



Cairo University
Faculty of Engineering

Image Compression using Wavelets

Presented for MTH1172 project

Presented to

Dr. Samah El-Tantawy

TEAM MEMBERS (1st year electronics and communication)

مجدي احمد عباس عبد الحميد	ID: 9210899 / SEC: 3 / BN: 35
محمد ابراهيم محمد على	ID: 9210906 / SEC: 3 / BN: 36
محمد احمد عبد العظيم محمد	ID: 9210915 / SEC: 3 / BN: 38
محمد احمد عبد العظيم عبد الله	ID: 9210914 / SEC: 3 / BN: 37
محمد ايهاب عبد الرحمن	ID: 9210945 / SEC: 3 / BN: 42
فاروق هاشم سعيد عبد اللطيف	ID: 9210798 / SEC: 3 / BN: 16
مازن احمد عمر مصطفى عمر	ID: 9210887 / SEC: 3 / BN: 33
مازن وائل ضياء الدين احمد رأفت	ID: 9210892 / SEC: 3 / BN: 34

IMAGE COMPRESSION

Abstract

As mass media becomes an integral part of daily life, so does visual media. Optimizing how we store and transmit images is heavily related to the process of image compression. The prospect of processing images with reduced costs has been garnering a lot of attention over an extended period of time and more so recently. Our research aims to find an optimized compression algorithm using wavelets, which solves the two main issues which face the compression process. The first issue, which is the formation of “blocking artifacts”, was remedied by utilizing wavelets rather than other transforms like the DCT (Discrete Cosine Transform). The second issue of finding the ideal balance between image quality and compression was solved by utilizing the use of thresholding and setting the value of the threshold to the value of standard deviation of the image used. Using that method produces compression ratios up to 180 times greater than lossless compression while only losing 0.03% of the original signal’s energy.

Problem Definition

INTRODUCTION

As the commercial introduction of computer-based electronic digital cameras revolutionized photography in the 20th Century improving many features drastically, the introduction of image compression techniques such as the DCT, SVD (Singular Value Decomposition) and DWT (Discrete Wavelet Transforms) led to the proliferation of digital images, so much so that digital images have become a fixture in the online community.

Image compression and its standardization has become a topic of increasing importance as image processing systems and applications come of age. Continuing cost improvements in computing power, storage and communications are making more and more such systems practical, with compression almost always included to achieve cost-effective solutions. The objective of image compression is to reduce irrelevance and redundancy of the image data to be able to store or transmit data in an efficient form. It is concerned with minimizing the number of bits required to represent an image. Although bandwidth and storage are much cheaper than they used to be, they still cost money. And images typically contain a huge amount of useless redundancy. So, compressing images saves storage space and communication bandwidth with very little impact on image quality.

Improvement in photography techniques has led to major breakthroughs in terms of quality and more specifically resolution. As resolution increased from SD to HD and UHD allowing us to obtain more detailed images, so did the cost of storage and transmission (Bandwidth). Therein lies the problem, which is finding the balance between quality and storage cost that would suit the requirements of each field that requires image compression. So, we aim to find an optimized algorithm to face these issues and fully reap the benefits of improved storage size and bandwidth. We'll discuss the two main approaches exist to find that balance between image compression and quality which are: Lossy and Lossless compression.

LOSSY COMPRESSION

Lossy compression or irreversible compression [1] involves some loss of information, and data that has been compressed using lossy techniques generally cannot be recovered or reconstructed exactly. In return for accepting this distortion in the reconstruction, we can generally obtain very high compression ratios which is desirable especially in applications such as streaming media, internet telephony and television broadcasting, where lower bandwidths would be required to transfer more images. Ideally, the loss would be either minimal or undetectable by human observations. As certain techniques such as DCT target the frequencies to which the human eye is less sensitive, which allows the algorithm to reduce storage size substantially and maintain a respectable image quality simultaneously. Lossy methods are especially suitable for natural images such as photographs in applications in which minor loss of fidelity is acceptable to achieve a substantial reduction in bit rate. Through that, the highly efficient DCT compression method became the basis for JPEG which has become the most widely used image file format. It was largely responsible for the popularization of digital images with several billion JPEG images produced every day as of 2015.

The main issue with lossy compression in general and JPEGs specifically is the introduction of compression artifacts [2], which are discernible distortions of media generally and artificially generated media specifically as shown in *figure (1.1)*. These errors occur when heavy compression is applied or when the compression algorithm may not be intelligent enough to discriminate between distortions of little subjective importance and those objectionable to the user, which is less apparent in natural images as previously stated. In the applications in which lossy compression is used, these errors are tolerated to an extent in exchange for the reduced cost of storage and transmission. Many workarounds have been proposed to remedy these errors, most of them focus on “post-processing”, which is processing images when received or viewed. But none managed to improve image quality in all cases and as a result, none gained widespread acceptance.

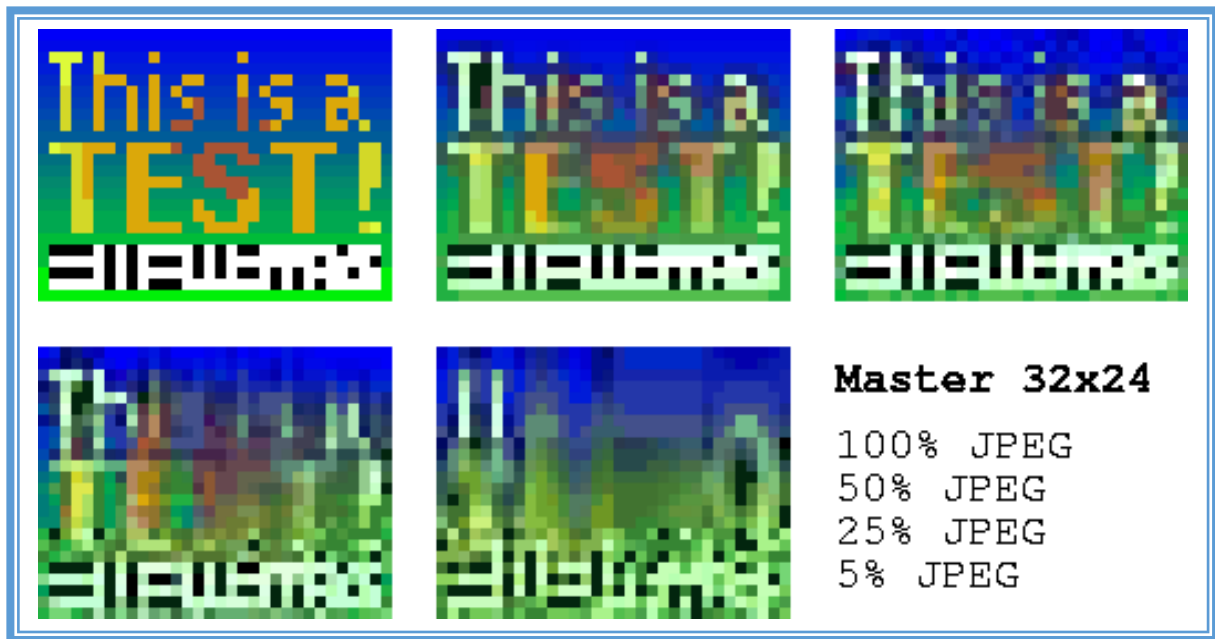


Figure (1.1) Compression artifacts in artificially generated image

LOSSLESS COMPRESSION

In fields where it is important that the original and the decompressed data be identical, or where deviations from the original data would be undesirable or disastrous.

Lossless compression [1] is used, which is a class of data compression methods that allows the original data to be perfectly reconstructed from the compressed data while obtaining compression ratios significantly lower than lossy compression. It is preferred for archival purposes and often for medical imaging, technical drawings, clip art, comics, remote sensing via satellite and for military communication systems through radars.

Ultimately, we will utilize one algorithm that can be used to work as both lossy and lossless. That image compression algorithm, which uses wavelets, remedies the issue of artifacts by working on the entire image rather than sections at a time which prevents “blocking artifacts” from damaging the images’ quality. Solving that issue, we’ll then focus on solving the issue of finding the right balance between image compression and quality.

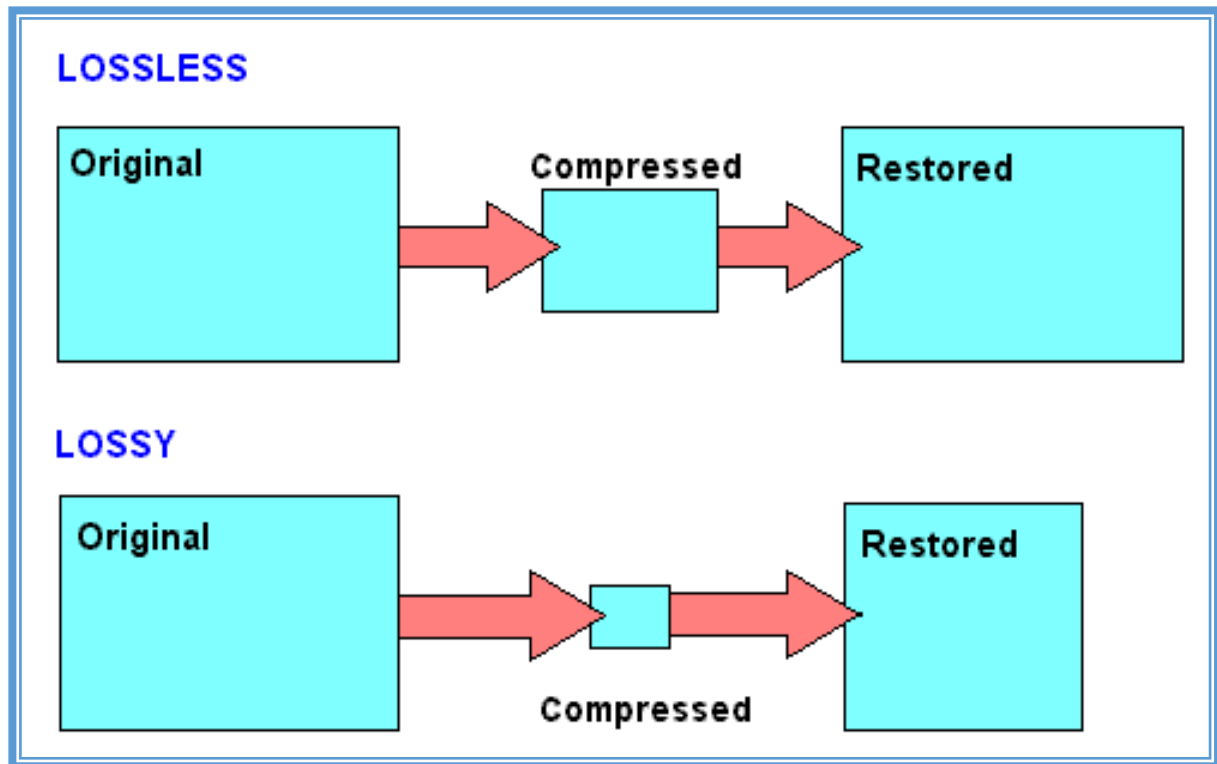


Figure (1.2) shows a comparison between lossy and lossless compression

AGENDA	
Section 2	Literature Review
Section 3	Mathematical Modelling & Methodology
Section 4	Experimental Work & Analysis
Section 5	Conclusion & Future Work

