Republic of Iraq
Ministry of Higher Education
and Scientific Research
Al-Nahrain University
College of Science
Department of Mathematics and
Computer Applications

Some Results of Mathematical-Based Transformation Methods and their Application to Image Compression

A Thesis

Submitted to the College of Science, Al-Nahrain University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Mathematics

by

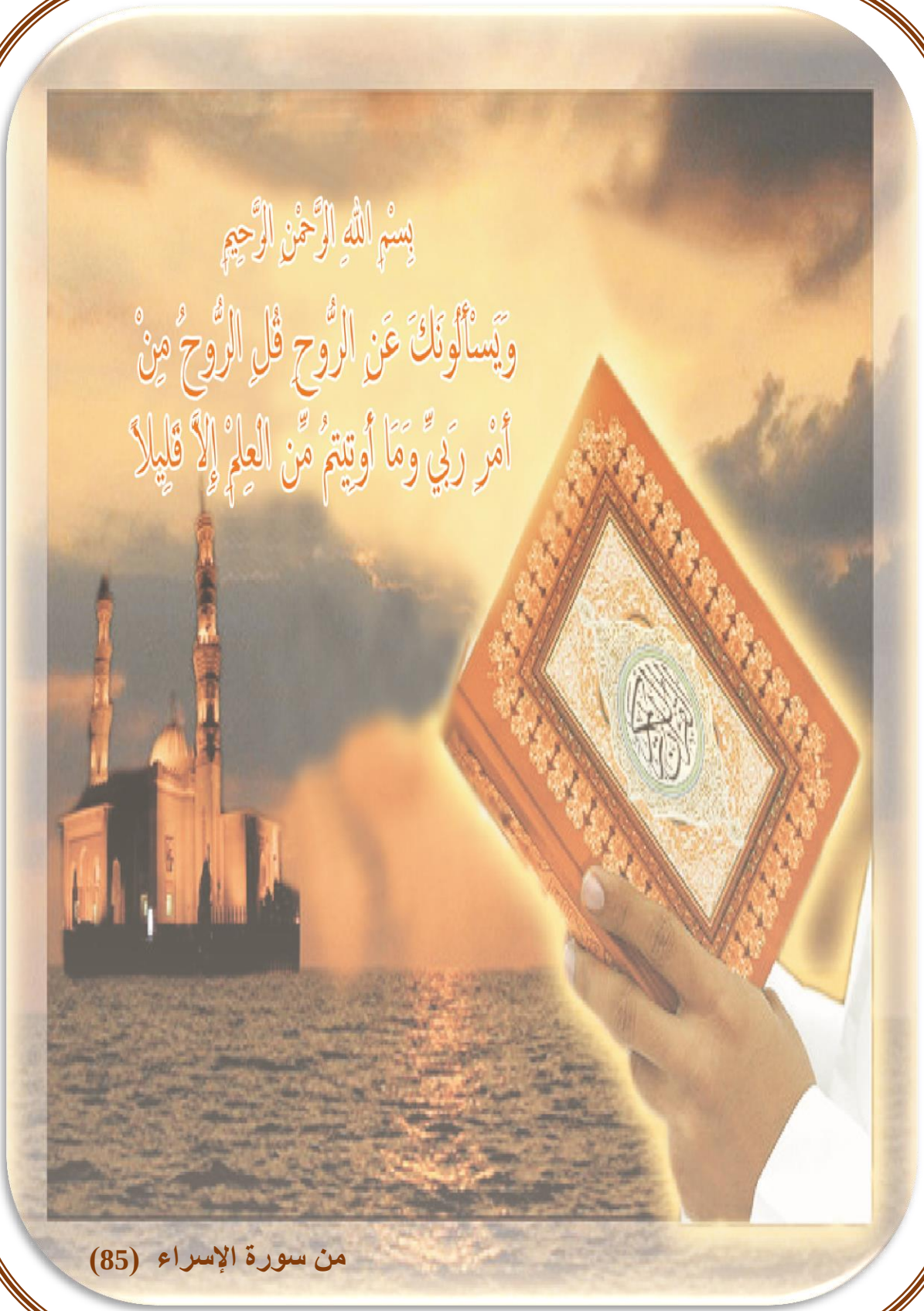
Hadeel Majid Rasheed Al-Ani

(B.Sc. mathematics / College Science / University of Technology, 1999)

Supervisor by

Asst. Prof. Dr. Ali Hasan Al-Fayadh

1438 *2017*


بِسْمِ اللهِ الرَّحْمنِ الرَّحِيمِ
وَيَسْأَلُونَكَ عَنِ الرُّوحِ قُلِ الرُّوحُ مِنْ
أَمْرِ رَبِّي وَمَا أُوتِيتُم مِّنَ الْعِلْمِ إِلاَّ قَلِيلاً
من سورة الإسراء (85)

Supervisor Certification

I certify that this thesis entitled,

" Some Results of Mathematical-Based Transformation Methods and their Application to Image Compression ",

was prepared by " ***Hadeel Majid Rasheed Al-Ani*** " , under my supervision at the Department of Mathematics and Computer Applications, College of Science, Al-Nahrain University, as a partial fulfillment of the requirements for the degree of Master of Science in Mathematics.

Signature

Name: ***Dr. Ali Hassan Nasser Al-Fayadh***

Scientific Degree: **Assist. Professor**

Address : College of Science / Al-Nahrain University

Date: / /2016

(Supervisor)

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature

Name: ***Dr. Osama H. Mohammed***

Scientific Degree: **Assist. Professor**

Address: Head of Mathematics and Computer Applications Department

Date: / /2016

Committee Certification

We, the examining committee, certify that we have read this thesis entitled **“ *Some Results of Mathematical-Based Transformation Methods and their Application to Image Compression* ”,** and examined the student **"*Hadeel Majid Rasheed Al-Ani* ",** in its contents and that in our opinion, it is accepted as a thesis for the degree of **Master of Science in Mathematics**.

Signature
Name: ***Dr. Luma Naji Mohammed Tawfiq***
Scientific Degree: **Professor**
Address : College of Education for Pure Science
Ibn-Al-Haitham/ Baghdad University
Date: / / 2017
(Chairman)

Signature
Name: ***Dr. Fadhel Subhi Fadhel***
Scientific Degree: **Assist. Professor**
Address : College of Science /
Al-Nahrain University
Date: / / 2017
(Member)

Signature
Name: ***Dr. Mushtaq Shakir A. Hussain***
Scientific Degree: **Assist. Professor**
Address : College of Science /
AL-Mustansiriyah University
Date: / / 2017
(Member)

Signature
Name**:** ***Dr. Ali Hassan Nasser Al-Fayadh***
Scientific Degree: **Assist. Professor**
Address: College of Science / Al-Nahrain University
Date: / /2017
(Supervisor)

I , hereby certify upon the decision of the examining committee .

Signature
Name: ***Dr. Hadi M. A. Abood***
Scientific Degree: **Professor**
Address: Dean of the College of Science
Al-Nahrain University
Date: / /2017

Acknowledgments

In the name of **ALLAH**, the Most Compassionate, the Most Merciful, praise be to **ALLAH** and pray and peace be on his prophet **Mohammed (Allah's blessing)** and peace be on his relatives.

Praise be to Allah, Who created us and gave us the ability and mentality to think and work for the benefit of human beings.

I would like to express my deep gratitude to my supervisor ***Asst. Prof. Dr. Ali Hassan Nasser Al-Fayadh*** for his supervision during the period of the research, encouragement, comments and help. He opened my mind to the world of a new vision and has given me the necessary information to build my own identity as a researcher, I wish the good health and success for him.

My deep thankfulness goes to the head of Mathematics and computer applications department ***Asst. Prof. Dr. Osama H. Mohammed*** and all the member of the staff in the Department for their support and continuous encouragement given during the course of my study.

I also greatly thank to the Administrative ***Asst. Prof Dr. Fadhel Subhi Fadhel*** for his assistance and bounty and his encouragement during the period of study; I wish the best hopes and success for him.

Finally, I would like to thank all those who stand beside me in my study and give me the encouragement to do this work, sincere thanks are extended to all those who gave a helpful hand to complete this study.

Hadeel

Dedication

Every challenging work needs self efforts as well as guidance of older especially who were very close to our heart.

My humble effort I dedicate to ; my sweet and loving

Family

Whose affection, love, encouragement and prays of day and night make me able to get such success and honor.

All that I am, or hope to be, owe to them.

My lovely best friends

Side by side or miles a part
They will always be connected with me by heart
And they were beside me all days of my life

Any one

Stand with me, love me, and devotion to me
And always there for me
When my spirits need a little lift

Hadeel

ABSTRACT

The main purpose of this thesis is to study and investigate the most important properties of integral transforms (the discrete fourier transform, the discrete sine transform, and the discrete wavelet transform) and their mathematical aspects both from the theoretical point of view and for the application to image compression.

As well as, we study the singular value decomposition method and its application to image representation.

Two mathematical models are developed. The first model consists of a new multi-transform method that takes advantage of each of the discrete wavelet transform and the singular value decomposition method while the second model takes advantage of the discrete sine transform and the singular value decomposition method. These models are applied to compress images. Results show that this new approach yields better function representation and reconstruction in image compression application than is possible with the use of a single fixed transform. The proposed models improve the efficiency of the compressing process in the discrete wavelet transform and the discrete sine transform domains.

Content

Chapter Two: Wavelet Transform and Multiresolution Analysis

Chapter Three: Concepts of Image Compression

Chapter Four Proposed Compressed Models

List of Figures

List of Tables

List of Abbreviations

FT	Fourier Transform
CFT	Continuous Fourier Transform
$\mathcal{F}(f)$	Fourier Transform of the function f in one dimension
$\mathcal{F}(u,v)$	Fourier Transform of the function f in two dimension
DFT	Discrete Fourier Transform
Eq.	Equation
FFT	Fast Fourier Transform
STFT	Short Time Fourier Transform
DST	Discrete Sine Transform
DCT	Discrete Cosine Transform
WT	Wavelet Transform
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
IDWT	Inverse Discrete Wavelet Transform
MRA	Multiresolution Analysis
SVD	Singular Value Decomposition
MSE	Mean Squared Error
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity
MSSIM	Mean Structural Similarity
CR	compression ratio
VQ	Vector Quantization
JPEG2000	Joint Photographic Expert Group 2000
CVQ	Classified Vector Quantization

List of Symbols

L^p-space	set of all real (or complex)-valued functions f
$L^2(I)$	set of all square integrable functions f on I
$\emptyset_k(x)$	The basis functions of the wavelet in one dimension spaces.
$\emptyset_{s,\tau}(x)$	The scaling functions in one dimension spaces.
$h_\emptyset(n)$	The scaling function coefficients or (lowpass filter coefficients)
$\psi(x)$	Mother wavelet in one dimension spaces.
$\psi_{s,\tau}(x)$	Wavelet Function in one dimension spaces.
$g_\psi(n)$	Wavelet function coefficients or (highpass filter coefficients)
$W(s,\tau)$	wavelet coefficient of the function f(x) with scale parameter a and position parameter τ.
$\psi^*(x)$	Complex conjugate of the analyzing wavelet $\psi_{s,\tau}(x)$
$\Psi(\omega)$	Fourier transform of $\psi(t)$
$\phi(x,y)$	Scaling function in two dimension spaces.
$\psi^H(x,y),$ $\psi^V(x,y),$ $\psi^D(x,y)$	Wavelet functions in two dimension spaces in horizontal, vertical and diagonals dimensional
$W_\emptyset(s_0,m,n)$	coefficients define an approximation of $f(x,y)$ at scale s_0
$W_\psi^i(s,m,n)$	Coefficients add horizontal, vertical, and diagonal details for scales $s > s_0$ and $i = \{H,V,D\}$
s_0	The arbitrary starting scale and $s > s_0$
$c_{s_0}(\tau)$	The approximation or scaling coefficients
$d_s(\tau)$	The detail or wavelet coefficients
V_s	The vector spaces corresponding to span $\{\phi_{s,\tau}(x)\}$ over a for any τ.
W_s	The vector spaces corresponding to span $\{\psi_{s,\tau}(x)\}$ over a for any τ.

Introduction

The concept of a transformation is familiar to mathematicians. It is a standard and a powerful mathematical tool that is employed to solve problems in many areas. The idea is to change a mathematical quantity (a number, a vector, a function, or anything else) to another form.

There are many transformation methods that transform a set of quantities from their normal spatial representation to a frequency domain. Some of these include Discrete Fourier Transform, Discrete Cosine Transform, Discrete Sine Transform, and Discrete Wavelet Transform.

Most real world applications can be reduced to the problem of function representation and reconstruction. These two problems which are closely related to synthesis and analysis of functions. The Fourier transform is the classical tool used to solve them. More recently, wavelets have entered the arena providing more robust and flexible solution to discretize and reconstruct functions.

Images are very important representative objects. Among these abovementioned transformations, discrete wavelet transform is widely used for image compression because of its decorrelation and energy compaction properties. The study of wavelet transform had been motivated by the overcome some weak points in representing functions and signals by the classical Fourier transform. Fourier transform analyzes the components of a stationary signal, but it fails to analyze the non-stationary signal where wavelet transform allows the components of non-stationary signal to be analyzed.

Fourier transform is one of the most important transforms in applied mathematics and other scientific fields such as engineering and computer sciences. The aim of this process is to transform the mathematical functions that dependent on time variable to functions dependent on frequency variable. There are several different transforms. The varieties of transformations depend on a function application, such as Fourier transform, Sine transform, Cosine transform, and Wavelet transform.

Fourier analysis is a family of mathematical techniques, all based on decomposing functions (signals) into sinusoids. This surprising fact is now known as Fourier expansion and it has many applications in engineering as well as in computer sciences, namely, in image processing. The limitation of Fourier expansion is that it does not tell us when (at which point or points in time) each frequency is active in a given signal. We therefore say that Fourier expansion offers frequency resolution but no time resolution.

Wavelets provide an alternative to classical Fourier methods for both one and two dimensional data analysis and synthesis. The wavelet transform is very powerful in localizing the global spatial (time) and frequency correlation. Applications of wavelets are quite diverse and include big parts of signal and image processing, data compression, and many other fields of science and engineering.

One of the most important applications for the wavelet transform is image compression. Images are very important representative objects. They can represent transmitted television or satellite pictures, medical or computer storage pictures and many more. When a two-dimensional light intensity signal is sampled and quantized to create a digital image, a huge amount of data is

produced. The size of the digitized picture could be so great that results in impractical storage or transmission requirements. Image compression deals with this problem such that the information required to represent the image is reduced while maintaining an acceptable image quality thus making the transmission or storage requirements of images more practical.

Wavelet Image Processing enables computers to store an image in many scales of resolutions. Wavelets at different scales (resolutions) produce different results [1]. Wavelets are of real use in approximating data with sharp discontinuities such as images with lots of edges. An image can be decomposed into approximate, horizontal, vertical and diagonal details.

Discrete Fourier transform (DFT) and its invariants transforms such as discrete cosine transform (DCT), discrete sine transform (DST), and discrete wavelet transform (DWT) are widely used in digital image representation.

On the other hand, Singular Value Decomposition (SVD) is a well-known method in linear algebra. It plays an interesting fundamental role in many different applications such as dimensionality reduction and image compression [2]. The use of SVD in image compression is motivated by its excellent energy compaction property in the least square sense [3]. As a result, the use of SVD technique in image compression has been widely studied [4-5].

