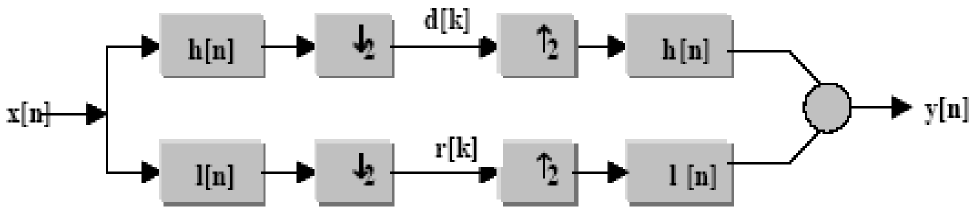

Figure (5) One-dimensional wavelet transforms

The above procedure can be expressed mathematically in the following equations:

$$d[k]=\sum_{n} x[n]\bullet h[2k-n] \qquad (17)$$

$$r[k]=\sum_{n} x[n]\bullet l[2k-n] \qquad (18)$$

$$x[n]=\sum_{n} \left(d[k]\bullet g[-n+2k]\right)+\left(r[k]\bullet h[-n+2k]\right) \qquad (19)$$

2-4-4 Multilevel Decomposition Wavelet Transform

For a multilevel decomposition, the above process is repeated. The previous level's lower resolution reference signal $r_i[n]$ becomes the next level sub-sampling input, and its associated detail signal $d_i[n]$ is obtained after each level filtering. **Figure (6)** illustrates this procedure. The original signal *x[n]* is input into the low-pass filter *l[n]* and the high-pass filter *h[n]*. After three levels of decomposition, a reference signal $r_3[n]$ with the resolution reduced by a factor of 2^3 and detail signals $d_3[n]$, $d_2[n]$, $d_1[n]$ are obtained. These signals can be used for signal reconstruction.

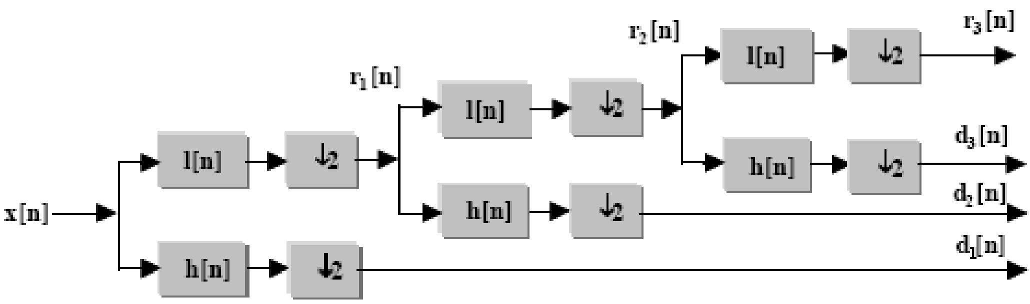

Figure (6) Three-level decomposition for wavelet transforms

The wavelet transform routine in the IWC code employs a shifting scheme to simplify the wavelet implementation. Therefore, it only requires integer adds and shifts, which make it easier to implement on hardware. The computation of the wavelet filter is performed according to the following equations.

$$\mathbf{D_0 = D_0 + D_0 - S_0 - S_0} \quad \text{.......... (20)}$$

$$\mathbf{S_0 = S_0 + (2 * D_0 / 8)} \quad \text{.......... (21)}$$

$$\mathbf{D_i = D_i + D_i - S_i - S_{i+1}} \quad \text{.......... (22)}$$

$$\mathbf{S_i = S_i + ((D_{i-1} + D_i) / 8)} \quad \text{.......... (23)}$$

In the above equations, D_i and S_i are odd and even pixels taken from one row or column. For every row or column do:

S_o D_o S_1 D_1 S_2 D_2 S_3 D_3 ... respectively. In image compression, one row or column of an image is regarded as a signal.

Calculation of the wavelet transform requires pixels taken from one row or column at a time. In Equations (20)$\rightarrow$(23). D_i should be calculated before processing S_i. Therefore, the odd pixel should be processed first, then the even pixel due to the data dependency. There are a total of three levels based on the 3-level decomposition wavelet transform algorithm discussed above. In each level, the rows are processed first then the columns. Each level's signal length (amount of each row/column pixels) is half of the previous level. Equations (20)$\rightarrow$(23) are grouped into a function called *image compression by MATLAB program.*

Figure (7) illustrates the three levels of wavelet transform implementation, and **Fig.(8)** shows compressed image by Haar class.