In section 2, we'll discuss previous work on image compression. Many different algorithms have been devised to tackle that problem and we'll briefly reflect on a few of them.

In section 3, we'll discuss our methodology and its mathematical model. Highlighting the computational advantages of that method will explain why we preferred to utilize it.

In the following section, we'll undergo experiments on images using our algorithm. Testing the different parameters that affect the method will allow us to find the optimal conditions for image compression after analyzing the results obtained through the experiments.

In the final section, we'll reflect on our work and results. We'll also discuss possible future work and expansions.

Literature Review

PREVIOUS WORK

The main issues that many of the previous studies in the field of lossless image compression focus on are the formation of blocking artifacts and finding the ideal balance between image quality and storage size. Our research will discuss algorithms that represent the two main classes of techniques that solve these issues, which are lossy compression algorithms and lossless compression algorithms.

Discrete Cosine Transform

We will discuss DCT as an example for lossy compression methods. This class of methods prioritizes maintaining high compression ratios and an acceptable image quality over solving the issue of blocking artifacts. Solving the latter is done by post-processing methods, but none reached mainstream appeal.

The DCT method [3], invented by Nasir Ahmed in 1974 and optimized for image compression by Chen and Pratt in 1984, represents lossy compression well. As, it is the most widely used data compression algorithm as it is the basis for the JPEG format.

The steps [4] through which an image is compressed by DCT are as follows:

First, the image is split into 8x8 blocks. Second, from left to right, top to bottom, the process is applied to each block. Third, each block goes through “quantization” which reduces storage size drastically. Finally, the image is reconstructed and decompressed by using inverse discrete cosine transforms (IDCT).

This results in an 8x8 matrix which is orthogonal. This simplifies the process greatly as the inverse of the matrix will be equal to its transpose, which is used later in finding the coefficient matrix.

The DCT method can be applied to both greyscale and RGB images. Where an RGB image, sometimes referred to as a true color image, is stored as an m-by-n-by-i data array that defines red, green, and blue color components for each individual pixel. Also, this model is additive: red, green, and blue light are added together in varying proportions to produce an extensive range of colors. While greyscale images are images in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel. Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. These 256 values range from 0 to 255 in steps of 1, where pure black is represented by 0 and pure white is represented by 255. Throughout our work, we will choose the greyscale model as an example for simplicity. The original matrix of pixel values is “leveled off” by subtracting 128 from each element as The DCT is designed to work on pixel values ranging from -128 to 127.

In the coefficients matrix of the DCT, the upper left coefficients represent the lower frequencies of the original block and as we move to the right and the bottom, the frequencies represented by the coefficients are increased. As the human eye is most sensitive to low frequencies, their coefficients will be maintained in the quantization step while the higher frequency coefficients will be “clipped” [4]. This is where the “loss” occurs in this lossy compression method.

A major advantage of the JPEG process is the flexibility provided during the Quantization [4] process, as varying levels of image compression and quality can be achieved through specific quantization matrices. This allows the user to choose a quality level ranging from 1 to 100. Where 1 gives the lowest image quality and highest compression, while 100 produces the best image quality and lowest compression. That allows for a great range of quality/compression ratios to suit many different needs in multiple fields. These quantization matrices have been calculated through subjective experiments involving the human visual system. At a quality level of 50, the result is a highly compressed image with high decompressed image quality.

Singular Value Decomposition

Singular Value Decomposition (SVD) [5] generally aims to approximate the dataset of large number of dimensions using fewer dimensions. SVD achieves this through ordering the data from most variation to the least. This helps to find the region of most variation and then later SVD can be used for reduction. SVD basically factorizes the given signal which can be represented by a matrix into multiple signals (matrices). The values that act as representatives for the original signal are called singular values [6]. These singular values are arranged in descending order. Most of the singular values will be discarded while few of them are retained. The fact that the singular values are arranged in the descending order aids in this reduction. As this arrangement causes the first value of the diagonal matrix contains most information and all the further values in the signal contain decreasing amount of information about the image. Therefore, discarding such values reduces the size of the image while avoiding noticeable distortion from the original picture. That’s why this method is considered a lossy compression technique. As, loss occurs in the original data elements. The SVD technique has garnered a lot of attention in the late 90’s and early 2000’s. The SVD is utilized in many different fields [7] such as Medical Imaging, Museums and Galleries, Web Applications, Telecommunication, Face recognition and Security Industry. One exciting use of the SVD is in the field of face image compression, representation, and recognition. As, it can be responsible for major leaps in areas of neural networks, pattern recognition, and machine learning specifically in computer-human interactions.

Several approaches to utilize SVD in face recognition have been proposed. Much of the work has focused on utilizing the presence of singular values by detecting individual features such as eyes, nose, mouth, and head outline. Ultimately, a face model can be defined by the position, size, and relationships among these features.

Discrete Wavelet Transform

We will use the discrete wavelet transform (DWT) as an example for lossless compression algorithms, this class of techniques alleviates the issue of blocking artifacts but produces lower compressions ratios compared to lossy compression methods. Noting that the DWT can also be utilized as a lossy compression algorithm. This method has been gaining a lot of popularity as late as the 2010's due to how much flexibility and room for improvement it provides.

Wavelet analysis can be seen to be far superior to the DCT, used for the JPEG format to compress images, in that it doesn't form 'blocking artifacts' [2]. Usually in other algorithms the image is split into blocks (sub images) of 8x8 or 16x16 pixels, then each block is transformed separately. However, this does not consider any correlation between blocks, and creates "Blocking artifacts", which are not good if a 'high-fidelity' image is required.

That's due to the fact that wavelets transform is applied to entire images, rather than sub images, so it produces no blocking artifacts.

The implementation of 2D – DWT [8] consists of two steps. Firstly, an input image decomposed up to a desired level by two separable 2D – DWT branches. Secondly, every two corresponding sub-bands which have the same pass – band are linearly combined by either averaging or differencing. This is a major advantage of wavelet compression over other transform compression methods. Wavelet analysis can be used to divide the information of an image into approximation and detail sub signals. The approximation sub signal shows the general trend of pixel values, and three detail sub signals show the vertical, horizontal, and diagonal details or changes in the image. Although other transforms have been used such as Fourier analysis in DCT for example, a signal is broken up into sine and cosine waves of different frequencies, and it effectively re-writes a signal in terms of different sine and cosine waves. Wavelet analysis does a similar thing, it takes a 'mother wavelet', then the signal is translated into shifted and scale versions of this mother wavelet. The best way to describe discrete wavelet transform is through a series of cascaded filters [2]. The input image X is fed into low pass filter (L) and high pass filter (H) separately. the output of the two filters is the subsampled. The resulting low pass sub band y_L and high pass sub band y_H are shown in figure (2.1). The original signal can be reconstructed by synthesis filters (L) and (H) which take the up sampled y_L and y_H as inputs.

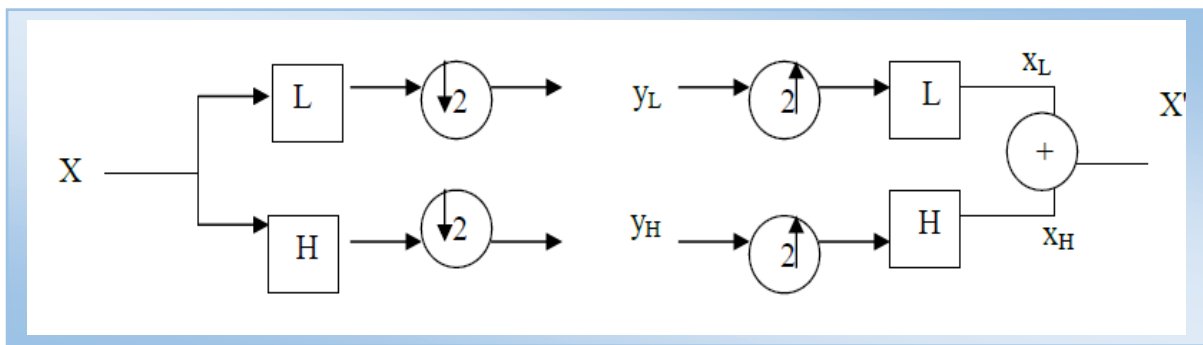


Figure (2.1) Wavelet decomposition and reconstruction process

For a two – dimensional images, the approach of the 2D implementation of the discrete wavelet transform (DWT) is to perform the one – dimensional DWT in row direction and it is followed by a one – dimensional DWT in column direction. See figure (2.2), in the figure, LL is a coarser version of the original image, and it contains the approximation information, which is low frequency, LH, HL, and HH are the high frequency sub band containing the detail information. In 2D, the images are considered to be matrices with M rows and N columns. At every level of decomposition, the horizontal data is filtered, then the approximation and details produced from this are filtered on columns. At every level, four sub-images are obtained: the approximation (LL), the vertical detail, the horizontal detail, and the diagonal detail (LH, HL, HH). As in Figure (2.3), Further computations of DWT can be performed as the level of decomposition increases, the concept [2] is illustrated in figure (2.3), the second and third level decompositions based on the principle of multiresolution analysis shows that the LL1 sub band shown in figure (2.3) is decomposed into four smaller sub band LL2, LH2, HL2, and HH2.