Another image compression technique is based on vector quantization (VQ). It is a well-known and very efficient approach to low bit-rate image compression [6]. A serious problem in ordinary VQ is edge degradation caused by employing the distortion measure, such as the mean square error (MSE), in searching for the closest codeword in the codebook, as mean square error does not accurately preserve the edge information. To tackle this problem, classified

vector quantization (CVQ) based on a composite source model, has been introduced by Ramamurthi and Gersho [7]. In CVQ model, the image is represented by shade blocks and edge blocks. In CVQ, the classifier separates these two classes and then the blocks belong to a class are coded only with the codeword belong to the same class. An obvious problem with pixel-based classification is high noise-sensitivity because of the high spatial correlation among neighboring pixels [8]. Furthermore, several researches have proposed the use of CVQ in the transform domain because it has an excellent compaction-energy [9].

This thesis proposes two mathematical models that combine different transforms for compressing still images. The first model makes use the DWT and SVD based CVQ, whereas the second model makes use the DST and SVD based CVQ. This approach combines the strengths of the DWT, DST, and the SVD based CVQ, while avoiding some of their limitations. The proposed models are lossy compression schemes that exploit the high correlation between the pixels inside the image blocks in the spatial domain as well as the energy compaction property of each of the DWT, DST and the SVD techniques, to obtain high quality reconstructed images at low bit-rates. Classification using transform domain, in which a few energy-compacted transform coefficients are used to distinguish edge direction and location, and hence it is computationally simpler than classification in the spatial domain.

This thesis consists of five chapters. Chapter one begins with an introduction to Fourier analysis; the basics of Fourier series and Fourier transform together with its invariant discrete sine transform.

Chapter two is devoted to wavelets, wavelet transforms, and multi resolution analysis (MRA) as well as the singular value decomposition method and their application to image compression. A brief description of some concepts of image processing such as sampling and quantization, an overview of the principal approaches used in digital image coding, and a general structure of image compression algorithms are presented in chapter three.

The proposed models, simulation results, are presented in chapter four.

Finally, conclusions and further work are presented in chapter five.

Literature Review

Many researches worked with Fourier transform, but that the first work is discovered by Joseph Fourier, and therefore the main branch of mathematics leading to wavelets began with Joseph Fourier (1807) with his theories of frequency analysis, now often referred to as Fourier synthesis. He asserted that any 2π −periodic function *f(x)* can be expressed as an infinite series of sine and cosine functions. Fourier's assertion played an essential role in the evolution of the mathematicians ideas about the functions. He opened up the door to a new functional universe [10].

After 1807, by exploring sense of functions, convergence of Fourier series, and orthogonal systems, mathematicians progressively were led from their prior concept of frequency analysis to the concept of scale analysis. That is, analyzing the function *f*(*x*) by creating mathematical structures that differ in scale. Take this basic structure, shift it, and scale it. Then, apply it to the same signal to get a new approximation [10].

The first mention of wavelets appeared in an appendix to the Ph.D. thesis of Alfred Haar (1909). One Haar wavelet's property is that it has compact support, that means it vanishes outside of a finite interval. In fact, Haar wavelets are not continuously differentiable that somewhat limits their applications [10]. It constitutes an orthonormal basis, but it lacks smoothness; and although the Haar basis is compact in physical space, it decays slowly in Fourier space [11].

After that, in the 1930, several groups working independently explored the representation of functions using scale-varying basis functions. Understanding the concepts of basis functions and scale-varying basis functions is the key to understanding wavelets [10].

In 1980, Morlet and Grossman, defined wavelets in the context of quantum physics. These two researchers provided a manner of thinking for wavelets based on physical intuition [10].

In 1985, Stephane Mallat gave wavelets an additional jump-start through his work in digital signal processing [10]. A couple of years later, Ingrid Daubechies utilized Mallat's work to construct wavelet orthonormal basis functions that are perhaps the most elegant, and have be the cornerstone of wavelet applications today [12].

In 1991, Myung-Sin Song focused on wavelet application in the field of image compression so as to observe how wavelet is implemented to be applied in the process of compression of an image, as well as, how wavelet's mathematical aspects affect the compression process and the results of it [13].

The concept of combining multiple transforms is getting popular in recent years as researchers have put efforts in developing schemes with various different discrete transforms. For example, the author in [14] presented a scalable algorithm for video coding where the DWT is performed on the DCT coefficients.

In 2006 Jain Y.K. and Jain S. [15], discussed important features of wavelet transform in compression of images, as well as, evaluate and compare seven different wavelet families on variety of test images set.

In order to benefit from the respective strengths of individual popular coding schemes, a new scheme, known as hybrid algorithm, has been developed where two transforms techniques are implemented together. There have been few efforts devoted to such hybrid implementation. Usama [16], presented a scalable hybrid scheme for image coding that combines both the wavelet and the Fourier transforms. Moreover, the Fourier-Wavelet Transform can be used to improve the denoising performance for images [17]

In [18], the authors have presented a hybrid transformation scheme for video coding, which minimizes prediction error. The DWT was used for intra-coding and the DCT for interceding. An extended version of the object-based coding algorithm is presented in [19].

In 2012 Harjeetpal S. and Sakhi S. [20], studied image compression techniques using DCT and DWT and concluded that DCT is used for transformation in JPEG standard and performs efficiently at medium bit rates, while DWT is used as basis for transformation in JPEG 2000 standard and provides high quality compression at low bit rates. DWT performs better than DCT in the context that it avoids blocking artifacts which degrade reconstructed images and DWT provides lower quality than JPEG at low compression rates as well as DWT requires longer compression time.

In 2015 Banti D. and Vieck A. [21], introduced and studying image compression using DCT and DWT, and explained that the image compression is of prime importance in real time applications like video conferencing, where data are transmitted through a channel. They concluded that DWT is more general and efficient than DCT due to no need to divide the input coding into non-overlapping 2-D blocks, and it has higher compression ratios while

avoiding blocking artifacts, allows good localization both in time and frequency domain.

Our work deal with combines the strengths of the DWT, DST, and the SVD based CVQ, while avoiding some of their limitations. They are lossy compression schemes that exploit the high correlation between the pixels inside the image blocks in the spatial domain as well as the energy compaction property of each of the DWT, DST and the SVD techniques, to obtain high quality reconstructed images at low bit-rates.

CHAPTER ONE

IMAGE TRANSFORMS; CONCEPTS AND MATHEMATICAL EXPRESSIONS

1.1 Introduction

In this chapter, we will recall some basic definitions and theorems from the linear algebra that will help to understand the concepts of Fourier transform and its invariants. Mathematics of some linear transforms which are used as transformations for image is reviewed. These transforms are Fourier Transform (FT), Discrete Fourier Transform (DFT), and Discrete Sine Transform (DST).

Image transforms are designed to have the following advantages:

1. They can be utilized to an image to convert it from one domain to another. Image viewing in domain such as frequency space enables the identification of features that may be not easily discovered in the time domain.
2. They are used to decorrelate of image data and transform the image data from dependent data to independent data.
3. They used to characterize the less important parts of the image by isolating the diverse frequencies of the image where the first transform coefficient should match to zero pixel frequency, and the remaining coefficients should match to higher and higher frequencies.
4. They also implement the inverse transforms.

1.2 Basic Definitions

Definition 1.2.1 [22]: Let *u* and *v* be two vectors in $\mathbb{R}^n$, such that

$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$; then

1. the ***dot product*** (also called the Euclidean ***inner product***) of *u* and *v* is denoted by $u.v$ and is defined by :

$$u.v = u_1v_1 + u_2v_2 + \cdots + u_nv_n = \sum_{i=1}^{n} u_iv_i \qquad (1.2.1)$$

2. the ***norm*** of u (also called the length of *u* or the magnitude of *u*) is denoted by $\|u\|$ and is defined by:

$$\|u\| = \sqrt{u_1^2 + u_2^2 + \cdots + u_n^2} = \sqrt{u.u} \qquad (1.2.2)$$

3. the ***distance*** between *u* and *v* is denoted by $d(u,v)$ and it is defined by

$$d(u,v) = \|u - v\| = \sqrt{(u-v)\,.\,(u-v)} \qquad (1.2.3)$$

The analog in the complex case is the following:

Definition 1.2.2 [22]: Suppose *z* and *w* are two vectors in C^n , such that

$z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}, w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$; then the (complex) ***dot product*** of *z* and *w* is

$$z.w = \sum_{j=1}^{n} z_j\,\overline{w}_j$$

where $\overline{w}_j$ is the complex conjugate of w_j. This complex dot product is a special case of a (complex) inner product.

Theorem 1.2.1 [22]: Cauchy—Schwarz Inequality

Let *u* and *v* are two vectors in $\mathbb{R}^n$, such that

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \text{ ; then}$$

$$|u.v| \leq \|u\|\|v\| \tag{1.2.4}$$

or in terms of components

$$|u_1v_1 + u_2v_2 + \cdots + u_nv_n| \leq (u_1^2 + u_2^2 + \cdots + u_n^2)^{\frac{1}{2}}.(v_1^2 + v_2^2 + \cdots + v_n^2)^{\frac{1}{2}}$$

Definition 1.2.3 [22]: Let *u* and *v* are two nonzero vectors in $\mathbb{R}^n$, and θ is the angle between *u* and *v*, then

$$cos\theta = \frac{|u.v|}{\|u\|\|v\|} \quad , \qquad 0 \leq \theta \leq \pi \tag{1.2.5}$$

Remark 1.2.1: From Eq. (1.2.5), the angle θ between two nonzero vectors *u* and *v* in $\mathbb{R}^n$ can be defined as

$$\theta = cos^{-1}\left(\frac{u.v}{\|u\|\|v\|}\right) \tag{1.2.6}$$

where the inverse cosine in Eq. (1.2.6) is not defined unless its argument satisfies the inequalities $-1 \leq \frac{u.v}{\|u\|\|v\|} \leq 1$. These inequalities in fact, do hold for all nonzero vectors *u* and *v* in $\mathbb{R}^n$ as a result of Theorem 1.2.1.

Definition 1.2.4 [22]: Two nonzero vectors *u* and *v* in $\mathbb{R}^n$ are said to be ***orthogonal*** (or perpendicular) if $u.v = 0$. A nonempty set of vectors in $\mathbb{R}^n$ is

called an ***orthogonal set*** if all pairs of distinct vectors in the set are orthogonal. An orthogonal set of unit vectors is called an ***orthonormal set***.

Definition 1.2.5 [22]: Le $\mathbb{F}$ be a field. A ***vector space*** V over a field $\mathbb{F}$ is a set with operations of vector addition (+) and scalar multiplication (·) satisfying the following properties:

1. (Closure for Addition): $u + v \in V \qquad \forall\, u, v \in V$
2. (Commutativity for Addition): $u + v = v + u \qquad \forall\, u, v \in V$
3. (Associativity for Addition):$(u + v) + w = u + (v + w) \;\; \forall\, u, v, w \in V$
4. (Existence of Additive Identity): There is a zero vector $0 \in V$ such that $u + 0 = u$
5. (Existence of Additive Inverse): For every vector $u \in V$ there is a vector $-u \in V$ such that $u + (-u) = 0$
6. (Closure for Scalar Multiplication): For any $u \in V$ and $\alpha \in \mathbb{F}$, the vector $\alpha u \in V$
7. (Associativity of scalar multiplication): For any $u \in V$ and $\alpha, \beta \in \mathbb{F}$, then $\alpha(\beta u) = (\alpha\beta)u$
8. (Multiplicative Identity): For any $u \in V$, $1.v = v$, where 1 is the multiplicative identity in $\mathbb{F}$.
9. (Distributive Property):
   a. For any $u, v \in V$ and $\alpha \in K$, then $\alpha(u + v) = \alpha u + \alpha v$
   b. For any $u \in V$ and $\alpha, \beta \in K$, then $(\alpha + \beta)u = \alpha u + \beta v$

Definition 1.2.6 [22]: Let *V* be a vector space and *W* is a subset of *V*. If the subset *W* itself is a vector space under the addition and scalar multiplication defined on *V*, then we say that *W* is a ***subspace*** of the vector space *V*.

Theorem 1.2.2 [22]: If *W* is a set of one or more vectors in a vector space *V*, then *W* is a subspace of *V* if and only if the following conditions hold:

a) If *u* and *v* are vectors in *W*, the $u + v$ is in *W*.

b) If *k* is any scalar and *u* is any vector in *W*, then ku is in *W*.

Definition 1.2.7 [22]: If *w* is a vector in a vector space *V*, let $r \in \mathbb{N}$ then *w* is said to be a ***linear combination*** of the vectors $v_1, v_2, \dots, v_r$ in *V* if *w* can be expressed in the form

$$w = k_1 v_1 + k_2 v_2 + \cdots + k_r v_r = \sum_{i=1}^{r} k_i v_i \qquad (1.2.7)$$

where $k_1, k_2, \dots, k_r$ are scalars. These scalars are called the coefficients of the linear combination.

Definition 1.2.8 [22]: The subspace of a vector space *V* that is formed from all possible linear combinations of the vectors in a nonempty set *S* is called the ***span*** of *S*, and it is said that the vectors in *S* span that subspace.

If $S = \{w_1, w_2, \dots, w_r\}$, then we denote the span of *S* by $span\ \{w_1, w_2, \dots, w_r\}$ or *span* (*S*). That is;

$$span(S) = \left\{ \sum_{i=1}^{r} k_i w_i \right\} \qquad (1.2.8)$$

where $k_1, k_2, \dots, k_r$ are scalars.

Definition 1.2.9 [22]: If $S = \{w_1, w_2, \dots, w_r\}$ is a nonempty set of vectors in a vector space *V*, then the vector equation $k_1 w_1 + k_2 w_2 + \cdots + k_r w_r = 0$ has at least one solution, namely $k_1 = 0,\ k_2 = 0, \dots, k_r = 0$. This solution is called the trivial solution. If this is the only solution, then *S* is said to be a ***linearly independent set***. If there are solutions in addition to the trivial solution, then *S* is said to be a ***linearly dependent set***.

Definition 1.2.10 [22]: Let *V* be a vector space and $S = \{w_1, w_2, \dots, w_r\}$ is a finite set of vectors in *V*, then *S* is called a ***basis*** for *V* if *S* is a linearly independent set such that *S* spans *V*.

Remark 1.2.2: A vector space that cannot be spanned by finitely many vectors is said to be ***infinite-dimensional***, whereas those that can are said to be ***finite-dimensional***.

Theorem 1.2.3 [22]: Let *V* be a vector space and let *U* be a non-empty subset of *V*. If *U* is finite, that is $U = \{u_1, u_2, \dots, u_n\}$ for some $n \in \mathbb{N}$ and $u_i \neq u_j$ for $i \neq j$. Then *U* is a basis for *V* if and only if for each $v \in V$ there exists unique $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{F}$ such that $v = \sum_{i=1}^{n} \alpha_i u_i$.

Definition 1.2.11 [23]: Let *V* be a vector space over the field $\mathbb{F}$, if *U* and *W* are subspaces of *V*; then the ***sum*** of *U* and *W* is the subset of *V* defined as:

$$U + W = \{u + w\,;\, u \in U\,, w \in W\}$$

The vector space *V* is said to be the ***direct sum*** of subspaces *U* and *W* if for every element $v \in V$ there exists unique elements $u \in U$ and $w \in W$ such that $v = u + w$. The direct sum is denoted by $V = U \oplus W$.