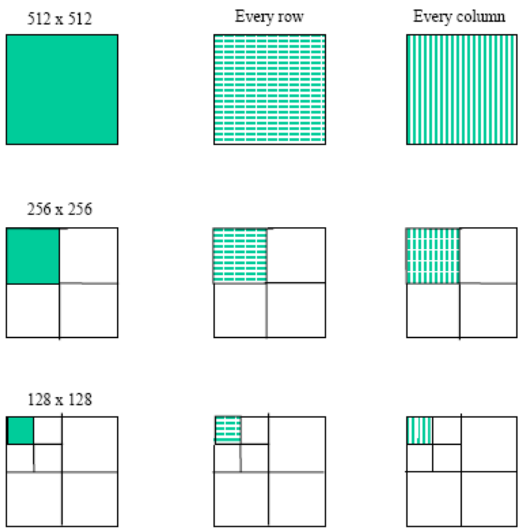

Figure (7) Wavelet transforms implementation

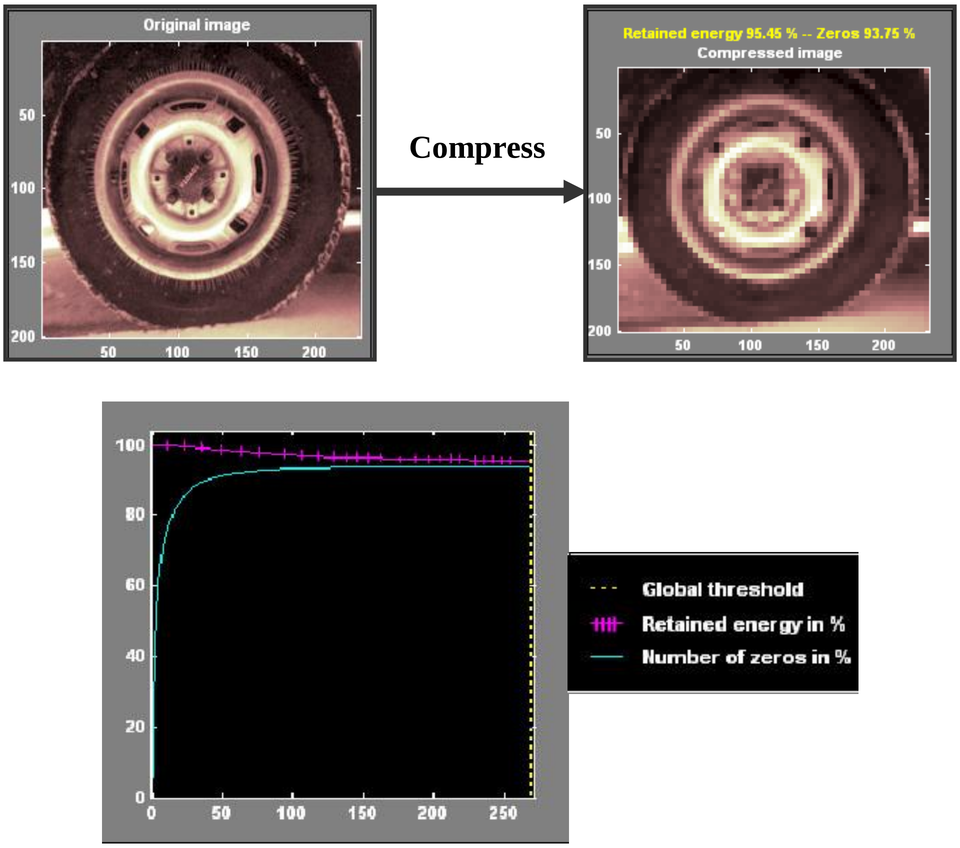

Figure (8) Image compression (200*200) analyzed at level 2 with Haar

2-5 Quantization Routine

After the three levels of the wavelet transform, the quantization routine follows. During the quantization routine, the original image is divided into 10 blocks; the first four will be 64 x 64 pixels (4096 pixels), then three will be 128 x 128 (16384 pixels), and the remaining three of 256 x 256 pixels (65536 pixels). Every block executes the same quantization process. **Figure (9)**, illustrates this as a block diagram. Before processing each block, some parameters should be prepared. First is the *blockthresh*, which should be provided by the developer. An array is used to hold these 10 block [7].

Blockthresh [10] = {0, 39, 27, 104, 79, 51, 191, 99999, 99999, 99999}

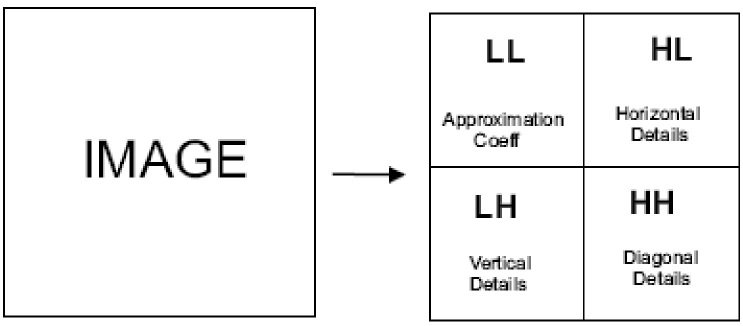

Figure (9) Output of 1-level 2-D decomposition

thresh1~thresh16; the formula to calculate these values are as follows:

$$\text{thresh}_n = \min + \frac{(\max - \min).\,n + 8}{2^4}, \qquad n:1\sim16 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(24)}$$

The $thresh_n$ is the *n*th thresh value in a block, and *n* value is from 1 to 16. Values *min* and *max* are the minimal and maximum pixel values within this block. After these numbers are computed, each block can run the quantization process. First, each input pixel's absolute value is compared with its corresponding *blockthresh [n]*, if it is smaller than the *blockthresh* value, the original pixel value is assigned to a constant value *ZERO_MARK*, which should be defined by the user. In this work, this is assigned a value of *16*. If the *abs (pixel)* value is not smaller than the *blockthresh* value, the pixel will be passed called "*image compression". The* original pixel value will then be changed into its corresponding $thresh_n$ value after this call. The pseudo code to represent the above calculation is shown in **Fig.(10)**.