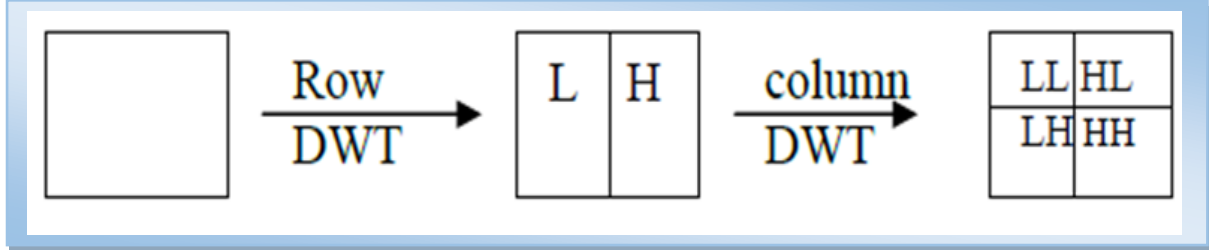


Figure (2.2): 2D row and column computation of DWT.

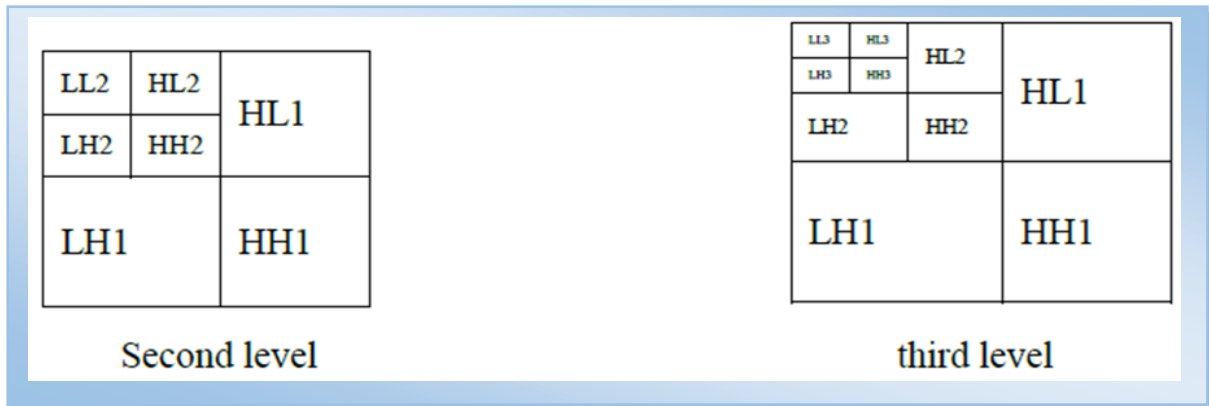


Figure (2.3): second and third level row and column decomposition.

Numerous filters can be used to implement the wavelet transform, each with its own specific coefficients, whereas the Daubechies [2] basis vectors (forward and inverse transform), for 4x4 segments, are:

$$\begin{aligned} \text{Low pass} &: \frac{1}{4\sqrt{2}} [1 + \sqrt{3}, 3 + \sqrt{3}, 3 - \sqrt{3}, 1 - \sqrt{3}] \\ \text{Low pass}_{inv} &: \frac{1}{4\sqrt{2}} [3 - \sqrt{3}, 3 + \sqrt{3}, 1 + \sqrt{3}, 1 - \sqrt{3}] \\ \text{High pass} &: \frac{1}{4\sqrt{2}} [1 - \sqrt{3}, \sqrt{3} - 3, 3 + \sqrt{3}, -1 - \sqrt{3}] \\ \text{High pass}_{inv} &: \frac{1}{4\sqrt{2}} [1 - \sqrt{3}, -1 - \sqrt{3}, 3 + \sqrt{3}, -3 + \sqrt{3}] \end{aligned}$$

We'll discuss the effect of using different wavelets on the compression process through our experimental work and analysis section.

Methodology & Mathematical Model

Reasoning Behind Choosing the Used Methodology

In order to compress any signal, especially images, many steps must be taken. The first of which is to use a “Transform”. So, why do we need a transform, or what is a transform anyway? Mathematical transformations [9] are applied to signals to obtain further information from that signal that is not readily available in the raw signal. As most of the signals in practice, are time-domain signals in their raw format. That is, whatever that signal is measuring, is a function of time. In other words, when we plot the signal one of the axes is time (independent variable), and the other (dependent variable) is usually the amplitude of that signal. When we plot time – domain signals, we obtain a time – amplitude representation of the signal. This representation is not always the best representation of the signal for most signal processing related applications. In many cases, the most distinguished information is hidden in the frequency content of the signal.

So, we translate the information of that signal into different representations. For example, the Fourier transform converts a signal between the time and frequency domains, such that the frequencies of a signal can be seen. However, the Fourier transform cannot provide information on which frequencies occur at specific times in the signal as time and frequency are viewed independently. To solve this dilemma, the Short – Term Fourier Transform (STFT) [9] introduced the idea of windows through which different parts of a signal are viewed. For a given window in time the frequencies can be viewed. However, Heisenberg’s Uncertainty Principle states that as the resolution of the signal improves in the time domain, by zooming on different sections, the frequency resolution gets worse. So, we cannot exactly know what frequency exists at what time instance, but we can only know what frequency bands exist at what time intervals. That leads to the fact that the more information we get regarding the signal in the time domain, the less information we get regarding the signal in the frequency domain.

Ideally, a method of multiresolution is needed, which allows certain parts of the signal to be resolved well in time, and other parts to be resolved well in frequency. That is why we chose the wavelet transform as the way to go forward with signal analysis. The power and magic [10] of wavelet analysis is exactly this multiresolution.

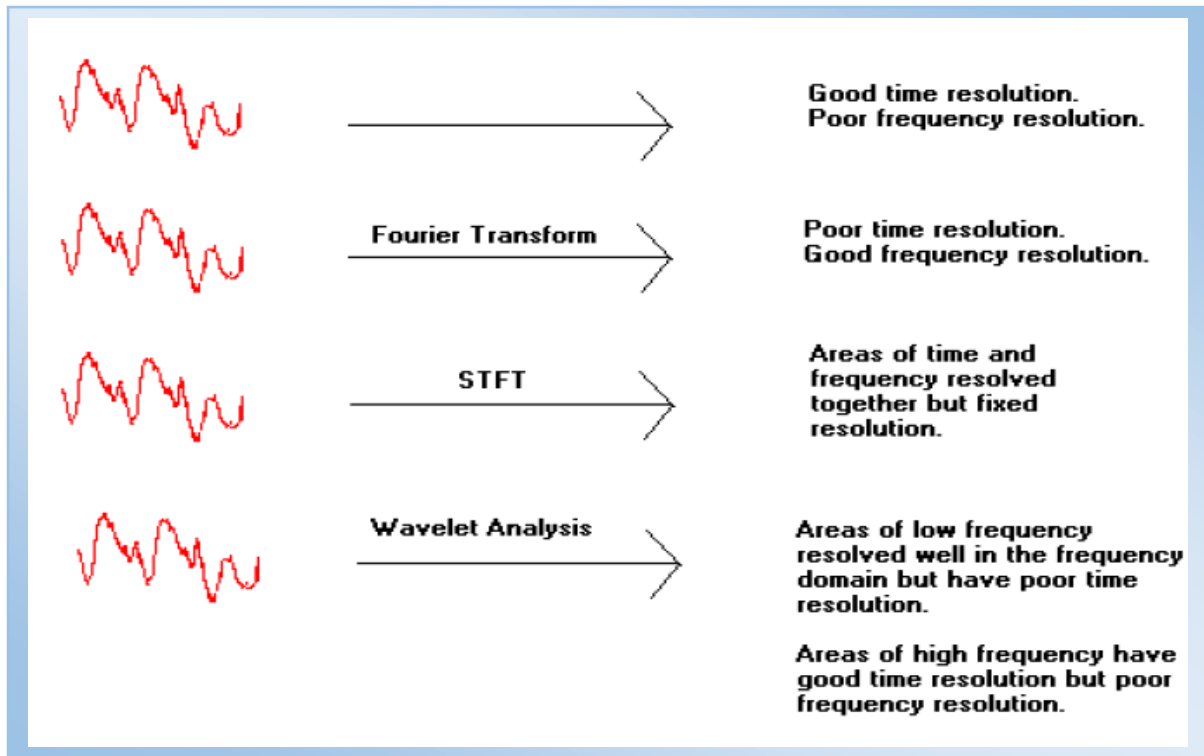
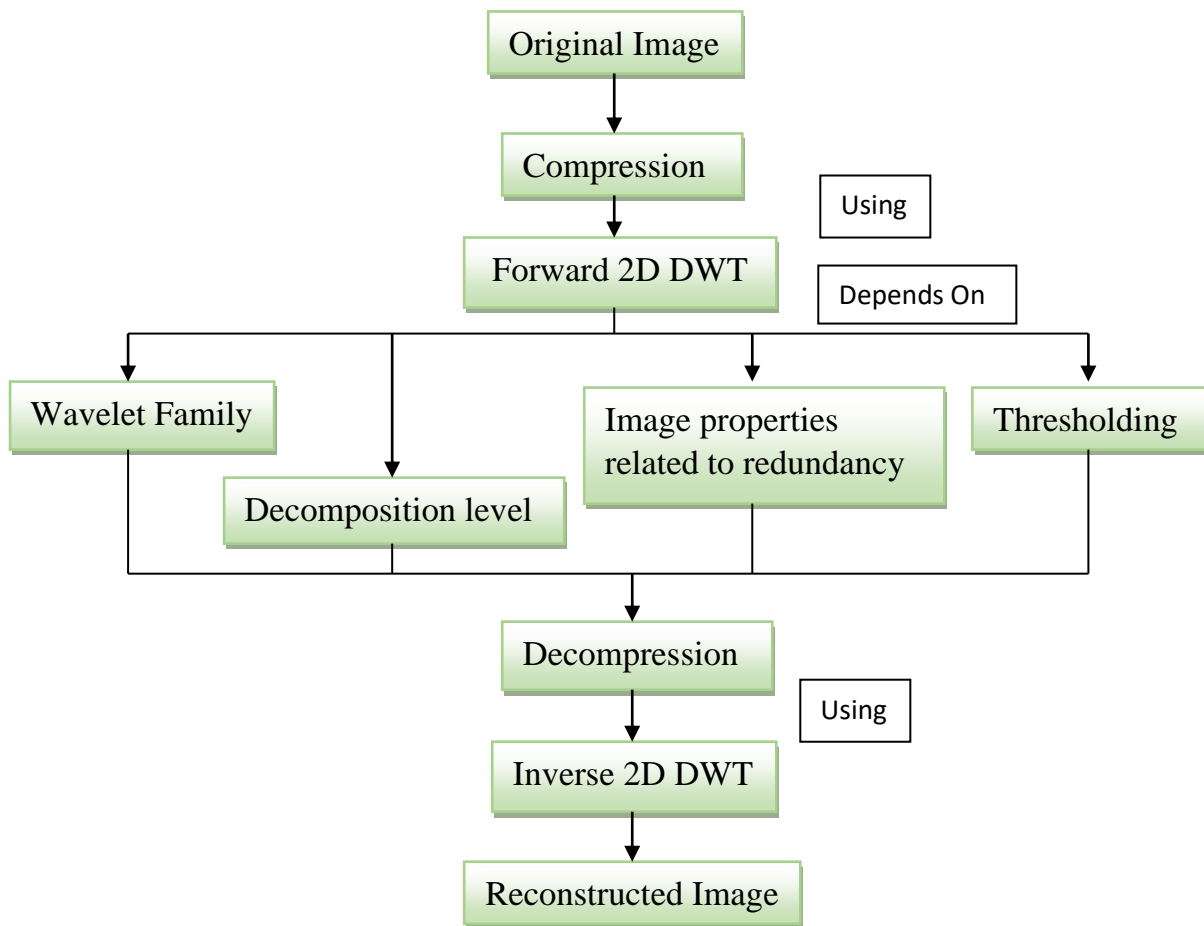