Theorem 1.2.4 [23]: Let U and W be subspaces of a vector space V. Then V is a direct sum of U and W if $U + W = V$ and $U \cap W = \{0\}$.

Definition 1.2.12 [22]: Let U and V be two vector spaces. A ***linear transformation*** T is a function $T: U \to V$ having the following properties:

1. $T(u + v) = T(u) + T(v) \; \forall \, u, v \in U$. [Additivity property]
2. $T(cu) = cT(u) \; \forall \, u \in U \; and \; all \; scalars \; c$. [Scalar homogeneity]

In the special case, where $U = V$, the linear transformation T is called a ***linear operator*** on the vector space V.

Definition 1.2.13 [22]: Let V be a real vector space. The ***inner product*** $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ is a function that associates a real number $\langle u, v \rangle$ with each pair of vectors in V such that the following axioms are satisfied for all vectors u,v, and w in V and all scalars k.

1. $\langle u, v \rangle = \langle v, u \rangle$. [Symmetry axiom]
2. $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$. [Additivity axiom]
3. $\langle ku, v \rangle = k \langle u, v \rangle$. [Homogeneity axiom]
4. $\langle v, v \rangle \geq 0 \; and \; \langle v, v \rangle = 0$ if and only if $v = 0$. [Positivity axiom]

A real vector space with an inner product is called a ***real product space***.

Definition 1.2.14 [24]: Let $p \geq 1$ be real number. Then the ***L^p*-space** is the set of all real-valued (or complex-valued) functions f *on* I, such that $\int |f(x)|^p \, dx < \infty \; (on \; I)$. If $f \in L^p(I)$, then its L^p-norm defined as:

$$\|f\|_p = \left(\int |f(x)|^p \, dx \right)^{\frac{1}{p}} \qquad (1.2.9)$$

Example 1.2.1:

a. The space $L^1(I)$ is the set of all integrable functions f on I, with L^1 – norm defined by $\|f\|_1 = \int |f(x)|\, dx < \infty \ (on\ I)$.

b. The space $L^2(I)$ is the set of all square integrable functions f on I, with L^2 –norm defined by $\|f\|_2 = (\int |f(x)|^2\, dx)^{\frac{1}{2}} < \infty \ (on\ I)$.

Remark 1.2.3 [24]:

a. Any continuous or piecewise continuous function with finite number of jump discontinuities on a finite closed interval I is in $L^1(I)$.

b. Any function bounded on an finite interval I is square integrable on I. This includes continuous and piecewise continuous functions with finite jump discontinuities on a finite closed interval.

Theorem 1.2.5 [24]: Let I be a finite interval. If $f \in L^2(I)$, then$f \in L^1(I)$. In other words, a square integrable function on a finite interval is integrable.

Remarks 1.2.4 [24]:

a. The conclusion of theorem 1.2.5 doesn't hold if I is an ***infinite interval***, for example,

$$f(x) = \begin{cases} \frac{1}{x} & , \quad x \geq 1 \\ 0 & , \quad x < 1 \end{cases}$$

$f \in L^2(I)$but $f \notin L^1(I)$.

b. The converse of theorem 1.2.5 is not true, for example,

$f(x) = \frac{1}{\sqrt{x}},\ x \in (0,1)$ is in $L^1((0,1))$ but not in $L^2((0,1))$.

Definition 1.2.15 [25]: The L^2-inner product on $L^2(I)$ is defined as

$$\langle f, g\rangle_{L^2} = \int f(x)\,\overline{g(x)}\,dx\,; f, g \in L^2(I) \qquad (1.2.10)$$

where $\overline{g(x)}$is the complex conjugate of g.

In case where the signal is discrete, we represent the signal as a sequence $X = \{x_n\}_{n=-\infty}^{\infty}$, where each x_n is the numerical value of the signal at the n^{th} time interval $[t_n\,, t_{n+1}]$.

Definition 1.2.16 [26]: A ***signal*** is a real- or complex-valued function of an integer, (n), $n = 0,1, \ldots$. Thus, $f(n)$ is the n^{th} real (or complex) number in the signal, and *n* represents time as an integer sample number. We can express that $f(n)$ as a real discrete-time signal by expressing it as a function mapping every integer to a real number:

$$f: Z \to \mathbb{R}$$

Alternatively, we can write $f(n) \in \mathbb{R}$ for all $n \in Z$. Similarly, a discrete-time *complex* signal is a mapping from each integer to a complex number:

$$f: Z \to C$$

$$f(n) \in C\,, \forall n \in Z$$

Definition 1.2.17 [25]: Let $p \geq 1$ be real number. Then the l^p-space is the set of all real-valued (or complex-valued) sequences of *X*, such that

$$\sum_{n=-\infty}^{\infty} |x_n|^p < \infty$$

The space l^2 is the set of all sequences X , with $\sum_{n=-\infty}^{\infty}|x_n|^2 < \infty$. The inner product on this space is defined by $\langle X, Y\rangle_{l^2} = \sum_{n=-\infty}^{\infty} x_n\overline{y_n}$, where $X = \{x_n\}_{n=-\infty}^{\infty}$ and $Y = \{y_n\}_{n=-\infty}^{\infty}$.

Definition 1.2.18 [27]: The convolution of two continuous functions *f*(x) and *h*(x) of one continuous variable *x* is defined as the following:

$$y(x) = h(x) * f(x) = \int_{-\infty}^{\infty} f(\tau)\, h(x-\tau) d\tau \qquad (1.2.11)$$

In the following sections, the Fourier analysis and different types of some well-known mathematical transformations such as Discrete Fourier Transform (DFT) and Discrete Sine Transform (DST) are discussed in detail.

1.3 Fourier Analysis (FA)

The subject of Fourier analysis (Fourier series and Fourier transform) is an old subject in mathematical analysis and is of great importance to mathematicians, scientist, and engineers alike. It is known from Fourier theory that under certain conditions, a signal (function) can be expressed as the sum of a series of sine and cosine functions. Such a representation is called a Fourier expansion (series) [28]. That is, the basic goal of Fourier series is to take a signal, which will be considered as a function of time variable *t*, and decompose into various frequency components. In other words, transform the signal from time domain to frequency domain, so it can be analyzed and processed. This classical method is used in applications such as storage of visual images on a computer. However, these methods fail to provide efficient representations for certain types of functions which have **discontinuities**. Another problem with this sum is that it is **infinite**. In use, only a finite number of terms can be used. More accuracy requires more terms in the series, but more terms require more time to compute and more space to store.

Definition 1.3.1 [1]: A function *f(x)* is periodic if there exists a nonzero constant ℓ such that $f(x+\ell) = f(x)$ for all values of *x*. The least such ℓ is called the period of the function.

Definition 1.3.2[10]: A *Fourier series* is the representation of a period function, *f*, in terms of a sum of sine and cosine functions.

$$f(x) = \frac{1}{2\pi}a_0 + \frac{1}{\pi}\sum_{n=1}^{\infty}(a_n \cos(nx) + b_n \sin(nx)) \qquad (1.3.1)$$

Where,

$$a_n = \int_0^{2\pi} f(x)\cos(nx)\,dx \quad for\ n = 0,1,2,\ldots \qquad (1.3.2)$$

$$b_n = \int_0^{2\pi} f(x)\sin(nx)\,dx \quad for\ n = 1,2,\ldots \qquad (1.3.3)$$

More generally, the definition can be generalized to a 2ℓ-periodic function *f*(*x*) for any positive real number ℓ by using the trigonometric functions $\cos(\frac{n\pi x}{\ell})$, and $\sin(\frac{n\pi x}{\ell})$ and the following lemma. Thus *f*(*x*) can be represented by its Fourier series,

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty}\left(a_n \cos\frac{n\pi x}{\ell} + b_n \sin\frac{n\pi x}{\ell}\right) \qquad (1.3.4)$$

Where the amplitudes a_n*'s* and b_n*'s* are the Fourier series coefficients defined by

$$a_n = \frac{1}{\ell}\int_{-l}^{\ell} f(x)\cos\left(\frac{n\pi x}{\ell}\right)dx \quad for\ n = 0,1,2,\ldots \qquad (1.3.5)$$

$$b_n = \frac{1}{\ell}\int_{-l}^{\ell} f(x)\sin\left(\frac{n\pi x}{\ell}\right)dx \quad for\ n = 1,2,\ldots \tag{1.3.6}$$

Lemma 1.3.1 [25**]:** Suppose f is any 2π -periodic function and c is any real number, then

$$\int_{-\pi+c}^{\pi+c} f(x)dx = \int_{-\pi}^{\pi} f(x)dx \tag{1.3.7}$$

In applications, Fourier series must be truncated, and this truncation will introduce error since the whole sum is no longer being used. Thus, one must try to find a balance between the number of terms one uses and how much error one is willing to accept. In order to achieve more accuracy, a greater number of terms are needed and this will take up more computer time and storage space.

A second problem with Fourier series is that although it represents the frequency of a function well, it does not do as good a job representing that functions localized properties. For example, a function may contain a high peak on an interval while it is small elsewhere. This function could represent a wave packet, which is just a peak traveling from one point to another in a straight line. Before and after the peak the amplitude is almost zero, as shown in Figure 1.3.1. A Fourier series will not do as well when representing this function because the sine and cosine functions that make up the Fourier series are all periodic and thus it is hard to focus in on the local behavior of this wave packet.

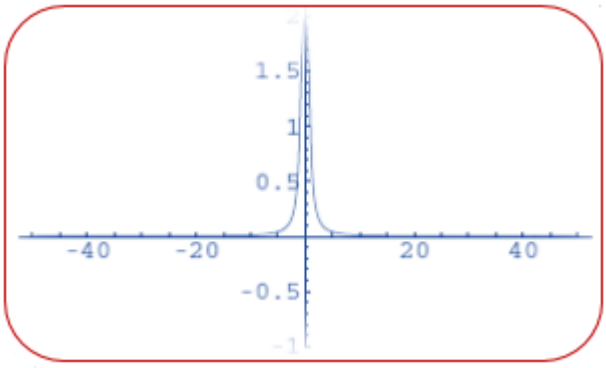


Figure 1.3.1: Example of Wave Traveling From Left to Right [11]

These were problems that mathematicians as well as physicists had to deal with until the 1980's when a new type of series called *wavelet series* was introduced. These new series were developed to get around some of the problems associated with Fourier series.

For example the reason it take Fourier series many terms to represent localized properties is because they depend on a single basis (sines and cosines) which represent frequencies well but whose support is not localized. Wavelet series give us an infinite number of new bases to choose from so we can choose the "*best basis*" for a function. Thus wavelet representations may take fewer terms to represent the same function to the same accuracy as the Fourier series.

Lemma 1.3.2 [25]: (The Riemann-Lebesgue Lemma)

Suppose f is piecewise continuous function on the interval $[a, b]$, Then

$$\lim_{n\to\infty} \int_a^b f(x)\sin(nx)dx = \lim_{n\to\infty} \int_a^b f(x)\cos(nx)\,dx = 0$$

There are two consequences of this theorem one of them is that only the first few terms in the Fourier series are the most important, since they

contribute more to the sum which means that only finite number of terms can be used to approximate the function. This is especially important in ***data compression***. Another one is used to proof the convergence of the Fourier series.

An interesting question about Fourier series is: When is an arbitrary function equal to its Fourier series and in what sense does that Fourier series converge? The answer lies in the notation of a ***complete orthonormal system***.

Definition 1.3.3 [24]: Given a collection of functions $\{g_n(x)\}_{n\in N} \in L^2(I)$, the ***span*** of $\{g_n(x)\}_{n\in N}$ denoted by $span\{g_n(x)\}_{n\in N}$ is the collection of all finite linear combinations of the elements of $\{g_n(x)\}_{n\in N}$. The ***closure*** of $span\{g_n(x)\}_{n\in N}$ denoted by $\overline{span}\{g_n(x)\}$ and it is defined as follows:
A function $f \in \overline{span}\{g_n(x)\}$ if for every $\varepsilon > 0$ there is a function $g(x) \in span\{g_n(x)\}_{n\in N}$ such that $\|f - g\|_2 < \varepsilon$.

Definition 1.3.4 [24]: If every function in $L^2(I)$ is in $\overline{span}\{g_n(x)\}$ where $\{g_n(x)\}_{n\in N}$ is orthonormal system, then we say that $\{g_n(x)\}_{n\in N}$ is ***complete*** on I, this means that every function in $L^2(I)$ is equal to its Fourier series in $L^2(I)$. A complete orthonormal system is called an ***orthonormal basis***.

Remark 1.3.1 [29]: An orthonormal set is said to be ***complete*** if no additional non-zero orthogonal vector exists which can be added to the set.

Example 1.3.1 [29]: One of the best-known discrete transforms is the *Fourier series expansion.* The basis functions are the complex exponentials

$$g_n(t) = \frac{e^{in\pi t}}{\sqrt{2}} , \qquad n = 0, \pm 1, \pm 2, \dots$$

which form a ***complete orthonormal set***.

The interval considered is $T = [-1,1]$. The weighting function is $h(t) = 1$. It is noted that any finite interval can be mapped onto the interval $T = [-1,1]$.

In conclusion, the Fourier series shows that any periodic function can be decomposed as an infinite sum of periodic functions (sines and cosines). This decomposition makes it easy an analysis of the frequencies present on the function *f*. There exists a fundamental frequency ω, and all of the other frequencies are integer multiples $\omega_i, i \in Z$, of this fundamental frequency.

The above description of the function *f* worked perfectly well. We were able to obtain an exact representation of the function *f* and this representation completely characterizes f by its frequencies. The only drawback is the fact that *f* was a periodic function.

Now: **Is it possible to extend the above results for non-periodic functions?** The extension of the Fourier calculus to the entire real line leads naturally to the Fourier transform, a powerful mathematical tool for the analysis of non-periodic functions.

In the next section, we motivate the construction by investigating how (rescaled) Fourier series behave as the length of the interval goes to infinity. The resulting Fourier transform maps a function defined on physical space to a function defined on the space of frequencies, whose values quantify the "amount" of each periodic frequency contained in the original function.

The inverse Fourier transform then reconstructs the original function from its transformed frequency components. One of the most important properties of the Fourier transform is that it converts calculus: differentiation and integration into algebra: multiplication and division. The study of Fourier

transform, its strength and limitations, is the starting point of the wavelets analysis.

1.4 Fourier Transform (FT)

The Fourier transform can be thought of as a continuous form of Fourier series. It can be thought as a limiting case of Fourier series.

Let $f(x)$ be a function defined for all $-\infty < x < \infty$. The goal is to construct a Fourier expansion for $f(x)$ in terms of basic trigonometric functions. One evident approach is to construct its Fourier series on progressively longer and longer intervals, and then take the limit as their lengths go to infinity. This limiting process converts the Fourier sums into integrals, and the resulting representation of a function is renamed the ***Fourier transform***. Since we are dealing with an infinite interval, there are no longer any periodicity requirements on the function $f(x)$. Moreover, the frequencies represented in the Fourier transform are no longer constrained by the length of the interval, and so we are effectively decomposing a quite general, ***non-periodic*** function into a continuous superposition of trigonometric functions of all possible frequencies.

The Fourier Transform is an important mathematical transform that is used widely in many application areas such as applied mathematics, physics, engineering, and computer science, especially in image processing.

A time domain function can be either continuous or discrete, and it can be either periodic or non periodic. The combination of these two features generates the types of Fourier transforms.