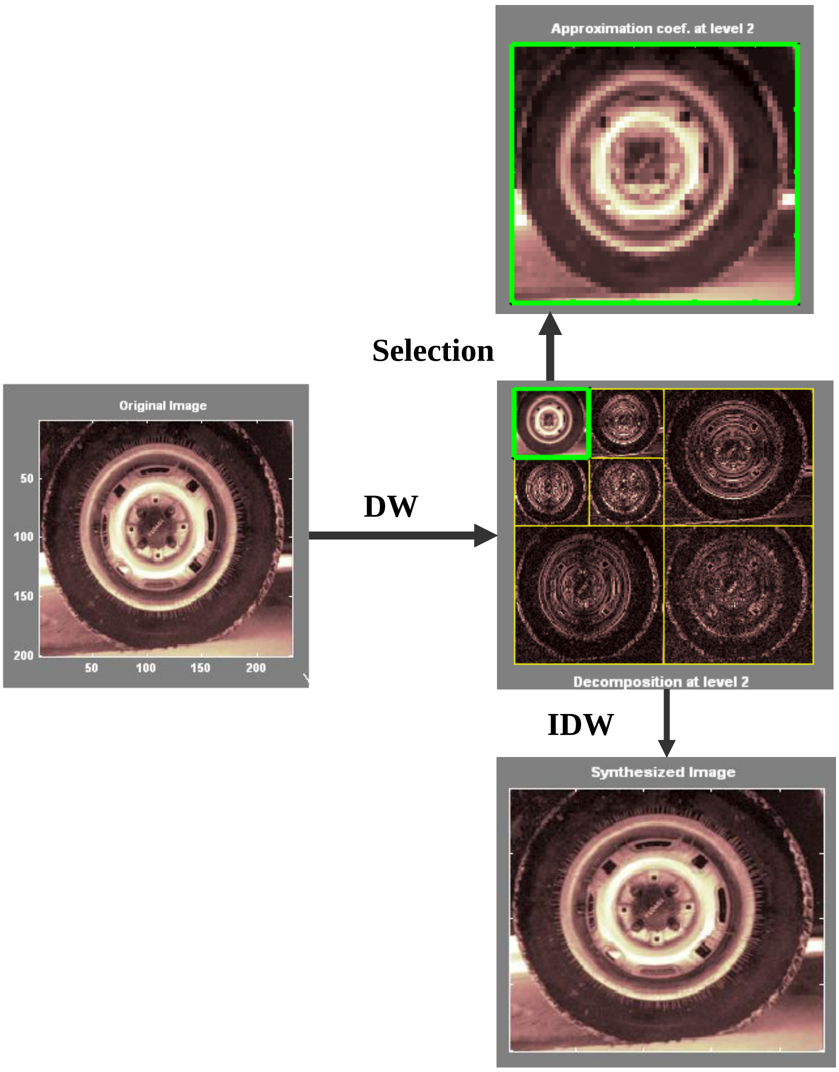

Figure (10) Represent compression by DWL

3. Conclusion

This work has identified and explained new parameters and techniques that are critical to high quality image compression performance. Although this work illustrates improved image compression performance for orthogonal wavelets, the develop theory and methods can be applied to equally and unequally-spaced time series in which the frequency components of time series may take general periodic shapes that include sinusoidal as special case.

4. References

1. M., Misiti, Y., Misiti, and J., Poggi, ***"Wavelet Toolbox for Use with MATLAB"***, Matlab Version 7.

2. Bruce Andrew, and Gao Hong-Ye, ***"Applied Wavelet Analysis with S-PL"***, Springer-Verlag Inc., New York.

3. M., Duelli, L., Ge, and R. W., Cohn, ***"Nonlinear Effect of Phase Blurring on Fourier Transform Holograms"***, J. Opt. Soc. Am 17, 2000, pp. 1594-1605.

4. Luiz Pires, and Deepa Pandalai, ***"Honeywell Technology Center"***, Minnepoli, MN 55412, "Compress", October 1988.

5. Christopher M. Brislawn, ***"Preservation of Subband Symmetry in Multirate Signal Coding"***, IEEE Trans. on Signal Proc., 43(12):3046{2050, December, 1995.

6. J., Villasenor, B., Belzer, and J., Liao, ***"Wavelet Filter: Evaluation for Image Compression"***, IEEE Transactions on Image Processing, Vol. 2, August 1995, pp. 1053-1060.

7. Stgabrata Rout, ***"Orthogonal vs. Biorthongonal Wavelets for Image Compression"***, Blacksburg, Verginia, 2003.