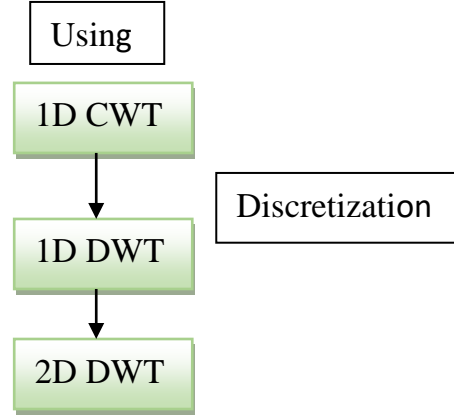
Figure (3.1) a comparison between the different transforms

Block Diagram Representing the usage of DWT In Image Compression



Mathematical Model of Wavelet Analysis

To represent wavelet analysis mathematically, we will use the following algorithm



1D Wavelet transform is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal. The frequency spectrum of a signal is basically the frequency components (spectral components) of that signal. The frequency spectrum of a signal shows what frequencies exist in the signal. The Discrete Wavelet Transform (DWT) is mathematically derived [11] from the Continuous Wavelet Transform (CWT) which is represented by the following equation:

Eq. (3.1)

$$C(s, \tau) = \int_R f(t) \psi_{s, \tau}^*(t) dt, s \in R^+ - \{0\}, \tau \in R$$

$c(s, \tau)$ represents wavelet coefficients. The $*$ denotes the complex conjugate. The subscripts denote: s – scale, τ – shift (translation). Wavelets are generated from the single mother wavelet $\Psi(t)$ by scaling s and shifting τ ; $s > 1$ dilates, $s < 1$ contracts the signal, and so the equation takes the form of (eq. (3.2)):

Eq. (3.2)

$$\psi_{s, \tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t - \tau}{s}\right)$$

The coefficient $\left(\frac{1}{\sqrt{s}}\right)$ is used because the energy must be normalized across different scales. The integral present in (eq. (3.1)) can be interpreted as the scalar (inner) product of the signal $f(t)$ and the particular wavelet (basis function) $\Psi_{s, \tau}(t)$. This scalar product tells what degree the shape of the signal is similar (correlated) to the particular wavelet. But there is a major issue regarding the use of the CWT in image compression which is the fact that the continuous change of scale s and shift τ

parameters would lead to a very redundant signal representation. It is convenient to change scale and shift parameters in discrete steps. This discretization yields discrete wavelet transformation (DWT). It is advantageous to use special values for shift τ and scale s while defining the wavelet basis, by introducing the scale step j (also called the scale parameter) and the shift step k (also the shift parameter): $s = 2^{-j}$ and $\tau = k \cdot 2^{-j}$. (where $j = 1, \dots$; $k = 1, \dots$) to (eq.1) which represents the CWT. We obtain (eq. (3.3)) which represents the new discretized wavelet basis bringing us closer to the DWT. Eq. (3.3)

$$\psi_{s,t}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t - \tau}{s}\right) = \frac{1}{\sqrt{2^{-j}}} \psi\left(\frac{t - k2^{-j}}{2^{-j}}\right) = 2^{\frac{j}{2}} \psi(2^j t - k)$$

Next, equation (3.4) represents the Forward 1D DWT.

$$c(j, k) = \sum_t f(t) \psi_{j,k}(t), \text{ where } \psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k)$$

While equation (3.5) represents the Inverse 1D DWT. Eq. (3.4)

$$f(t) = \sum_k \sum_j c(j, k) \psi_{j,k}(t)$$

Eq. (3.5)

To obtain the 2D form of the DWT model of a discrete signal $f(n)$ as a weighted summation of wavelets $\Psi(n)$ and plus a coarse approximation $\Phi(n)$, the following equation is used: Eq. (3.6)

$$f(n) = \frac{1}{\sqrt{M}} \sum_k W_\phi(j_o, k) \phi_{j_o, k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_o}^{\infty} \sum_k W_\psi(j, k) \psi_{j, k}(n)$$

Where j_o is an arbitrary starting scale and $(n = 0, 1, 2, \dots, M)$ and the coefficient $\left(\frac{1}{\sqrt{M}}\right)$ is similar to the scaling coefficient $\left(\frac{1}{\sqrt{s}}\right)$ present in the 1D model. Where Ψ represents the mother wavelet [11], which is a wavelet function that characterizes the basic wavelet shape and covers the entire domain of interest. While Φ represents the father wavelet, which is a scaling function that characterizes the basic wavelet scale and allows the expression needed details of the approximated function in the domain of interest. All other derived wavelets are called daughter wavelets. Daughter wavelets are defined in terms of parent wavelets with the help of the generating (basis) functions $\Psi_{j, k}(n)$, $\Phi_{j, k}(n)$. The inverse is found using the basis similar to the 1D DWT. (present in eq. (3.5)) Finally, the following equations represent the coefficients used to represent the coarse approximation and the detail wavelets.

“Approximation” Coefficients

Eq. (3.7)

$$W_{\phi}(j_o, k) = \frac{1}{\sqrt{M}} \sum_x f(x) \phi_{j_o, k}(x)$$

“Detail” Coefficients

Eq. (3.8)

$$W_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_x f(x) \psi_{j, k}(x)$$

To summarize the DWT mathematically, it decomposes a signal into a set of mutually orthogonal wavelet basis functions [9]. These functions differ from sinusoidal basis functions in that they are spatially localized - that is, nonzero over only part of the total signal length. Furthermore, wavelet functions are dilated, translated, and scaled versions of a common function, which is the mother wavelet. As is the case in Fourier analysis, the DWT is invertible, so that the original signal can be completely recovered from its DWT representation. Unlike the DFT, the DWT, in fact, refers not just to a single transform, but rather a set of transforms, each with a different set of wavelet basis functions. Two of the most common are the *Haar* wavelets and the Daubechies set of wavelets. Some important properties of wavelet functions:

- ① Wavelet functions are spatially localized.
- ② Wavelet functions are dilated, translated, and scaled versions of a common mother wavelet.
- ③ Each set of wavelet functions forms an orthogonal set of basis functions.

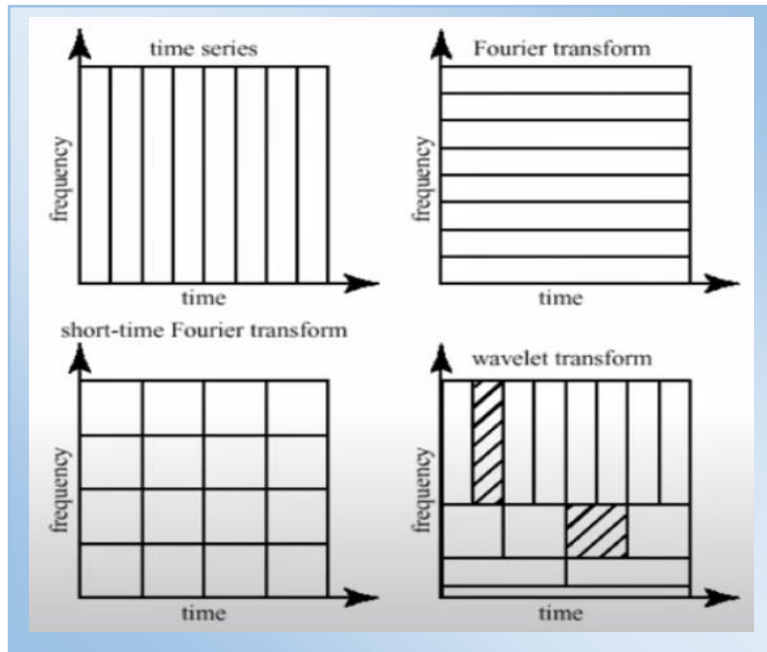


Figure (3.2) representation of information presented by multiple transforms.