The Fourier series decomposes a signal (function) on $[-\pi, \pi]$ into components that vibrate at integer frequencies. By contrast, the FT decomposes a signal defined on an infinite time interval into a ω-frequency component, where ω can be any real (or even complex number). Since the support of the bases function $e^{-i\omega x}$ covers the whole time domain (infinite support), the Fourier transform of $f(x)$ denoted by $\mathcal{F}\{f(x)\}$, depends on the values of $f(x)$ for all times. This makes the Fourier transform a global transform that cannot analyze local or transient properties of the original function (signal) $f(x)$.

The time and frequency domains are alternative ways of representing signals (functions). The FT is the mathematical relationship between these two representations [27]. In image compression application, FT decomposes such an image function into a set of orthogonal functions and converts the spatial (time) intensity image into its frequency domain [30].

There are two types of Fourier transforms, which are ***the continuous Fourier transform*** (*CFT*), and ***the discrete Fourier transform*** (*DFT*). We shall give the definition of FT in one-dimensional and two-dimensional spaces of continuous function (CFT) and the corresponding discrete Fourier transform(DFT).

1.4.1 Continuous Fourier Transform (CFT)

Let $f(x)$ be a continuous function of a continuous variable, x. The one – dimensional (1-D) CFT is given by the Equation:

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\omega x} dx \qquad (1.4.1.1)$$

where ω is also a continuous variable.

The inverse of CFT in 1-D is given by the Equation:

$$f(x) = \int_{-\infty}^{\infty} \mathcal{F}(\omega)\, e^{i2\pi\omega x} d\omega \qquad (1.4.1.2)$$

Where $f(x)$ is a given continuous–time signal and $\mathcal{F}(\omega)$ is a continuous Fourier transform of it and ω is the frequency variable [29].

However, the CFT in two-dimensional (2-D) is given by:

$$\mathcal{F}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i2\pi(ux+vy)} dxdy \qquad (1.4.1.3)$$

and its inverse is given by:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{F}(u,v)\, e^{i2\pi(ux+vy)} dudv \qquad (1.4.1.4)$$

Where $f(x,y)$ be a given continuous–time signal (function) of two continuous variables *x*,*y* ; and $\mathcal{F}(u,v)$ be a continuous Fourier transform of it and *u*,*v* are the frequency variables [31].

The Fourier transform is generally complex; by using Euler's formula we can express [30]:

$$\mathcal{F}(\omega) = R(\omega) + i\, I(\omega) = |\mathcal{F}(\omega)| e^{i\theta(\omega)} \qquad (1.4.1.5)$$

$$\mathcal{F}(\omega) = \int_{-\infty}^{\infty} f(x)\, [cos(2\pi\omega x) - i\, sin(2\pi\omega x)] dx \qquad (1.4.1.6)$$

$$f(x) = \int_{-\infty}^{\infty} \mathcal{F}(\omega)\,[cos(2\pi\omega x) + i\,sin(2\pi\omega x)]d\omega \qquad (1.4.1.7)$$

Where $R(\omega)$ and $I(\omega)$ are the real and imaginary components of $\mathcal{F}(\omega)$, respectively, and

$$|\mathcal{F}(\omega)| = \sqrt{R^2(\omega) + I^2(\omega)} \qquad (1.4.1.8)$$

$$\theta(\omega) = tan^{-1}(I(\omega)/R(\omega)) \qquad (1.4.1.9)$$

$|\mathcal{F}(\omega)|$ is called the Fourier spectrum of *f*(x) (the magnitude of *F(u,v)*), $\mathcal{F}(\omega)^2$ is called the energy (or power) of spectrum, and $\theta(\omega)$ is called the phase angle [30].

1.4.2 Discrete Fourier Transform (DFT)

The input to the DFT is a sequence of numbers rather than a continuous function of time *f(x)*. The sequence of numbers usually results from periodically sampling a continuous signal *f(x)* at an interval of Δx [30].

The DFT is derived from a continuous-time function *f(x)* using N samples taken at times $x = 0, \Delta x, 2\Delta x, \dots, (N-1)\Delta x$ where *x* is the sampling interval. These N samples of *f(x)* form a data sequence:

$$\{f(x\ \ ), f(x\ \ + \Delta x), f(x\ \ + 2\Delta x), \dots, f(x\ \ + [N-1]\Delta x)\}$$

It will be convenient to use *x* as either a continuous or a discrete variable, depending on the context of discussion. Therefore, $f(x) = f(x_0 + x\Delta x)$, where *x* now assumes the discrete values $0,1,2,\dots,N-1$ [32]. The simplified notations $\{f(0), f(1), f(2), \dots, f(N-1)\}$ mean N uniformly

spaced samples of a continuous function. Figure 1.4.2.1 illustrations this concept [30].

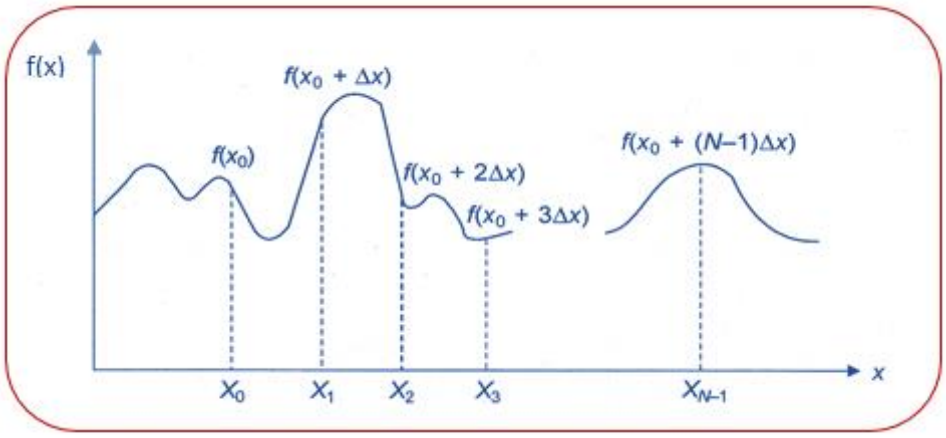


Figure 1.4.2.1: Sampling a Continuous Function[32]

The DFT pair in one – dimensional (1-D) is given by:

$$F(\omega) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, e^{\frac{-2i\pi\omega x}{N}}, \omega = 0,1,\dots,N-1, \qquad (1.4.2.1)$$

The inverse of DFT in (1-D) is defined as:

$$f(x) = \sum_{\omega=0}^{N-1} F(\omega)\, e^{\frac{2i\pi\omega x}{N}}, x = 0,1,\dots,N-1, \qquad (1.4.2.2)$$

The values $\omega = 0,1,2,\dots,N-1$ in the DFT given in Eq. (1.4.2.1) correspond to samples of the continuous transform at values $0, \Delta\omega, 2\Delta\omega, \dots, (N-1)\Delta\omega$ [32]. The following examples calculate the four-points DFT of the aperiodic sequence $x[k]$ and its inverse of length $N = 4$.

Example 1.4.2.1:

Calculate the four-point DFT of the aperiodic sequence $x[k]$ of length $N = 4$, which is defined as follows:

$$x[k] = \begin{cases} 2 & k = 0 \\ 3 & k = 1 \\ -1 & k = 2 \\ 1 & k = 3 \end{cases}$$

Solution :

Using Eq. (1.4.2.1), the four-point DFT of $x[k]$ is given by

$$X[r] = \frac{1}{4}\sum_{k=0}^{3} x[k]e^{-j(\frac{2\pi kr}{4})}$$

$$= \frac{1}{4}\left(2 + 3.\ e^{-j(\frac{2\pi r}{4})} - 1.\ e^{-j\left(\frac{2\pi(2)r}{4}\right)} + 1.\ e^{-j\left(\frac{2\pi(3)r}{4}\right)}\right),$$

for $0 \leq r \leq 3$. On substituting different values of r, we obtain

$r = 0$ $\quad X[0] = \frac{1}{4}(2 + 3 - 1 + 1) = \frac{5}{4}$;

$r = 1$ $\quad X[1] = \frac{1}{4}\left(2 + 3\ e^{-j\left(\frac{2\pi}{4}\right)} - 1\ e^{-j\left(\frac{2\pi(2)}{4}\right)} + 1\ e^{-j\left(\frac{2\pi(3)}{4}\right)}\right)$

$$= \frac{1}{4}(2 + 3(-j) - 1(-1) + 1(j)) = \frac{3 - 2j}{4}$$

$r = 2$ $\quad X[2] = \frac{1}{4}\left(2 + 3\ e^{-j\left(\frac{2\pi(2)}{4}\right)} - 1\ e^{-j\left(\frac{2\pi(2)(2)}{4}\right)} + 1\ e^{-j\left(\frac{2\pi(3)(2)}{4}\right)}\right)$

$$= \frac{1}{4}\big(2 + 3(-1) - 1(1) + 1(-1)\big) = \frac{-3}{4}$$

$r = 3$ $\quad X[3] = \frac{1}{4}\left(2 + 3\ e^{-j\left(\frac{2\pi(3)}{4}\right)} - 1\ e^{-j\left(\frac{2\pi(2)(3)}{4}\right)} + 1\ e^{-j\left(\frac{2\pi(3)(3)}{4}\right)}\right)$

$$= \frac{1}{4}\big(2 + 3(j) - 1(-1) + 1(-j)\big) = \frac{3 + 2j}{4}$$

Example 1.4.2.2:

Calculate the inverse DFT of

$$X[r] = \begin{cases} \frac{5}{4} & r = 0 \\ \frac{3-2j}{4} & r = 1 \\ \frac{-3}{4} & r = 2 \\ \frac{3+2j}{4} & r = 3 \end{cases}$$

Solution:

Using Eq. (1.4.2.2), the inverse DFT of $X[r]$ is given by

$$x[k] = \sum_{k=0}^{3} X[r] e^{j\left(\frac{2\pi kr}{4}\right)}$$

$$= \left[\frac{5}{4} + \frac{3-2j}{4} e^{j\left(\frac{2\pi k}{4}\right)} - \frac{3}{4} e^{j\left(\frac{2\pi(2)k}{4}\right)} + \frac{3+2j}{4} e^{j\left(\frac{2\pi(3)k}{4}\right)}\right]$$

for $0 \leq k \leq 3$. On substituting different values of k, we obtain

$$x[0] = \frac{1}{4}[5 + (3-2j) - 3 + (3+2j)] = 2\,;$$

$$x[1] = \frac{1}{4}\left[5 + (3-2j)e^{j\left(\frac{2\pi}{4}\right)} - 3e^{j\left(\frac{2\pi(2)}{4}\right)} + (3+2j)e^{j\left(\frac{2\pi(3)}{4}\right)}\right]$$

$$= \frac{1}{4}[5 + (3-2j)(j) - 3(-1) + (3+2j)(-j)] = 3;$$

$$x[2] = \frac{1}{4}\left[5 + (3-2j)e^{j\left(\frac{2\pi(2)}{4}\right)} - 3e^{j\left(\frac{2\pi(2)(2)}{4}\right)} + (3+2j)e^{j\left(\frac{2\pi(3)(2)}{4}\right)}\right]$$

$$= \frac{1}{4}[5 + (3-2j)(-1) - 3(1) + (3+2j)(-1)] = -1;$$

$$x[3] = \frac{1}{4}\left[5 + (3-2j)e^{j\left(\frac{2\pi(3)}{4}\right)} - 3e^{j\left(\frac{2\pi(2)(3)}{4}\right)} + (3+2j)e^{j\left(\frac{2\pi(3)(3)}{4}\right)}\right]$$

$$= \frac{1}{4}[5 + (3-2j)(-j) - 3(-1) + (3+2j)(j)] = 1$$

Remark 1.4.2.1: The DFT can be expressed in matrix form, as the following: By expanding Eq.(1.4.2.1),

$$F(\omega) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\, e^{\frac{-2i\pi\omega x}{N}} \qquad \omega = 0,1,\dots,N-1$$

An alternative representation for computing the DFT in term of the time and frequency indices (ω,x) , the resulting equations are expressed as follows [33]:

$$\left.\begin{aligned}
F[0] &= \tfrac{1}{N}f[0] + \tfrac{1}{N}f[1] + \tfrac{1}{N}f[2] + \cdots + \tfrac{1}{N}f[N-1] \\
F[1] &= \tfrac{1}{N}f[0] + \tfrac{1}{N}f[1]e^{-i\left(\frac{2\pi}{N}\right)} + \tfrac{1}{N}f[2]e^{-i\left(\frac{4\pi}{N}\right)} + \cdots + \tfrac{1}{N}f[N-1]e^{-i\left(\frac{2(n-1)\pi}{N}\right)} \\
F[2] &= \tfrac{1}{N}f[0] + \tfrac{1}{N}f[1]e^{-i\left(\frac{4\pi}{N}\right)} + \tfrac{1}{N}f[2]e^{-i\left(\frac{8\pi}{N}\right)} + \cdots + \tfrac{1}{N}f[N-1]e^{-i\left(\frac{4(n-1)\pi}{N}\right)} \\
&\vdots \\
F[N-1] &= \tfrac{1}{N}f[0] + \tfrac{1}{N}f[1]e^{-i\left(\frac{2(N-1)\pi}{N}\right)} + \tfrac{1}{N}f[2]e^{-i\left(\frac{4(N-1)\pi}{N}\right)} + \cdots + \tfrac{1}{N}f[N-1]e^{-i\left(\frac{2(N-1)(N-1\pi}{N}\right)}
\end{aligned}\right\} \qquad (1.4.2.3)$$

In the matrix-vector format they are given by

$$\underbrace{\begin{bmatrix} F[0] \\ F[1] \\ F[2] \\ \cdot \\ \cdot \\ \cdot \\ F[N-1] \end{bmatrix}}_{DFT\ vector:\vec{F}} = \underbrace{\frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-i(2\pi/N)} & e^{-i(4\pi/N)} & \dots & e^{-i(2(N-1)\pi/N)} \\ 1 & e^{-i(4\pi/N)} & e^{-i(8\pi/N)} & \dots & e^{-i(4(N-1)\pi/N)} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \ddots & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & e^{-i(2(N-1)\pi/N)} & e^{-i(4(n-1)\pi/N)} & \cdots & e^{-i(2(N-1)(N-1)\pi/N)} \end{bmatrix}}_{DFT\ matrix\ F} \underbrace{\begin{bmatrix} f[0] \\ f[1] \\ f[2] \\ \cdot \\ \cdot \\ \cdot \\ f[N-1] \end{bmatrix}}_{Signal\ vector:\vec{f}} \qquad (1.4.2.4)$$

Eq. (1.4.2.4) shows that the DFT coefficients $F(\omega)$ can be computed by left multiplying the DT sequence $f(x)$, arranged in a column vector x in a scending order with respect to the time index ω, by the DFT matrix F.

Similarly, the expression for the inverse DFT given in Eq. (1.4.2.2) can be expressed as follows [33]:

$$\underbrace{\begin{bmatrix} f[0] \\ f[1] \\ f[2] \\ \cdot \\ \cdot \\ \cdot \\ f[N-1] \end{bmatrix}}_{Signal\ vector:\vec{f}} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{i(2\pi/N)} & e^{i(4\pi/N)} & \dots & e^{i(2(N-1)\pi/N)} \\ 1 & e^{i(4\pi/N)} & e^{i(8\pi/N)} & \dots & e^{i(4(N-1)\pi/N)} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & \ddots & \cdot \\ 1 & e^{i(2(N-1)\pi/N)} & e^{i(4(n-1)\pi/N)} & \cdots & e^{i(2(N-1)(N-1)\pi/N)} \end{bmatrix}}_{DFT\ matrix\ G=F^{-1}} \underbrace{\begin{bmatrix} F[0] \\ F[1] \\ F[2] \\ \cdot \\ \cdot \\ \cdot \\ F[N-1] \end{bmatrix}}_{DFT\ vector\ \vec{F}:} \quad (1.4.2.5)$$

The disadvantage application of the *DFT* is the difficulty of using this transformation due to the requirement to process both real and imaginary components as shown in Eq. (1.4.2.4).

For image, an $N \times N$ image can be decomposed into a weighted sum of 2-D sinusoidal term. The 2-D (DFT) equation is given by [32]:

$$F(u,v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) e^{-2j\pi(\frac{ux}{N}+\frac{vy}{N})} \qquad u,v = 0,1,\dots,N-1 \quad (1.4.2.6)$$

The inverse of DFT in 2-D is defined as:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} f(u,v) e^{2j\pi(\frac{ux}{N}+\frac{vy}{N})}, \qquad x,y = 0,1,\dots,N-1 \qquad (1.4.2.7)$$

The application of two-dimensional DFT to image arrays produces a two-dimensional spectrum of the data where highly correlated image data have small energy at high spatial frequencies [34]. One important property of the DFT is its *separability* in which the two-dimensional basis image can be decomposed into two product terms. If the basis images are separable, then the result can be found by successive application of two, one-dimensional transforms. This can be done by first separating the basis image term into product terms as follows:

$$e^{\frac{-j2\pi(ux+vy)}{N}} = e^{\frac{-j2\pi ux}{N}} e^{\frac{-j2\pi vy}{N}} \quad (1.4.2.8)$$

Eq.(1.4.2.6) can be written as

$$F(u,v) = \frac{1}{N^2}\sum_{x=0}^{N-1} e^{\frac{-j2\pi ux}{N}} \sum_{y=0}^{N-1} f(x,y) e^{\frac{-j2\pi vy}{N}}, \qquad u,v = 0,1,\dots,N-1, \quad (1.4.2.9)$$

As it can be noticed from Eq. (1.4.2.9), the advantage of the separability property is that $F(u, v)$ or $f(x, y)$ can be obtained in two steps by successive application of the one-dimensional Fourier transform or it's inverse. One of the main reasons that the *DFT* has become an important tool in signal processing is that it can be implemented using *Fast Fourier Transforms* (*FFT*) [31].

The inverse *DFT* generates samples which are periodic extension of the first N samples [35], that is,

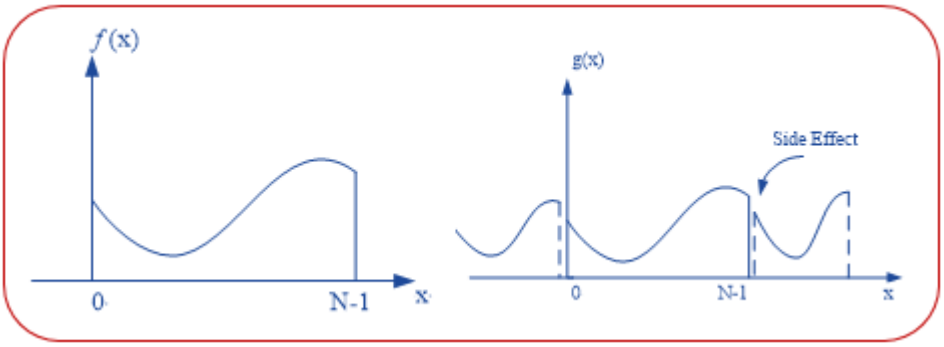
$$g(x) = f(x), \quad x = 0,1,2,\dots,N-1, \quad (1.4.2.10)$$

and

$$g(x) \neq f(x), \quad x = N, N+1, N+2, \dots\dots\dots, \quad (1.4.2.11)$$

$$g(x+N) = g(x) \quad (1.4.2.12)$$

This periodicity in *DFT* causes discontinuities at the beginning and end of each block. This effect can be seen in Figure 1.4.2.2 .



(a) **(b)**

Figure 1.4.2.2: Side Effect in DFT

A big problem in using Fourier transform is that it is not localized. It does not reveal how the signal's frequency contents vary with time. This localization is of great importance in several applications, and in particular in the problem of function representation. The existence of a certain frequency detects the presence of some object, and the localization of a frequency allows us to determine the object position. Because the temporal structure of the signal is not detected, the feature of the Fourier transform is limited; specifically, it is not suitable for analyzing non-stationary signals.

In order to overcome the limitation of the Fourier transform, the ***Short-Time Fourier Transform*** (*STFT*) was introduced by Dennis Gabor in 1946 **[**36**]**. The *STFT* is a modification in the definition of the Fourier transform to obtain a transform with better localization properties in the time-frequency domain. It consists of an analysis window of specific length that glides during the signal along the time axis to perform a "time-localized" Fourier transform.

1.4.3 Short Time Fourier Transform (STFT)

The Short-Time Fourier Transform (STFT) or windowed Fourier Transform, is the simplest time frequency representation. Instead of transforming the whole signal, all at once using the Fourier Transform, it is transformed on a (block-by-block) basis using a moving time-window, centered at the time instant τ [37].

The Short-Time Fourier Transform (STFT) employs a sliding window function *g(t)* that is centered at time τ. For each certain τ, a time-localized Fourier transform is performed on the signal *x(t)* in the window. If the window shape is given by *g*(*t*), the *STFT* is formally given by [36]

$$F(\omega,\tau) = \int x(t)g(t-\tau)e^{-j\omega t}\,dt \qquad (1.4.3.1)$$

This means that the signal (function) *x*(*t*) which is to be analyzed is multiplied with an analysis window $g(t-\tau)$ which is centered at time τ **.** Then, the window is movable by τ along the time line, and another Fourier transform is performed. Through such sequential operations, Fourier transform of the whole signal can be performed. The signal segment in the window function is assumed to be approximately stationary. As a result, the Short-Time Fourier Transform (STFT) decomposes a time domain signal into a 2D time-frequency representation, and alteration of the frequency content of that signal within the window function are revealed, as illustrated in Figure 1.4.3.1 [36].

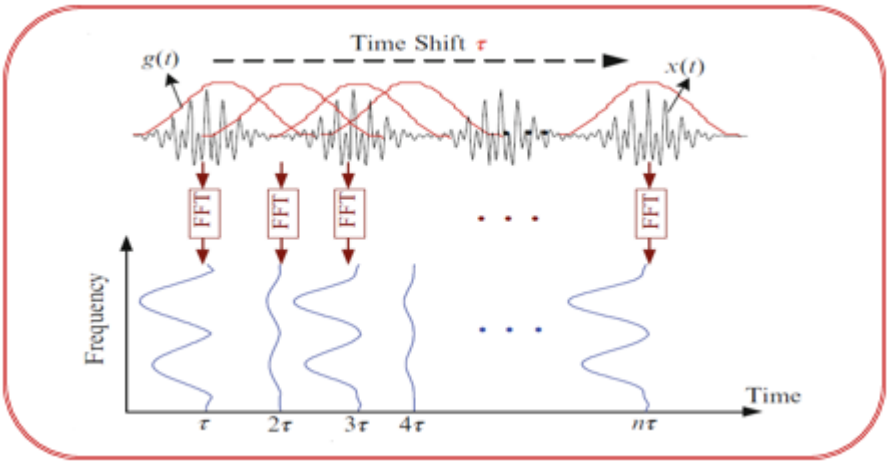


Figure 1.4.3.1: Illustration of Short-Time Fourier Transform on The Test Signal x(t)[36]

It should be known that the choice of the window function directly impact the time and frequency resolutions of the analysis result. While higher resolution in general provides best separation of the constituent components within a signal, the time and frequency resolutions of the Short-Time Fourier Transform (STFT) technique cannot be selection arbitrarily at the same time, according to the uncertainty principle **[**36**].**

1.5 Discrete Sine Transform (DST)

The Discrete Sine Transform (DST) is a member of a family of sinusoidal unitary transforms. A sinusoidal unitary transform is an invertible linear transform whose kernel describes a set of complete, orthogonal discrete cosine and/or sine basis functions. The DST has found applications in digital signal and image processing and particularly in transform coding systems for data compression/decompression. It was firstly introduced into the digital image processing by Jain [38]. It is found that in the case of image with high correlation coefficient, DST gives lower bit rate **[**39**].**

The most common DST definition of a 1-D sequence of length N is given by:

$$S(u) = \propto (u) \sum_{x=0}^{N-1} f(x)\, sin\left[\frac{\pi(2x+1)u}{2N}\right], \qquad u = 0,1,\dots,N-1 \qquad (1.5.1)$$

Similarly, the inverse transformation is defined as

$$f(x) = \sum_{u=0}^{N-1} \propto (u) S(u)\, sin\left[\frac{\pi(2x+1)u}{2N}\right], \qquad x = 0,1,\dots,N-1 \qquad (1.5.2)$$

In both equations of DST and its inverse; $\propto (u)$ is defined as

$$\propto (u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u \neq 0 \end{cases} \qquad (1.5.3)$$

In addition, the 2-D DST is a direct extension of the 1-D case and is given by

$$S(u,v) = \propto (u) \propto (v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)\, \mathrm{Sin}\left[\frac{\pi(2x+1)u}{2N}\right] \mathrm{Sin}\left[\frac{\pi(2y+1)v}{2N}\right], \qquad (1.5.4)$$

For $u, v = 0,1,2,\ldots,N-1$ and $\propto(u)$ and $\propto(v)$ are defined in Eq.(1.5.3). The inverse transform is defined as

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \propto(u) \propto(v) S(u,v) \operatorname{Sin}\left[\frac{\pi(2x+1)u}{2N}\right] \operatorname{Sin}\left[\frac{\pi(2y+1)v}{2N}\right], \qquad (1.5.5)$$

for $x, y = 0,1,2,\ldots,N-1$.

Besides being real, orthogonal, and separable, the DST properties are relevant to data compression and fast algorithms for its computation have proved to be of practical value. Figure 1.5.1 depicts the 64 basis image of the 2-D DST of an 8×8 image matrix [38].

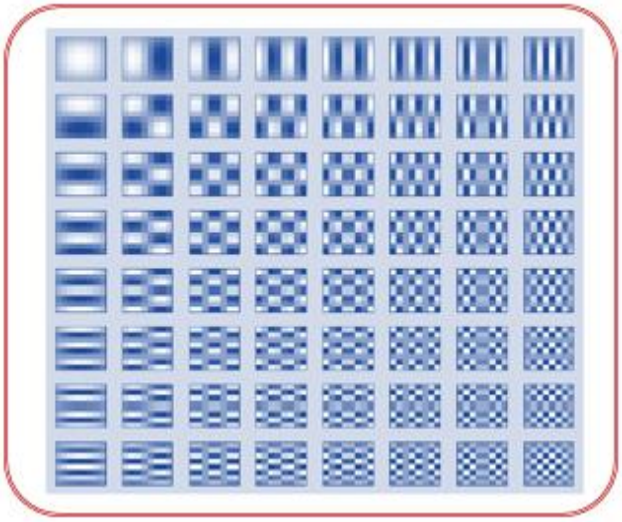


Figure 1.5.1: The 64 Basis Images of the DST in Two Dimensions[1]

CHAPTER TWO

WAVELET TRANSFORM AND MULTIRESOLUTION ANALYSIS

2.1 Introduction

In this chapter we introduce the basic idea of the wavelet transform (WT) and its properties, as well as, its applications. Wavelet application in the field of image compression will be discussed so as to observe how wavelet is implemented to be applied to an image in the process of compression, and also how mathematical aspects of wavelet affect the compression process and the results of it. The insights of how wavelets in mathematics are implemented in a way to fit the engineering model of image compression will be studied. Then the concept of Multiresolution Analysis (MRA) and Singular Value Decomposition (SVD) are explained.

2.2 Wavelet Transform (WT)

For the analysis of a non-stationary (transient) signal, using the FT does not give suitable results since sinusoids are smooth and predictable and are good at describing constant frequencies (stationary) signals. Wavelets are better at describing irregularities, and other events that start and stop within the signal. Therefore, wavelet analysis (or the wavelet transform) is a successful approach to the problem of analyzing a non-stationary signal both in time and in frequency [11].

Wavelets are mathematical functions defined over a finite interval and having an average value of zero that transform data into different frequency components, representing each component with a resolution matched to its scale [40]. It was first introduced by A. Grossmann and J. Morlet in 1984 [41]. The basic idea of the (WT) is to represent any arbitrary function $f(t)$ as a superposition of a set of such wavelets or basis functions. These basis

functions are obtained from a single wavelet called ***mother wavelet*** $\psi(t)$ because all other wavelet functions within the family are obtained by dilating and translating of $\psi(t)$ by amounts s and τ respectively as given bellow [42]:

$$\psi_{s,\tau}(t) = \left\{\psi\left(\frac{t-\tau}{s}\right) ;\ (s,\tau) \in \mathbb{R}^{+} \times \mathbb{R}\right\} \qquad (2.2.1)$$

The process of changing the two parameters s and τ that result in the basis functions are shown in Figure 2.2.1.

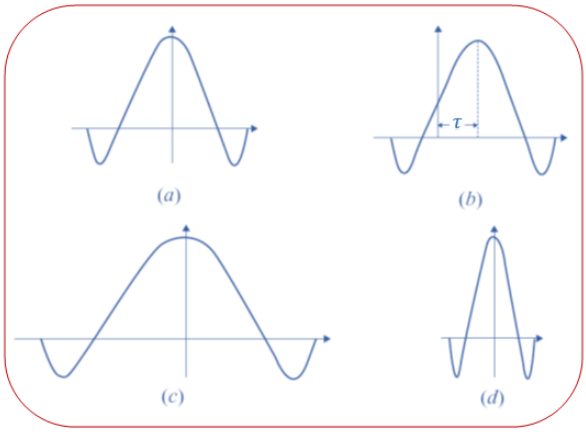


Figure 2.2.1: Effect of Time Dilation and Translation on the Mother Wavelet[42]

(a) Mother wavelet $\psi(t) = \psi_{1,0}$; $s = 1, \tau = 0$.**(b)** Wavelet $\psi_{1,\tau}(t)$; $s = 1, \tau \neq 0$.
(c) Wavelet $\psi_{2,0}(t)$; $at\ scale\ s = 2, \tau = 0$.**(d)** Wavelet $\psi_{0.5,0}(t)$; $at\ scale\ s = 0.5, \tau = 0$.

These wavelets are distinguishing by compactly supported functions defined over a finite interval and having an average value of zero, and that leads to efficient implementation. The mother wavelet $\psi(t)$ is the function with zero translation and a dilation of one.

In wavelet transform the basis functions are wavelets. Wavelets tend to be irregular and symmetric. All wavelet functions, $\psi(2st-\tau)$, are derived from a single mother wavelet, $\psi(t)$. This wavelet is a small wave or pulse like the one shown in Figure 2.2.2 [40].