There are many forms of redundancy in images and Redundancy reduction in DWT is aimed at removing duplication in the image which is apparent in three [12] different types of redundancy relevant to images:

- (I) Spatial Redundancy - correlation between neighboring pixels.
- (II) Spectral Redundancy - correlation between different color planes and spectral bands.
- (III) Psychovisual redundancies - That is due to the types of data that is ignored by the human visual system. (Such as the high frequency signals present in many images)

Where there is high correlation, there is also high redundancy, so it may not be necessary to record the data for every pixel. Images are treated as two dimensional signals, they change horizontally and vertically, thus 2D wavelet analysis must be used for images. 2D wavelet analysis uses the same 'mother wavelets' but requires an extra step at every level of decomposition. The 1D analysis filtered out the high frequency information from the low frequency information at every level of decomposition; so only two sub signals were produced at each level. In 2D, the images are considered to be matrices with M rows and N columns. At every level of decomposition, the horizontal data is filtered, then the approximation and details produced from this are filtered on columns. So, the DWT analyzes 2D signals (Images) by the following algorithm: The first decomposition step takes the input and provides two sets of coefficients at level 1: approximation coefficients cA_1 and detail coefficients cD_1 . The signal s is passed through a low-pass filter for approximation and through a high-pass filter for detail. This procedure is repeated recursively [11] to obtain approximation and detail coefficients at further levels. This yields a tree-like structure of filters called filter bank.

The filter bank structure of coefficients for level $j = 3$ appears in figure (3.3).

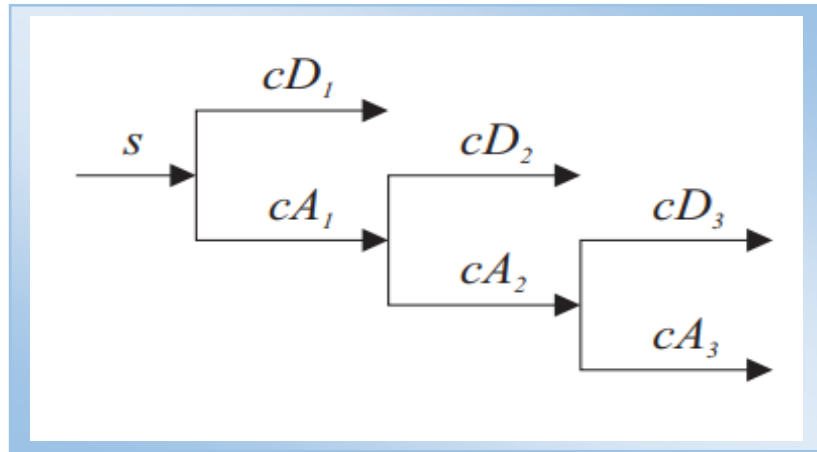


Figure (3.3) filter bank structure.

The 2D discrete wavelet transformation decomposes a single approximation coefficient at level (j) into four components at level ($j + 1$):

- ① the approximation coefficient cA_{j+1} and detail coefficients at three orientations:
- ② horizontal $cD^h_{(j+1)}$.
- ③ vertical $cD^v_{(j+1)}$.
- ④ and diagonal $cD^d_{(j+1)}$.

The symbol $(\text{col} \downarrow 2)$ represents down-sampling columns by keeping only even indexed columns. Similarly, $(\text{row} \downarrow 2)$ means down-sampling rows by keeping only evenly indexed rows.

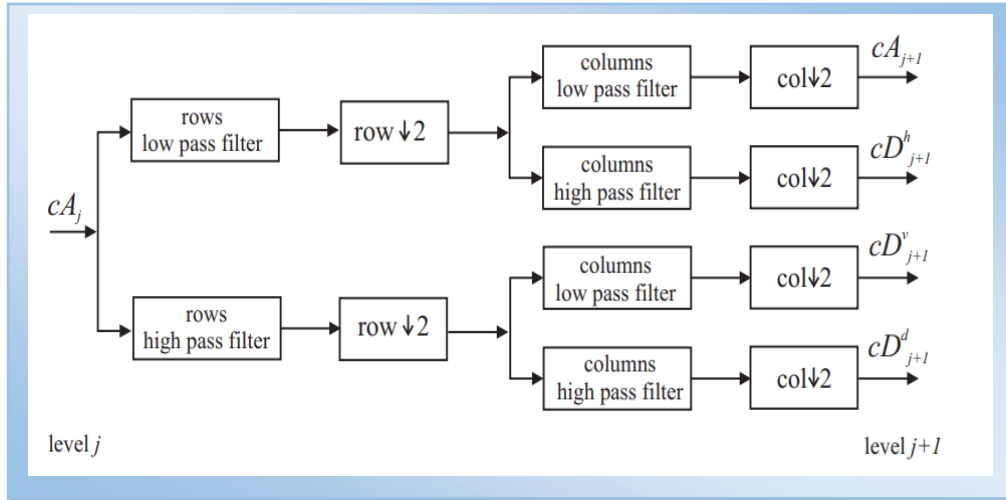


Figure (3.4) A decomposition step in 2D discrete wavelet transform

At every level of decomposition, four sub-images are obtained: the approximation, the vertical detail, the horizontal detail, and the diagonal detail. Below in figure (3.5) the original image used has been decomposed to one level. The wavelet analysis has found how the image changes vertically, horizontally, and diagonally as shown in figure (3.6).

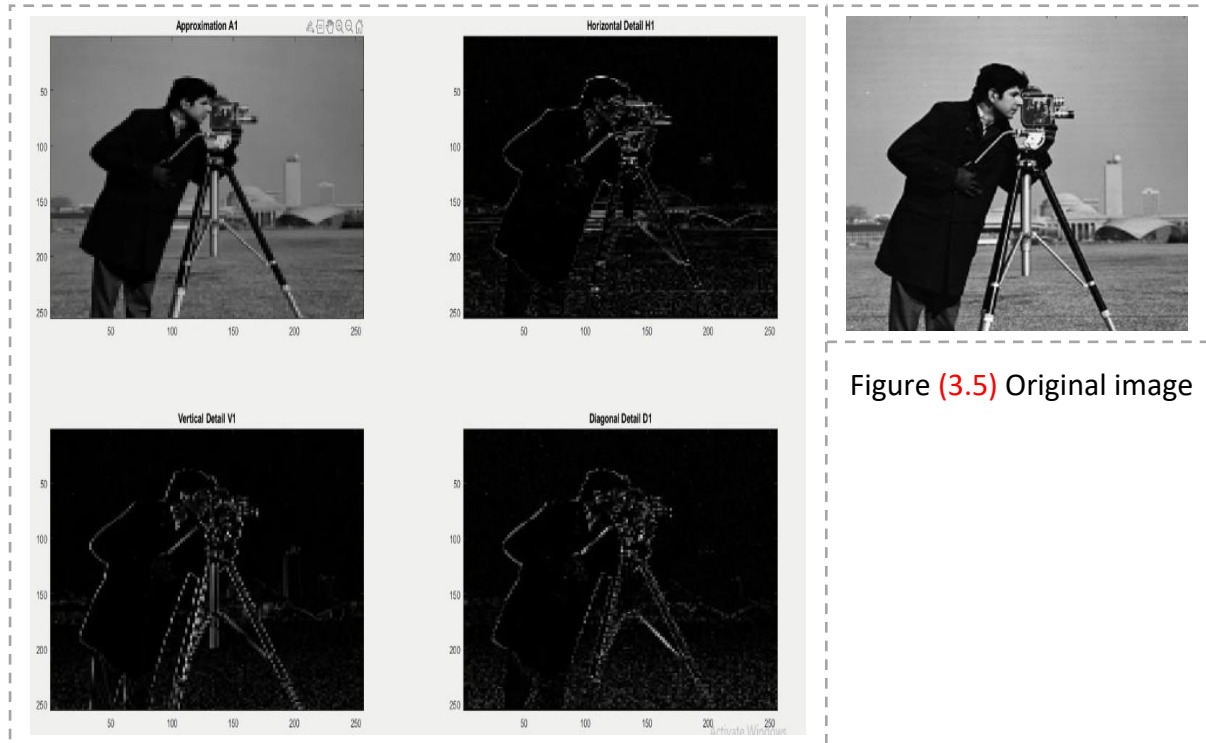


Figure (3.5) Original image

Figure (3.6) The image after one decomposition step in 2D DWT.

The Use of Thresholding In DWT

Thresholding allows us to use the DWT as both a lossy and lossless compression methodology. It also allows us to somewhat control the compression ratio and its effect on image quality. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than a fixed value called the threshold, or a white pixel if the pixel intensity is greater than that threshold. While in some cases, the threshold can or should be selected manually by the user, there are many cases where the user wants the threshold to be automatically set by an algorithm. In those cases, the threshold should be the "best" threshold in the sense that it should separate in two classes the brighter objects considered to be part of the foreground and the darker objects considered to be part of the background. Image data can be represented by coefficients of discrete image transforms. Coefficients that make only small contributions to the information contents can be omitted. If these details are very small, then they can be set to zero without significantly changing the image. The value below which details are considered small enough to be set to zero is known as the threshold. The greater the number of zeros the greater the compression [10] that can be achieved. The amount of information retained by an image after compression and decompression is known as the "energy retained" and this is proportional to the sum of the squares of the pixel values. If the energy retained is 100% then the compression is known as lossless, as the image can be reconstructed exactly. This occurs when the threshold value is set to zero, meaning that the detail has not been changed. If any values are changed then energy will be lost, and this is known as lossy compression. Ideally, during compression the number of zeros and the energy retention will be as high as possible. However, as more zeros are obtained more energy is lost, so a balance between the two needs to be found. For some signals, many of the wavelet coefficients are close to or equal to zero. Thresholding can modify the coefficients to produce more zeros. In hard thresholding any coefficient below a threshold T , is set to zero. This should then produce many consecutive zeroes which can be stored in much less space and transmitted more quickly. To compare different wavelets, the number of zeros is used. More zeros will allow a higher compression rate, if there are many consecutive zeros, this will give an excellent compression rate. The energy retained describes the amount of image detail that has been kept, it is a measure of the quality of the image after compression. The number of zeros is a measure of compression. A greater percentage of zeros implies that higher compression rates can be obtained.

There are a number of different options [10] for thresholding. These include:

- ① The approximation signal is thresholded or not thresholded.
- ② Level dependent or global threshold values.
- ③ Threshold different areas of an image with different threshold values.

In our experimental work, we'll utilize the global threshold method, due to its simplicity and time efficiency.

Finally, the inverse Wavelet Transform takes as an input the approximation coefficients cA_j and detail coefficients cD_j and inverts the decomposition step [11]. The vectors are extended (up sampled) to double length by inserting zeros at odd-indexed elements and forming the result with the reconstruction filters. Similar to down sampling, up sampling is denoted $\uparrow 2$ in the block diagrams.

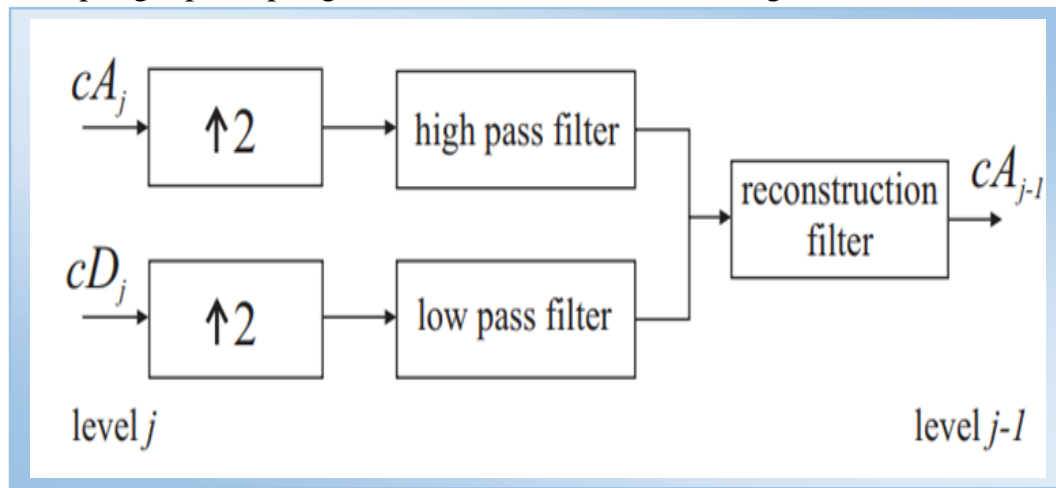


Figure (3.7) Reconstruction and Decompression through Inverse DWT.

Experimental Work

Before we start our testing process, we have to address one defining characteristic of images and that is “Entropy” [10] or randomness. The fact that images store so much information in the intensity values of the of their greyscale pixels which range from 0 (pure black) to 255 (pure white) is inherently why each image is different. The variation between these values also represents the level of detail presented in the image. The significance of the difference in entropy values is noted by observing the difference in detail between fig (4.1) and fig (4.11). That obviously affects how these images are processed generally and compressed specifically. So, in order to undergo meaningful testing of any image compression technique it is necessary to acknowledge that property. Simply, a highly correlated picture will have a low entropy. For example, a very low frequency, highly correlated image will be compressed sufficiently by many different techniques; it is more the image inherent property and not the compression algorithm that is mainly responsible for the good compression rates. Also, a fact that really affects our research is that a compression algorithm that is good for some images will not necessarily be good for all images, as each algorithm targets specific redundancies and properties of an image to compress it. So, in order to obtain the best possible compression ratio, we would have to choose that technique on a case – by – case basis based on its properties and entropy. Through our work, we attempt to find an optimized technique that allows us to process a

multitude of different images efficiently. One way of estimating entropy mathematically [10] is represented by the following equation:

$$\tilde{H}_e = - \sum_{k=0}^{G-1} \tilde{P}(k) \log_2[\tilde{P}(K)]$$

Eq. (4.1)

Eq. (4.2)

$$\tilde{P}(k) = \frac{h(k)}{MN}$$

Where G represents the number of grey – levels and $\tilde{P}(k)$ represents an estimation of the probability of grey – level k and is calculated through (Eq. (4.2))

Where h(k) stands for the frequency of grey – level k in an image, M represents the number of rows and N represents the number of columns of the image's matrix.

Using the MATLAB functions in our experimental code, we calculated the image entropy of multiple images as illustrated in Table (1).

Image	Image Entropy
Moon [<i>fig(4. 1)</i>]	5.5128
Family [<i>fig(4. 6)</i>]	5.5169
Cameraman [<i>fig(4. 11)</i>]	7.0097

Table 1

As we test the effect of Entropy on the image compression process, we'll also test and analyze the significance and effect of three main parameters which are: Threshold used (and utilizing an optimized threshold), Level of decomposition and Wavelet used.

Measures Of Effectiveness

To define the overall quality of the compression process, many measures of effectiveness were defined. We will utilize the following:

① Compression Ratio.

Eq. (4.3)

$$K = \frac{b}{\tilde{H}_e}$$

Where b is the smallest number of bits for which the image grey levels can be represented, and \tilde{H}_e represents the estimate of image entropy obtained through (Eq. (4.1)).

As this metric increases, the cost of storage and transmission decreases but the loss in terms of image quality increases simultaneously and vice versa. That's due to the fact that compression targets the similarities and redundancies in an image to decrease storage size.

② Energy retained.

It's calculated through (Eq. (4.4)) and it represents the flipside of the previous metric. As it represents how faithful the output image is to the original image. It's heavily dependent [10] on the threshold value set. So, as it increases, the loss in detail decreases. And if it's at 100%, then we're dealing with lossless compression.

$$\frac{100 * (\text{vector} - \text{norm}(\text{coeffs of the curent decomposition}, 2))^2}{(\text{vector} - \text{norm}(\text{original signal}))^2}$$

Where vector represents the used wavelet, noting that each wavelet has its own coefficients.

Eq. (4.4)

③ Mean square error

The mean squared error (MSE) is defined as the mean of the square of the difference between the original and reconstructed pixels, x and x_o . The average of the square of the difference between the desired response and the actual system output. As a loss function, As MSE increases, the image quality degrades and vice versa. It's calculated through (Eq. (4.5)).

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

Eq. (4.5)

Where mn is the image size and $I(i, j)$ is the input image and $K(i, j)$ is the retrieved image.

④ Peak Signal to Noise Ratio

The PSNR represents the quality of reconstruction of the output of the lossy compression process. The signal in this case is the original data, and the noise is the error introduced by compression. A higher PSNR would normally indicate that the reconstruction is of higher quality. It is used as an approximation to human perception of reconstruction quality. However, in some cases one reconstruction may appear to be closer to the original than another even though it has a lower PSNR as it's only an approximation. The PSNR is calculated by using (Eq. (4.6)).

Eq. (4.6)

$$PSNR = 10. \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

Here, MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. More generally, when samples are represented using B bits per sample, MAX_I is $2^B - 1$.

In our First Scenario of testing, we'll use the level of decomposition as a control variable (set as 1) and the wavelet used is (*Bior 3.7*). We'll monitor how the global threshold set affects the compression process.






Figure #	Image	Notes
(4.1)		Original Image Image Entropy = 5.5128 Its Standard deviation = 78.0031
(4.2)		$K = 0.3131$ Retained Energy = 100 $MSE = 1.0833 * 10^{-34}$ PSNR = 387.7832 Threshold set = zero
(4.3)		$K = 74.5145$ Retained Energy = 99.9689 $MSE = 2.8523 * 10^{-7}$ PSNR = 113.5789 Threshold set = 10
(4.4)		$K = 75$ Retained Energy = 99.9556 $MSE = 2.8523 * 10^{-7}$ PSNR = 113.5789 Threshold set = 78.0031
(4.5)		$K = 75$ Retained Energy = 99.9556 $MSE = 2.8523 * 10^{-7}$ PSNR = 113.5789 Threshold set = 100