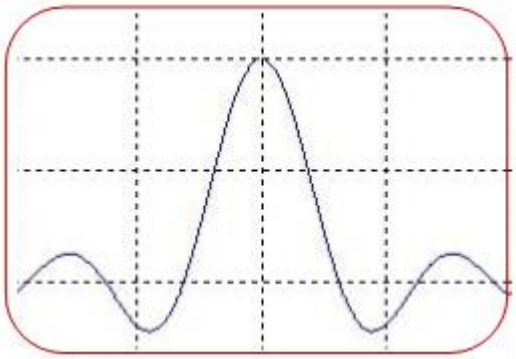


Figure 2.2.2: Mother Wavelet $\psi(t)$ [43]

Normally it starts at time $t = 0$ and ends at $t = T$. The ***shifted*** wavelet $\psi(t-\tau)$ starts at $t=\tau$ and ends at $t=\tau+T$. The ***scaled*** wavelets $\psi(2st)$ start at $t = 0$ and end at $t = T/2s$. Their graphs are $\psi(t)$ compressed by the factor of $2s$ as shown in Figure 2.2.3. For example, when $s = 1$, the wavelet is shown in Figure 2.2.3(a). If $s = 2$ and 3, they are shown in (b) and (c), respectively [40]

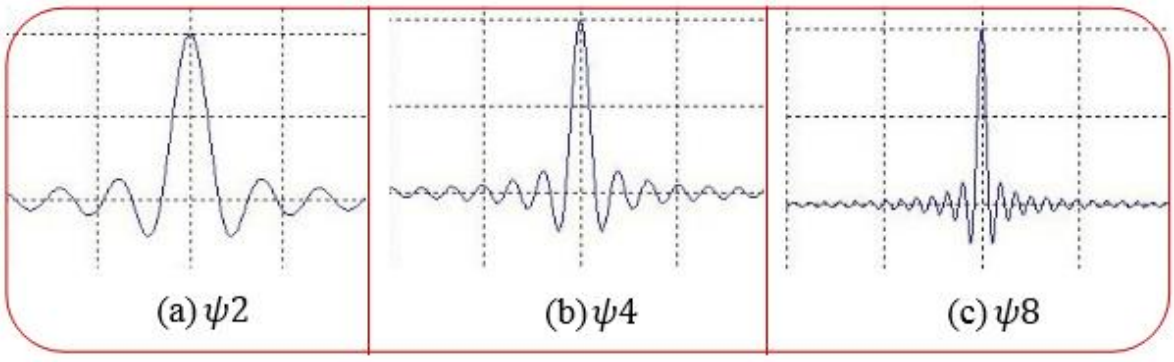


Figure 2.2.3: Scaling the Wavelets [43]

By a dilation we mean a scaling of the argument; so, given any function $\psi(t)$ and a parameter $s>0$, $\psi(\frac{t-\tau}{s})$ is a dilation of $\psi(t)$. Consequently, a dilation of a function corresponds to either a spreading out or contraction of the function. The factor $\frac{\mathbf{1}}{\sqrt{|s|}}$ is introduced with $\psi_{s,\tau}(t)$ equation to keep the energy of the mother wavelets constant that it yields normalization necessary to have an orthonormal wavelet basis.

The translation simply means a shift of the argument along the real axis, that is, given, the translation of $\psi(t)$ by τ is $\psi(t-\tau)$. For any analyzing wavelet $\psi(t)$ we thus define a family of functions $\psi_{s,\tau}(t)$ by the dilations and translations of $\psi(t)$ as given by the Eq. (2.2.2):

$$\psi_{s,\tau}(t) = \left\{\frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) ;\ s,\tau \in \mathbb{R}^{+} \times \mathbb{R}\right\} \qquad (2.2.2)$$

Each $\psi_{s,\tau}$ is called a wavelet.

Where, $\psi(t)$ is a mother wavelet,

s is coefficient of expansions or scaling (dilation),

τ is a coefficient of translation.

The translation τ and dilation s allow the wavelet transform to be localized in time and frequency. Also, wavelet basis functions can represent functions with discontinuities and spikes in a more compact way than sine and cosine [30].

The wavelets are called ***orthogonal*** when their inner products are zero. The smaller the scaling factor is, the wider the wavelet is. Wide wavelets are comparable to low-frequency sinusoids and narrow wavelets are comparable to high frequency sinusoids [40].

The following are some important definitions which are related to the wavelet transform.

Definitions 2.2.1:

- **Translation[11]**

  Let $f: \mathbb{R} \to \mathbb{R}$ and $\in \mathbb{R}$, then the translation $T_y: \mathbb{R} \to \mathbb{R}$, is an operator defined by

$$T_y f(x) = f(x - y) \tag{2.2.3}$$

- **t-Dilation[11]**

  Let $f: \mathbb{R} \to \mathbb{R}$ and $t > 0$, then the t-dilation $f_t: \mathbb{R} \to \mathbb{R}$ is defined by

$$f_t(x) = \frac{1}{t} f\left(\frac{x}{t}\right) \tag{2.2.4}$$

- **Orthogonally[44]**

  The expansion functions $\{\psi_{s,\tau}(x)\}$ form an orthogonal of basis functions:

$$\langle \psi_{s,\tau}(x), \psi_{s_1,\tau_1}(x) \rangle = \int \psi_{s,\tau}(x)\, \psi_{s_1,\tau_1}(x) dx$$
$$= \begin{cases} 1 & for \quad s = s_1, \ \tau = \tau_1 \\ 0 & for \quad s \neq s_1, \ \tau \neq \tau_1 \end{cases} \tag{2.2.5}$$

  Where, $\psi_{s,\tau}(x)$: is a decomposition wavelet,

  s and s_1: are dilation (scaling) factors,

  τ and τ_1: are translating (shifting) factors.

- **Compact Support[1]**

We say that $\psi(x)$ has compact support on interval *I*, if it has zero values (vanish) outside this interval, so it is limited in time domain.

➢ **Admissibility Condition[1]**

The inverse wavelet transform is held if the wavelet function is satisfied for any $\psi(x)$, then:

$$c = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty \qquad (2.2.6)$$

Where $\Psi(x)$ is the Fourier transform of the fundamental mother wavelet $\psi(x)$, $\Psi(\omega) = \int_{-\infty}^{\infty} \psi(x)\, e^{-j\omega x} dx$, and c represents how closely correlated of the wavelet with this section of the signal, so the higher value of c is the more of similarity. Then c is positive and finite, in most cases, this simply means that $\Psi(0) = 0$ and $\Psi(\omega) \to 0\ as\ \omega \to \infty$ fast enough to make $c < \infty$.

The requirement that c be positive and finite imposes another restriction of the choice of wavelet.

➢ **Normalized[44]**

A wavelet function is defined as a function with a zero average

$$\int_{-\infty}^{\infty} \psi(x)\, dx = 0 \ \ and \ \ \|\psi(x)\| = 1 \qquad (2.2.7)$$

It is normalized, and centered in the neighborhood for $x = 0$.

➢ **Series Expansion[45]**

A function or signal $f(x)$ can often be better analyzed as a linear combination of expansion function.

$$f(x) = \sum_{k} \alpha_k \emptyset_k(x) \qquad (2.2.8)$$

Where, k is an integer index of the finite or infinite sum,

α_k are real valued expansion coefficients,

$\emptyset_k(x)$ are real valued expansion functions and called (basis functions).

The expressible functions form a function space that is referred to as the closed span of the expansion set, denoted by:

$$V = \overline{span\{\emptyset_k(x)\}} \tag{2.2.9}$$

- **Scaling Functions [45]**

The set of expansion functions composed of integer translations and binary scaling of the real, square integrable function $\emptyset(x)$; is the set $\{\emptyset_{s,\tau}(x)\}$ where

$$\emptyset_{s,\tau}(x) = 2^{s/2}\emptyset(2^s x - \tau), \forall s, \tau \in \mathbb{Z}\ and\ \emptyset(x) \in L^2(\mathbb{R}) \tag{2.2.10}$$

Where, s : is $\emptyset_{s,\tau}(x)$'s width, how broad or narrow it is along *x*-axis,

τ : is the position of $\emptyset_{s,\tau}(x)$ along the *x*-axis,

$2^{\frac{s}{2}}$: is controls $\emptyset_{s,\tau}(x)$'s height or amplitude.

Because the shape of $\emptyset_{s,\tau}(x)$ changes with *s*, $\emptyset(x)$ is called a ***scaling function***. By Eq. (2.2.9), we will denote the subspaces spanned over τ for any *s* as,

$$V_s = \overline{span_\tau\{\emptyset_{s,\tau}(x)\}} \tag{2.2.11}$$

A V_s sequence of nested closed subspaces functions spanned by $\emptyset_{s,\tau}$ over τ for any *s*; by Eq. (2.2.8), if $f(x) \in V_s$ then, it can be written as

$$f(x) = \sum_\tau \alpha_\tau \emptyset_{s,\tau}(x) \tag{2.2.12}$$

It is noted that increasing *s* increases the size of V_s , which implies that functions with smaller variations will be included in the subspace. This is

because when s is increasing, the set $\{\emptyset_{s,\tau}(x)\}$ becomes narrower and separated by smaller changes in *x*, as will be seen in the following example:

Example 2.2.1 [45]: Consider the unit-height, unit-width scaling function $\emptyset(x)$, which is called Haar scaling function, where;

$$\emptyset(x) = \begin{cases} 1 & 0 \le x < 1 \\ 0 & otherwise \end{cases} \qquad (2.2.13)$$

Figures 2.2.4 (a)-(d) depicts some of the many expansion functions by substituting the given scaling function into Eq. (2.2.10).

Note that the expansion functions for s = 1 in Figures 2.2.4(c) and (d) are half as wide as those for s = 0 in Figures 2.2.4(a) and (b). In addition, it can be defined twice as many V_1 scaling functions as V_0 scaling functions (for example: $\emptyset_{1,0}$ and $\emptyset_{1,1}$ of V_1 versus $\emptyset_{0,0}$ of V_0 for $\in [0,1)$).

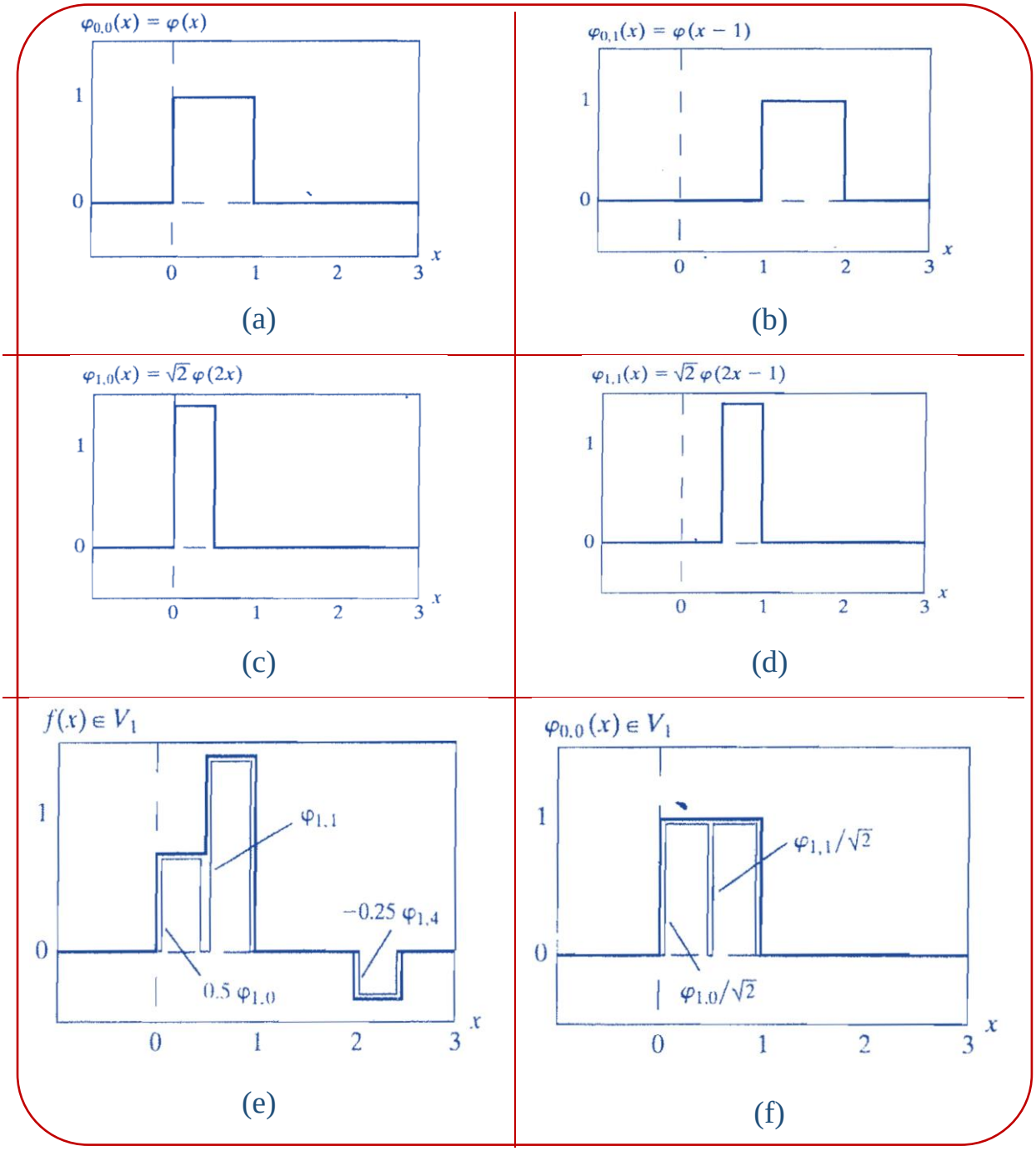
Figure 2.2.4(e) shows a member of subspace V_1 . This function does not belong to V_0 because the V_0 expansion functions in Figures 2.2.4(a) and (b) are too coarse to represent it. Higher resolution functions like those in Figures 2.2.4(c) and (d) are required. They can be used, as shown in (e), to represent the function by the expansion

$$f(x) = \frac{1}{2}\emptyset_{1,0}(x) + \emptyset_{1,1}(x) - \frac{1}{4}\emptyset_{1,4}(x) \qquad (2.2.14)$$

Also, the decomposition of $\emptyset_{0,0}(x)$ as a sum of V_1 expansion functions is included in Figure 2.2.4(f). Similarly, any V_0 expansion function can be decomposed using the following relation:

$$\emptyset_{0,\tau}(x) = \frac{1}{\sqrt{2}}\emptyset_{1,2\tau}(x) + \frac{1}{\sqrt{2}}\emptyset_{1,2\tau+1}(x) \qquad (2.2.15)$$

Thus, if $f(x) \in V_0$ then $f(x) \in V_1$. This means $V_0 \subset V_1$.



Figures 2.2.4 : Some Haar Scaling Functions [45]

➢ **Wavelet Function [45]**

A wavelet function $\psi_{s,\tau}(x)$ can be defined as spans for the difference between any two adjacent scaling subspaces, V_s and V_{s+1}.