Table 2

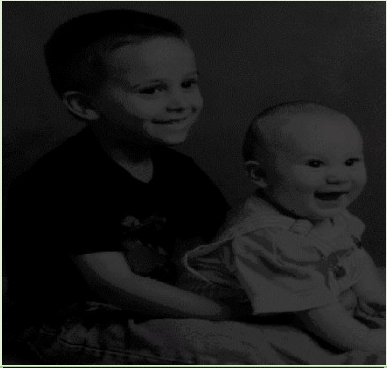
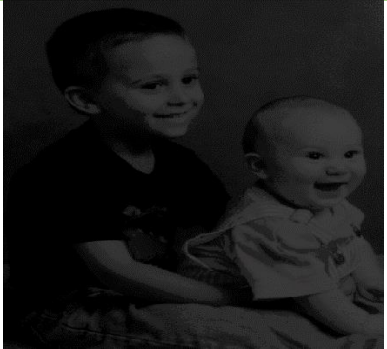
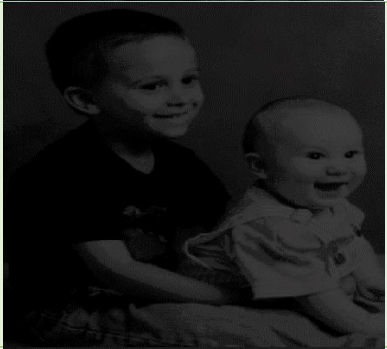
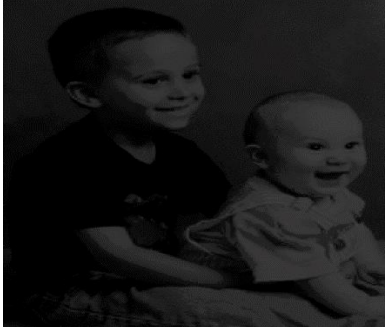
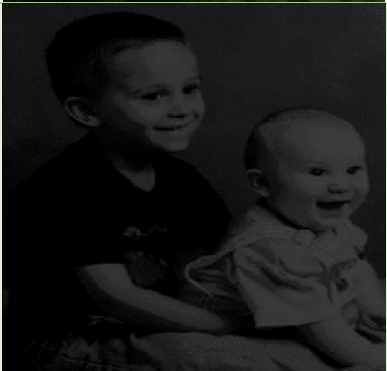
Figure #	Image	Notes
(4.6)		Original Image Image Entropy = 5.5169 Its Standard deviation = 19.2288
(4.7)		$K = 6.6170$ Retained Energy = 100 $MSE = 3.0815 * 10^{-33}$ PSNR = 373.2432 Threshold set = 0
(4.8)		$K = 74.7264$ Retained Energy = 99.5683 $MSE = 1.2986 * 10^{-5}$ PSNR = 96.9961 Threshold set = 10
(4.9)		$K = 75.9007$ Retained Energy = 99.3877 $MSE = 1.4673 * 10^{-5}$ PSNR = 96.4655 Threshold set = 19.2288
(4.10)		$K = 75.9662$ Retained Energy = 99.3569 $MSE = 1.4673 * 10^{-5}$ PSNR = 96.4655 Threshold set = 100

Table 3






Figure #	Image	Notes
(4.11)		<p>Original Image Image Entropy = 7.0097 Its Standard deviation = 62.3417</p>
(4.12)		<p>$K = 0.4115$ Retained Energy = 100 $MSE = 4.9304 * 10^{-32}$ PSNR = 361.2020 Threshold set = 0</p>
(4.13)		<p>$K = 66.0878$ Retained Energy = 99.9709 $MSE = 1.1203 * 10^{-6}$ PSNR = 107.6377 Threshold set = 10</p>
(4.14)		<p>$K = 74.6063$ Retained Energy = 99.7033 $MSE = 3.9625 * 10^{-6}$ PSNR = 102.1511 Threshold set = 62.3417</p>
(4.15)		<p>$K = 74.9259$ Retained Energy = 99.6007 $MSE = 3.9625 * 10^{-6}$ PSNR = 102.1511 Threshold set = 100</p>

Table 4

In our second Scenario of testing, we'll use the global threshold set as a control variable (set as the image's standard deviation) and the wavelet used is (Bior 3.7). We'll monitor how the level of decomposition affects the compression process.






Figure #	Image	Notes
(4.16)		Original Image
(4.17)		$K = 74.6063$ Retained Energy = 99.7033 $MSE = 3.9625 * 10^{-6}$ PSNR = 102.1511 Decomposition Level used = 1
(4.18)		$K = 90.4451$ Retained Energy = 99.4847 $MSE = 1.0668 * 10^{-4}$ PSNR = 87.8501 Decomposition Level used = 2
(4.19)		$K = 93.5468$ Retained Energy = 99.5763 $MSE = 2.7004 * 10^{-4}$ PSNR = 83.8165 Decomposition Level used = 3
(4.20)		$K = 94.2198$ Retained Energy = 99.7408 $MSE = 2.3074 * 10^{-4}$ PSNR = 84.4995 Decomposition Level used = 4

Table 5

In our final Scenario of testing, we'll use both the global threshold set and the level of decomposition as control variables

(set as the image's standard deviation and level 1 respectively).

We'll monitor how the wavelet used affects the compression process.




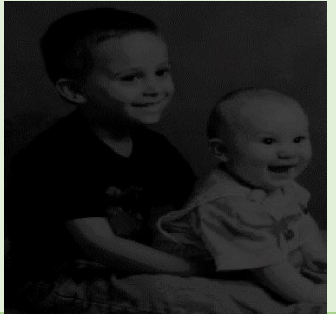

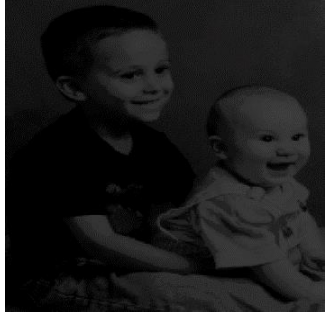
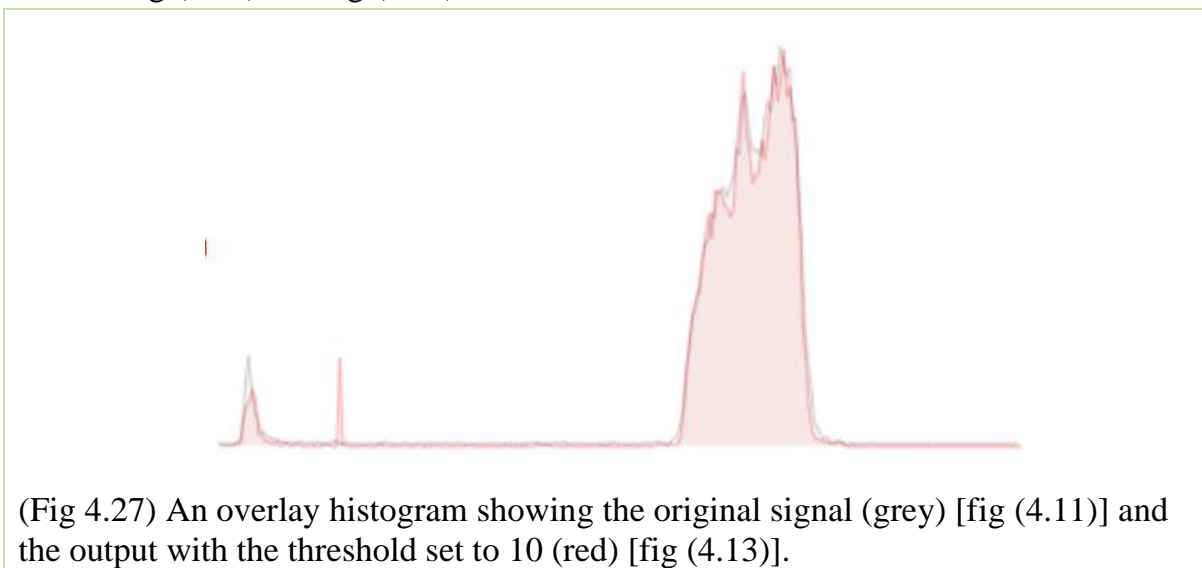
<p>$K = 74.6063$ Retained Energy = 99.7033 $MSE = 3.9625 * 10^{-6}$ PSNR = 102.1511 Wavelet used = <i>Bior 3.7</i></p> <p>Fig (4.21)</p> 	<p>$K = 75.9007$ Retained Energy = 99.3877 $MSE = 1.4673 * 10^{-5}$ PSNR = 96.4655 Wavelet used = <i>Bior 3.7</i></p> <p>Fig (4.24)</p> 
<p>$K = 73.8846$ Retained Energy = 99.5736 $MSE = 0.0011$ PSNR = 77.7072 Wavelet used = <i>Haar</i></p> <p>Fig (4.22)</p> 	<p>$K = 77.8208$ Retained Energy = 99.2257 $MSE = 9.5367 * 10^{-5}$ PSNR = 88.3368 Wavelet used = <i>Haar</i></p> <p>Fig (4.25)</p> 
<p>$K = 74.2308$ Retained Energy = 99.5860 $MSE = 4.4276 * 10^{-5}$ PSNR = 91.6692 Wavelet used = <i>db2</i></p> <p>Fig (4.23)</p> 	<p>$K = 77.2427$ Retained Energy = 99.3084 $MSE = 5.6291 * 10^{-6}$ PSNR = 100.6264 Wavelet used = <i>db2</i></p> <p>Fig (4.26)</p> 

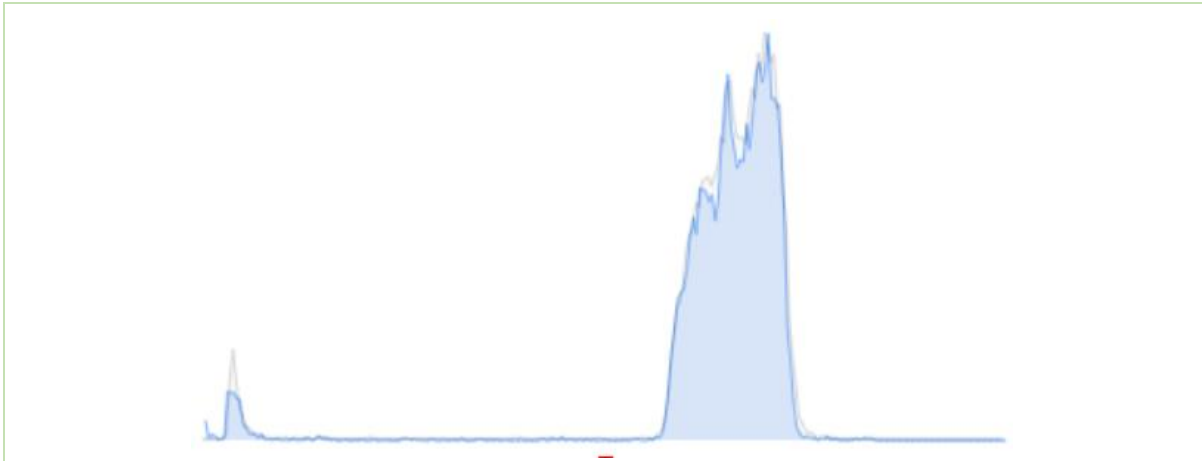
Table 6

Result Analysis

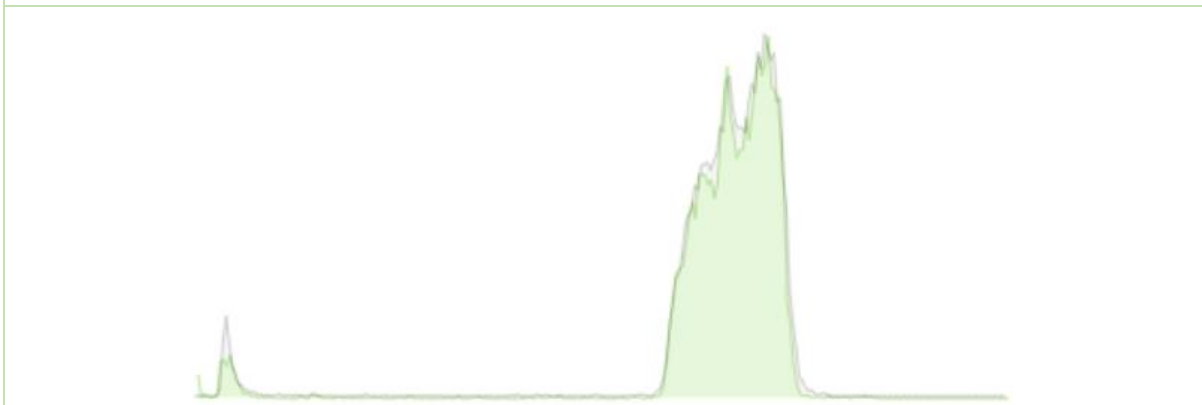
Comparing the results yielded for fig (4.3) and fig (4.8) to the values obtained for fig (4.13) shows major similarity in the value of the compression ratio between the images similar in entropy values. That proves the relation between the entropy and the compressibility of an image, as in the same conditions a 27% increase in image entropy caused a 11% decrease in compressibility.

As our work aims to find the optimum method to utilize DWT in image compression, we started by finding the optimum threshold value for compression (as previously explained, it's the value under which the approximation coefficients are set to zero causing compression and loss in detail). As illustrated in Tables (2, 3 and 4), we used a simple global threshold. Using the value 0 to represent lossless compression, which provides no loss in detail and minimal compression. We then used the value of the standard deviation of the image as a threshold. The Standard deviation is a measure of how dispersed the data is in relation to the mean. That fact allows our algorithm to adapt to the entropy of each image, which allows us to find a very simple yet very efficient way to set threshold values. As shown in tables (2,3,4), the obtained values for MSE and PSNR are capped at their optimum values for their respective level of decomposition when the value of standard deviation is reached. Using Higher threshold values also yields a highly inefficient increase in compression ratio. Comparing results for fig (4.14) and fig (4.15), for a 60% increase in Threshold value only a 0.43% increase in compressibility arises. It also causes an unnecessary loss in detail and energy retention. This redundancy is also illustrated in the image pixel values histogram illustrated in figures (4.27), (4.28) and (4.29). Where the details far from the mean were minimized going from fig (4.27) to (4.28) increasing MSE but compressing the image well, while the resultant figure remains almost the same between fig (4.28) and fig (4.29).



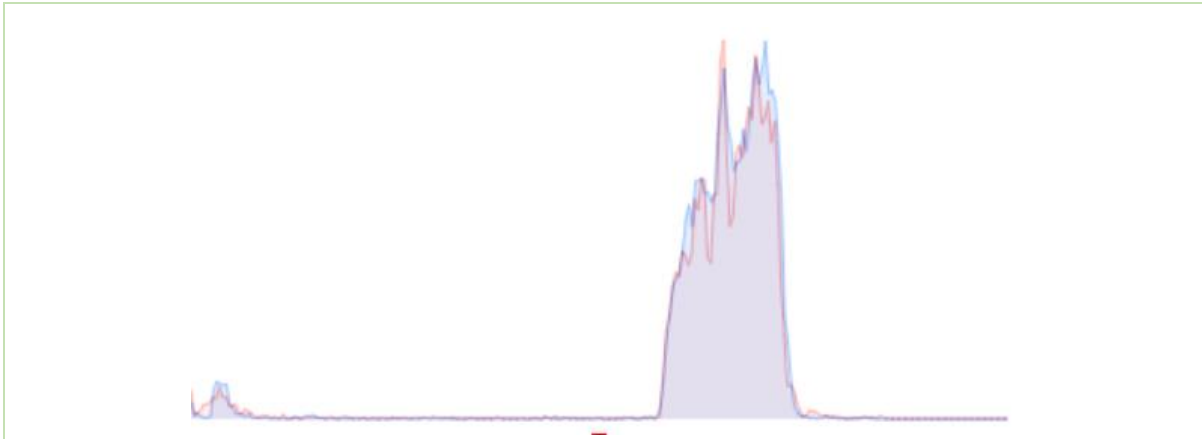


(Fig 4.28) An overlay histogram showing the original signal (grey) [fig (4.11)] and the output with the threshold set to the Standard deviation (blue) [fig (4.14)].



(Fig 4.29) An overlay histogram showing the original signal (grey) [fig (4.11)] and the output with the threshold set to 100 (green) [fig (4.15)]

Using the results present in table 5, It's easy to infer how significant the level of decomposition is to the compression. As increasing the decomposition level from one to two in fig (4.17) and fig (4.18) causes a significant increase in compression ratio (21%) and a noticeable decrease in quality both quantitatively a 14% decrease in PSNR and qualitatively as the edges get visibly distorted and background details are affected as well. However, the extra step in decomposition shown in fig (4.19) only presents a 3.43% increase in compression ratio (a seventh of the previous jump) while further decreasing PSNR by 4.6% (a third of the previous decline). From the 4th step onwards, the image is almost unrecognizable.



(Fig 4.30) An overlay histogram showing declv12 [fig (4.18)] and declv13 [fig (4.19)]

The figure (4.30) shows the serious deviation of the third decomposition level in the middle section, which represents the densest, intense, and significant values. That explains the noticeable decline in quality. Finally, no concrete relation between the wavelet used and the overall compression quality can be inferred through the results present in (table 6). Comparing fig (4.21) and fig (4.22), where the ***Haar*** wavelet is used rather than the biorthogonal wavelet family. Both the energy retention and compression ratio decrease while the MSE increases massively. While for fig (4.22) and fig (4.23), where the Daubechies wavelet family is used. The compression ratio increases while the energy retention decreases even less. That supports the fact that each compression method is unique in how it targets redundancies and minimizes them. As, Wavelets attempt to approximate how an image is changing, thus the best wavelet to use for an image would be one that approximates the image well.

Table 7

Conclusion

Reflecting on our work, choosing the wavelet transform method provided us with the flexibility needed to undergo both lossy and lossless compression. It also allowed us to solve the issue of blocking artifacts. We acknowledged the random nature of images, which is expressed through its entropy. Through that fact, we came to the conclusion that some images are more compressible compared to other images as they differ in detail density. Through it, we also concluded that no wavelet or algorithm is universally ideal to compress all images. That changes on a case-by-case basis, but through our algorithm we managed to obtain the best overall quality of compression for any certain type of wavelet. We managed to optimize our use of thresholding by tying it to the image's standard deviation value. At that value, the mean square error and peak signal to noise ratio (MSE & PSNR) are capped at their ideal values. In a case using a threshold 420% above the standard deviation only increased the compression ratio by 0.0655. That also caused an extra 0.03% loss in energy retained, all while the MSE and PSNR are constant. Tying our global threshold to a characteristic value for each specific image provided us with a great deal of adaptability, allowing the algorithm to take the image's level of detail into account. We also monitored the effect of the decomposition level on the compression process, finding that the earlier levels produce the best results. And as the level goes up, that balance between image compression and quality is lost.

Future Work

A possible avenue of optimization on the wavelet compression process is to introduce an extra step of compression, using another compression algorithm that is the discrete fractional Fourier transform (*DFrFT*) [13]. That extra step would only be done on the LL sub – band (the approximation sub-image), which could benefit from further compression. Further testing would be done to find the optimum values of the parameters that affect the *DFrFT* (called fractional orders), also monitoring various scenarios in which increasing the level of decomposition would affect this extra step of compression. Also, finding whether the type of wavelet would affect that modification.

REFERENCES

- [1] K. Sayood, *Introduction To Data Compression*, Elsevier, 2018.
- [2] N. M. Al-Shereefi, "Image Compression Using Wavelet Transform," *Journal of Babylon University*, vol. 21, p. 10, 2013.
- [3] N. Ahmed, *How I Came Up With The Discrete Cosine Transform*, 1991.
- [4] K. C. a. P. Gent, "Image Compression and the Discrete Cosine Transform," *College of the Redwoods Journal*, p. 11.
- [5] H. R. Swathi, "Image compression using singular value," 2017.
- [6] L. Cao, "Singular Value Decomposition Applied to DIP," *Arizona State University Journal*, p. 15.
- [7] R. Jameela, "JPEG Image Compression using Singular Value Decomposition," 2011.
- [8] N. S. Afifi, "Image compression using Wavelets," 2011.
- [9] D. S. Roy, "Wavelet based Digital Image Compression," 2020.
- [10] K. Lees, "Image Compression Using Wavelets," 2002.
- [11] V. Hlaváč, "Wavelets transformation," Prague.
- [12] B. Reddaiah, "A Study on Image Compression and its Applications," *International Journal of Computer Applications*, vol. 177 , no. 38, p. 4, 2020.
- [13] R. J. d. B. & B. J. Naveen Kumar, "A lossless image compression algorithm using wavelets and fractional Fourier transform," *SN Applied Sciences Journal*, vol. 1, no. 266, 2019.