The set $\psi_{s,\tau}(x)$ of wavelets can be defined as a basis wavelet from the mother function $\psi(x)$ such as:

$$\psi_{s,\tau}(x) = 2^{\frac{s}{2}}\psi(2^s x - \tau) \tag{2.2.16}$$

Where $s, \tau \in \mathbb{Z}$ and $\psi(x) \in L^2(\mathbb{R})$ that spans the W_s spaces as

$$W_s = \overline{span_\tau\{\psi_{s,\tau}(x)\}} \tag{2.2.17}$$

Where W_s is a sequence of closed subspaces of functions spanned by $\psi_{s,\tau}(x)$ over τ for any s, by Eq. (2.2.8), if $f(x) \in W_s$ then, it can be written as

$$f(x) = \sum_\tau \alpha_\tau \psi_{s,\tau}(x) \tag{2.2.18}$$

Remark 2.2.1 [45]:

The subspaces spanned by the scaling function at low scales are nested within those spanned at higher scales. So, the subspaces containing high-resolution function must also contain all lower resolution functions, that is,

$$V_{-\infty} \subset \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_\infty = L^2(R) \tag{2.2.19}$$

Figure 2.2.5 explains this remark.

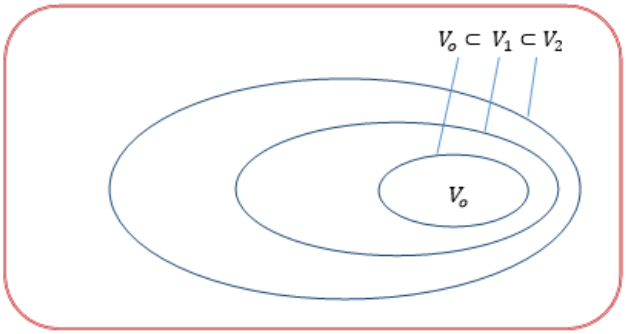


Figure 2.2.5: The Nested Function Spaces Spanned By a Scaling Function [45]

Remark 2.2.2 [45]:

The scaling and wavelet functions subspace are related by

$$V_{s+1} = V_s \oplus W_s \;\; and \;\; L^2(R) = V_{s_0} \oplus W_{s_0} \oplus W_{s_0+1} \oplus \cdots \qquad (2.2.20)$$

Where $\oplus$ denotes the union of spaces (like the union of sets) and $V_s \perp W_s$.

For constructing orthogonal wavelets, the theory of ***multiresolution analysis*** (*MRA*) explained a systematic method to achieve this task. The first basic concept of (*MRA*) was introduced by Mallat [46]. This concept is the material of the following section.

2.3 Multiresolution Analysis (MRA)

The (*MRA*) is related to the representation and analysis of signals (or images) at more than one resolution. It is to approximate a function *f(x)* at various levels of resolution by analyzing a function at different scales [45]. In (*MRA*), two functions are taking into consideration: the scaling function $\phi(x)$ which structures, using Eq.(2.2.10), a number of scaling functions $\phi_{s,\tau}(x)$ by the dilated (scaled) and translated (shifted); and the mother wavelet $\psi(x)$, which structures, using Eq.(2.2.16), a number of orthogonal wavelet basis functions $\psi_{s,\tau}(x)$.

$\|\psi_{s,\tau}\|_{L^2}^2 = \|\psi\|_{L^2}^2$, because if we let $= (2^s x - \tau)$, we obtain $\|\psi_{s,\tau}\|_{L^2}^2 = \int_{\mathbb{R}} \left|2^{s/2}\psi(2^s x - \tau)\right|^2 dx = \int_{\mathbb{R}} 2^s |\psi(2^s x - \tau)|^2 \, dx$

$\int_{\mathbb{R}} |\psi(u)|^2 \, du = \|\psi\|_{L^2}^2$.

Similarly, $\|\phi_{s,\tau}\|_{L^2}^2 = \|\phi\|_{L^2}^2$. In fact, $\|\phi_{s,\tau}\|_{L^2}^2 = \|\psi_{s,\tau}\|_{L^2}^2$, $\forall s, \tau \in \mathbb{Z}$.

To design a multiresolution analysis, we need a set of nested spaces, by selecting the functions $\phi_{s,\tau}, \forall s, \tau \in \mathbb{Z}$; we determine the nested spaces V_s , $\forall s \in \mathbb{Z}$. For fixed s , the set of scaling functions $\phi_{s,\tau}$ are orthonormal.

Now consider V_s to be the vector spaces corresponding to spanning set $\{\phi_{s,\tau}(x)\}$, assuming that the resolution increases with decreasing s, and these vector spaces characterize successive approximation vector spaces, (i.e, each space V_s is contained in the next resolution space V_{s+1}), this is depicted in Figure 2.2.5 and each with resolution 2^s such as in Eq.(2.2.19) [46]. To say that $f(x) \in V$ means that f(x) belongs to the closed span of $\phi_{s,\tau}(x)$ and can be written in the form of Eq.(2.2.12).

Thus, a (*MRA*) with scaling function ϕ consists of a sequence of closed subspaces $\{V_s\,,\ s \in \mathbb{Z}\}$ of $L^2(R)$ which have the following properties [45]:

1. $V_s \subset V_{s+1}$, $\forall s \in \mathbb{Z}$. (Nested)
2. $\overline{\bigcup_{s\in\mathbb{Z}} V_s} = L^2(\mathbb{R})$. (Density)
3. $\bigcap_{s\in\mathbb{Z}} V_s = \{0\}$. (Trivial intersection)
4. The following scale relations exist:

   $f(x) \in V_s \leftrightarrow f(2x) \in V_{s+1}$. (Scaling invariance property)

   $f(x) \in V_0 \leftrightarrow f(x-n) \in V_0$, $\forall n \in \mathbb{Z}$. (Scaling invariance property)
5. There exists a scaling function $\phi \in V_0$ such that its integer translates,

   $\{\phi(x-n)\}$ is an orthonormal basis of V_0, where $\phi_{s,\tau}(x)$ as in Eq.(2.2.10)

This means that the basic rule of multiresolution analysis is that whenever the above properties are satisfied, there exists an orthonormal wavelet basis and scaling basis such that any $f(x) \in L^2(\mathbb{R})$ can be expanded as a linear combination of both the scaling basis function $\phi_{s,\tau}(x)$ and the wavelet basis functions $\psi_{s,\tau}(x)$.

In a (MRA), since $\phi(x) \in V_0 \subset V_1$ and $\{\phi_{s,\tau}(x), \tau \in \mathbb{Z}\}$ is an orthonormal basis of V_1, there exists some set of coefficients $\{a_n, n \in \mathbb{Z}\}$ such that the function $\phi(x) \in V_0$ can be represented as a linear combination of the functions from V_1 and so [45],

$$\phi(x) = \sum_n a_n \phi(2x - n) \tag{2.3.1}$$

Alternatively, if we consider property (4: Scaling invariance property) of (MRA) written as a linear combination of $\phi(x)$ in the scaled form defined in Eq.(2.3.1), the recursion for $\phi(x)$ can be written in terms of a new set of coefficients $\{h_n\}$ as:

$$\phi(x) = \sqrt{2} \sum_n h_n \phi(2x - n) \tag{2.3.2}$$

where,

$$\sum_n a_n = \sqrt{2} \sum_n h_n \tag{2.3.3}$$

For some coefficients $\{h_n, n \in \mathbb{Z}\}$, using the fact that $\{\phi_{s,\tau}(x)\}$ are orthonormal. The coefficients $\{h_n, n \in \mathbb{Z}\}$ can be obtained by computing the inner product:

$$h_n = \sqrt{2} \int_{-\infty}^{\infty} \phi(x)\, \phi(2x - n)\, dx \tag{2.3.4}$$

Where the function $\phi(x)$ is the scaling function.

It is interesting to notice that the scaling function $\phi(x)$ can be recursively generated by scaled (shrunk to half) and shifted versions of itself, as it is described by Eq. (2.3.1). This means that the scaling function $\phi(x)$ has the self similarity property.

For each (*MRA*), it is also possible to define a mother wavelet, $\psi(x)$, which will explain the detail at each level s. Assume W_s to be the orthogonal

complement of V_s in V_{s+1} ; that is W_s is the difference between the function space V_{s+1} spanned by scaling functions $\phi_{s+1,\tau}(x)$; and the function space V_s spanned by $\phi_{s,\tau}(x)$, so that

$$V_{s+1} = V_s \oplus W_s \tag{2.3.5}$$

Where $\oplus$ represents the union of the two spaces. The space W_s is composed of all functions representable in V_{s+1} but not representable in V_s . This can be carried out recursively to get:

$$L^2(\mathbb{R}) = V_\infty = V_s \oplus W_s \oplus W_{s+1} \oplus W_{s+2} \oplus \ldots \tag{2.3.6}$$

Similar to a function space V_s spanned by the scaling functions $\phi_{s,\tau}(x)$, the function space W_s is also spanned by a set of basis function, called the wavelet functions. Then the fundamental result is that $\{\psi(x-\tau),\ \tau \in \mathbb{Z}\}$ forms an orthonormal basis for W_s , and $W_0 \subset V_1$, orthogonal to all functions in V_s , where $\psi_{s,\tau}(x)$ is defined as in Eq. (2.2.16).

The wavelet functions $\psi_{s,\tau}(x)$ can be expanded in the space V_{s+1} as:

$$\psi_{s,\tau}(x) = \sum_n g_n\, \phi_{s+1,n}(x) = \sum_n g_n\, 2^{(s+1)/2} \phi(2^{s+1}x - n) \tag{2.3.7}$$

Where g_n are the expansion coefficients. Usually we let $s = 0$ and drop the subscripts s and τ to indicate that any wavelet function $\psi_s(x) \in V_1$ can be expressed as a linear combination of the basis scaling functions ϕ_{s+1} of the functions from V_1 ;

$$\psi(x) = \sqrt{2} \sum_n g_n\, \phi(2x - n) \tag{2.3.8}$$

This is in the same form for the scaling functions Eq.(2.3.2).

The Eq. (2.3.8) can be expressed as:

$$\psi(x) = \sum_n (-1)^n a_{1-k} \phi(2x - n) \tag{2.3.9}$$

Where,

$$g_n = (-1)^n 2^{-1/2} a_{1-k} \tag{2.3.10}$$

The coefficients $\{g_n\}$ can be obtained by computing the inner product:

$$g_n = \sqrt{2} \int_{-\infty}^{\infty} \psi(x)\, \phi(2x - \tau) dx \tag{2.3.11}$$

These coefficients are called coefficients of highpass filter. Coefficients of highpass filter can be calculated from coefficients of lowpass filter using this Equation [45]:

$$g_n = (-1)^n h_{1-n} \tag{2.3.12}$$

Remark 2.3.1: The scaling function in example (2.2.1) satisfies the requirements of (*MRA*).

Example 2.3.1: The Haar scaling coefficients was defined with the boundary conditions:

$h_\phi(0) = h_\phi(1) = \frac{1}{\sqrt{2}}$. Using Eq.(2.3.12), the coefficients for the wavelet function are:

$g_0 = (-1)^0 h_{1-0} = h_1 = \frac{1}{\sqrt{2}}$ and $g_1 = (-1)^1 h_{1-1} = -h_0 = -\frac{1}{\sqrt{2}}$.

Substituting these values into Eq.(2.3.8), we obtained the wavelet functions as:

$$\psi(x) = \sqrt{2} \sum_n g_n\, \phi(2x - n) = \sqrt{2} \frac{1}{\sqrt{2}} \phi(2x) - \sqrt{2} \frac{1}{\sqrt{2}} \phi(2x - 1)$$

$= \phi(2x) - \phi(2x-1)$.

Which is plotted in Figure 2.3.1(a), thus, the Haar wavelet function is

$$\psi(x) = \begin{cases} 1 & 0 \le x < 0.5 \\ -1 & 0.5 \le x < 1 \\ 0 & otherwise \end{cases}$$

Using Eq. (2.2.16), we can generate the universe of scaled and translated Haar. Figure 2.3.1(a) explains $\psi_{0,0}(x) = 2^0\psi(2^0x - 0) = \psi(x)$, and two such wavelets $\psi_{0,2}(x)$ and $\psi_{1,0}(x)$ are plotted in Figure 2.3.1(b) and Figure 2.3.1(c) respectively, where;

$$\psi_{0,2}(x) = 2^0\psi(2^0x - 2) = \psi(x-2)$$

$$\psi_{1,0}(x) = 2^{\frac{1}{2}}\psi(2^1x) = \sqrt{2}\psi(2x)$$

Finally, Figure 2.3.1(d) shows a function of space V_1 that is not in the subspace V_0 . However, Eq.(2.3.5) indicates that it can be expanded using V_0 and W_0 expansion functions as:

$$f(x) = f_a(x) + f_d(x) \qquad (2.3.13)$$

Where,

$$f_a(x) = \frac{3\sqrt{2}}{4}\phi_{0,0}(x) - \frac{\sqrt{2}}{8}\phi_{0,2}(x)$$

and $$f_d(x) = \frac{-\sqrt{2}}{4}\psi_{0,0}(x) - \frac{\sqrt{2}}{8}\psi_{0,2}(x).$$

Here, $f_a(x)$ is an approximation of $f(x)$ using V_0 scaling functions, while $f_d(x)$ is the difference $f(x) - f_a(x)$ as a sum of W_0 wavelets. The two expansions are shown in Figures 2.3.1(e) and (f) [45].

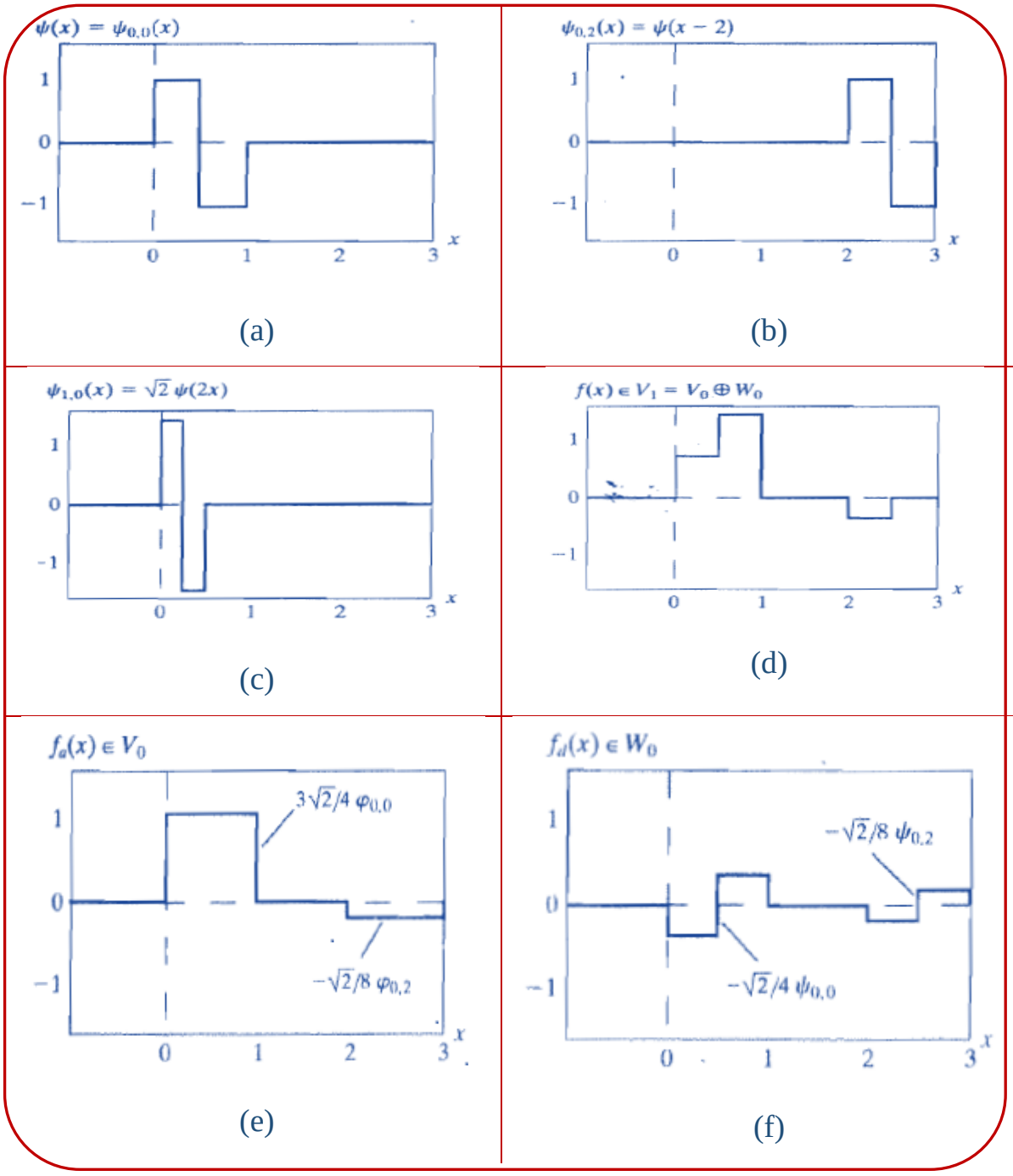


Figure 2.3.1: Haar Wavelet Functions in W_0 and W_1 [45]

The wavelet transform has two types of transforms that are explained in the following section.

2.4 Wavelet Transform Types

2.4.1 The Continuous Wavelet Transform (CWT):

The Continuous wavelet transform (CWT) of a function *f(x)* includes a mother wavelet $\psi(x)$. The mother wavelet can be any continuous function, real or complex, that satisfies the following properties [1]:

1. The total area under the curve of the function is zero, such that :

$$\int_{-\infty}^{\infty} \psi(x)dx = 0 \qquad (2.4.1.1)$$

2. The total area of $|\psi(x)|^2$ is finite , such that

$$\int_{-\infty}^{\infty} |\psi(x)|^2 \, dx \; < \infty \qquad (2.4.1.2)$$

The CWT in 1-D of a square integrable function *f*(x) with respect to the wavelet $\psi(x)$ is a function $W_\psi(s,\tau)$ of two variables s,τ , and defined as:

$$W_\psi(s,\tau) = \int_{-\infty}^{\infty} f(x)\psi_{s,\tau}(x)dx \qquad (2.4.1.3)$$

Where,

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-\tau}{s}\right), s,\tau \in \mathbb{R}; s \neq 0 \qquad (2.4.1.4)$$

$W_\psi(s,\tau)$: is the wavelet coefficient of the function *f(x)*.

s : is the scale parameter.

τ : is the position parameter.

The quantity $\frac{1}{\sqrt{s}}$ is a normalizing factor that guarantees the energy of $\psi(x)$ remains independent of s and τ, such that [1]:

$$\int_{-\infty}^{\infty} |\psi_{s,\tau}(x)|^2 dx = \int_{-\infty}^{\infty} |\psi(x)|^2 \, dx \qquad (2.4.1.5)$$

For any s, $\psi_{s,\tau}(x)$ is a copy of $\psi_{s,0}(x)$ shifted τ along the time axis. Setting $\tau = 0$, τ is said to be ***translation*** parameter, such that :

$$\psi_{s,0}(x) = \frac{1}{\sqrt{s}} \psi\left(\frac{x}{s}\right) \quad for\ any\ s \qquad (2.4.1.6)$$

The parameter s is said to be a ***scaling*** (dilation) parameter.

If $s > 1$ stretch the wavelet, while $0 < s < 1$ shrink the wavelet, as shown in Figure 2.4.1.1 [1].

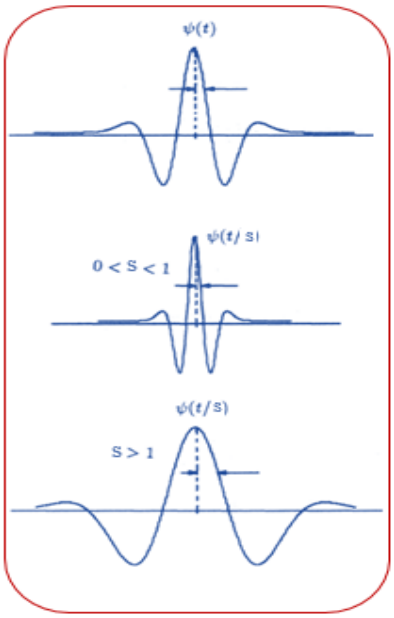


Figure 2.4.1.1: Typical Wavelet Family in Time and Frequency Domains[47]

A function *f(x)* can be restored using the inverse continuous wavelet transformation by the formula:

$$f(x) = \frac{1}{C_\psi} \int_0^\infty \int_{-\infty}^\infty \frac{1}{s^2} W_\psi(s,\tau)\psi_{s,\tau}(x) d\tau\, ds \qquad (2.4.1.7)$$

Where,

$W_\psi(s,\tau)$: is a given wavelet coefficient

$\psi_{s,\tau}(x)$: is Wavelet function.

C_ψ : is defined as

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega \qquad (2.4.1.8)$$

And $\Psi(\omega)$ is the Fourier transform of $\psi(x)$, such that:

$$\Psi(\omega) = \int_{-\infty}^{\infty} \psi(x) e^{-i\omega x} dx \qquad (2.4.1.9)$$

The inverse CWT exists if C_ψ is positive and finite. Since C_ψ is defined by means of Ψ, which itself is defined by means of the wavelet $\psi(x)$, the requirement that C_ψ be positive and finite imposes another restriction, called the ***admissibility condition***, on the choice of wavelet. So the wavelet is called admissible if $C_\psi < \infty$ [1].

In applications such as image compression, the sampled data are discrete in time. Thus, a discrete representation of time and frequency is needed, which is called the **Discrete *Wavelet Transform*** (DWT). Before giving the definition of DWT, we need to explain the concept of ***wavelet series expansion***.

➢ **Wavelet Series Expansion [45]**

For a specific value $s = s_0$, by means of these subspaces, the Eq. (2.3.5) discussed in section (2.3) can be decomposed as

$$L^2(\mathbb{R}) = V_\infty = V_{s_0} \oplus W_{s_0} \oplus W_{s_0+1} \oplus W_{s_0+2} \oplus \dots \qquad (2.4.1.10)$$

This is shown in Figure 2.4.1.2.

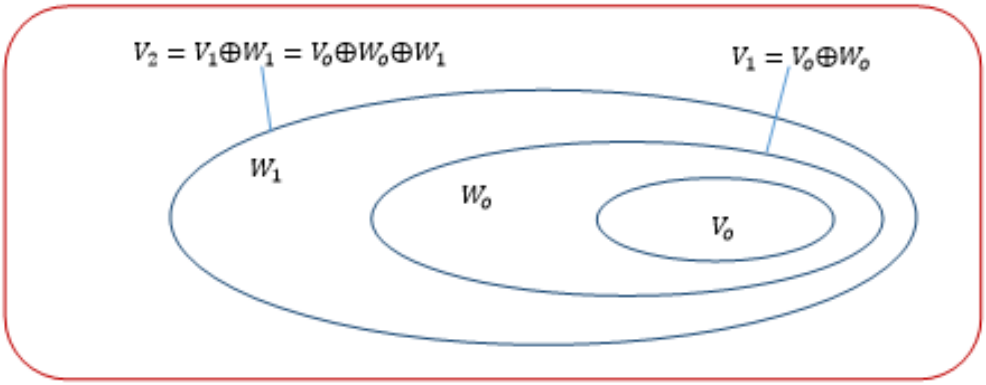


Figure 2.4.1.2: The Relationship Between Scaling and Wavelet Function Space [45]

It indicates that any square-integrable function $f(x) \in L^2(\mathbb{R})$ can be expanded as a linear combination of both the scaling basis functions $\emptyset_{s_0,\tau}(x)$ and the wavelet basis functions $\psi_{s,\tau}(x)$, $s = s_0, s_0 + 1, \dots$. That is, each function $f(x) \in L^2(\mathbb{R})$ can be represented in its wavelet series expansion:

$$f(x) = \sum_{\tau} c_{s_0,\tau} \emptyset_{s_0,\tau}(x) + \sum_{s=s0}^{\infty} \sum_{\tau} d_{s,\tau}\, \psi_{s,\tau}(x) \qquad (2.4.1.11)$$

where $c_{s_0,\tau}$ is called ***approximation coefficient*** or ***scaling coefficients*** defined as:

$$c_{s_0,\tau}(\tau) = \langle f(x), \emptyset_{s_0,\tau}(x) \rangle = \int f(x)\, \emptyset_{s_0,\tau}(x)\, dx \qquad (2.4.1.12)$$

and $d_{s,\tau}$ is called ***detail coefficient*** or ***wavelet coefficients*** defined as:

$$d_{s,\tau}(\tau) = \langle f(x), \psi_{s,\tau}(x) \rangle = \int f(x)\, \psi_{s,\tau}(x)\, dx \qquad (2.4.1.13)$$

The first term contained in the wavelet expansion of the function $f(x)$ Eq. (2.4.1.11) represents the ***approximation*** of the function at scale level s_0 by the linear combination of the scaling functions $\emptyset_{s_0,\tau}(x)$, and the summation with index s in the second term in the expansion is for the ***details*** of different levels contained in the function $f(x)$ approximated by the linear combination of the wavelet functions of progressively higher scales $s_0 + 1, s_0 + 2, \ldots$.

Example 2.4.1.1: Consider a continuous function $f(x)$, which is defined over the period $0 \le x \le 1$:

$$f(x) = \begin{cases} x^2 & 0 \le x \le 1 \\ 0 & otherwise \end{cases}.$$

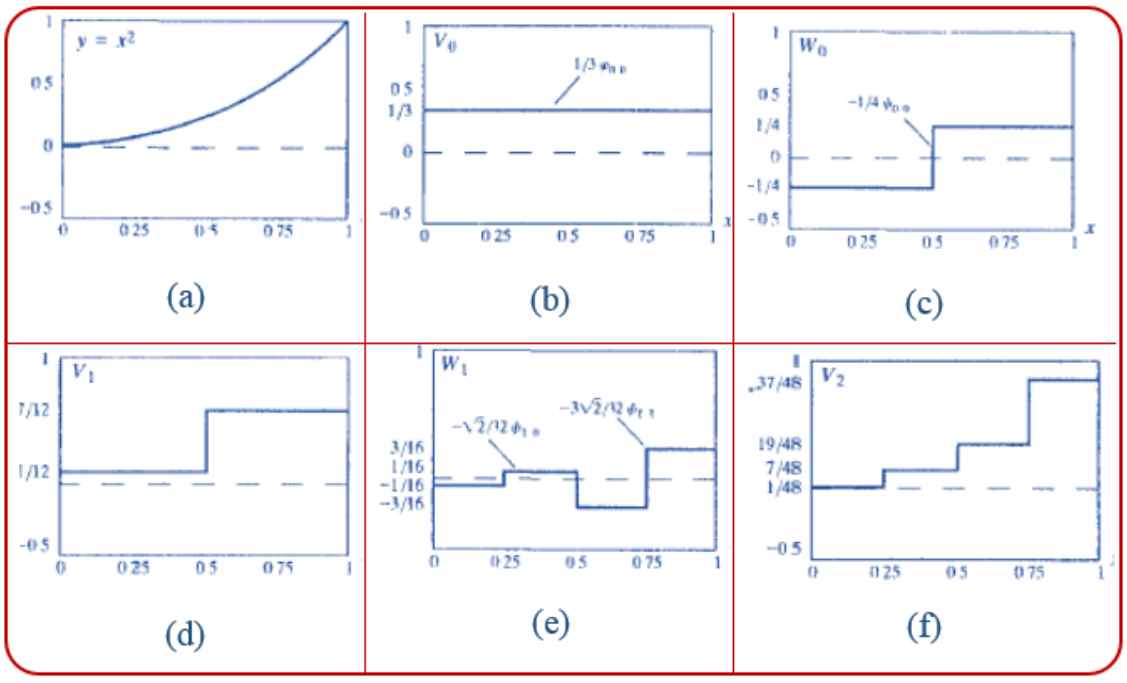


Figure 2.4.1.3: Wavelet Series Expansion of $f(x)=x^2$ Using Haar Wavelets [45]

A shown in Figure 2.4.1.3. Using Harr wavelet and a starting scale $s_0 = 0$, each individual space $V_0, W_0, W_1, \ldots$ is spanned by different number of basis functions. Eqs. (2.4.1.12) and (2.4.1.13) can be used to compute the expansion coefficients:

Since $c_{s_0}(\tau) = \int f(x), \emptyset_{s_0,\tau}(x)\, dx$, then;

$$c_0(0) = \int_0^1 x^2\, \emptyset_{0,0}(x) dx = \int_0^1 x^2\, dx = \frac{1}{3}$$

$$d_0(0) = \int_0^1 x^2\, \psi_{0,0}(x) dx = \int_0^{\frac{1}{2}} x^2\, dx \; - \int_{\frac{1}{2}}^1 x^2\, dx \; = -\frac{1}{4}$$

$$d_1(0) = \int_0^1 x^2\, \psi_{1,0}(x) dx = \int_0^{\frac{1}{4}} x^2\, \sqrt{2}\, dx \; - \int_{\frac{1}{4}}^{\frac{1}{2}} x^2\, \sqrt{2}\, dx \; = -\frac{\sqrt{2}}{32}$$

$$d_1(1) = \int_0^1 x^2\, \psi_{1,1}(x) dx = \int_0^{\frac{3}{4}} x^2\, \sqrt{2}\, dx \; - \int_{\frac{3}{4}}^1 x^2\, \sqrt{2}\, dx \; = -\frac{3\sqrt{2}}{32}$$

Substituting these values into Eq. (2.4.1.11), we get the wavelet series expansion

$$y = \underbrace{\underbrace{\frac{1}{3}\emptyset_{0,0}(x)}_{V_0} + \underbrace{\left[-\frac{1}{4}\psi_{0,0}(x)\right]}_{W_0}}_{V_1 = V_0 \oplus W_0} + \underbrace{\left[-\frac{\sqrt{2}}{32}\psi_{1,0}(x) - -\frac{3\sqrt{2}}{32}\psi_{1,1}(x)\right]}_{W_1} + \cdots}_{V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1}$$

The first term in this expansion uses $c_0(0)$ to generate a subspace V_0 approximation of the function being expanded. This approximation is shown in Figure 2.4.1.3(b) and is the average value of the original function. The second term uses $d_0(0)$ to refine the approximation by adding a level of detail from subspace W_0. The added detail and resulting V_1 approximation are shown in Figures 2.4.1.3(c) and (d), respectively. Another level of detail is added by the subspace W_1 cofficients $d_1(0)$ and $d_1(1)$. This additional detail is shown in Figure 2.4.1.3(e) and the resulting V_2 approximation is depicted in Figure 2.4.1.3(f).

2.4.2 The Discrete Wavelet Transform (DWT):

The DWT was proposed by Mallat (1989) [46], is an efficient algorithm for calculating the coefficients of the wavelet transform of a discrete series. It is like the Fourier series expansion, the wavelet series expansion of the previous section maps a function of a continuous variable into a sequence of coefficients. If the function being expanded is discrete, the resulting coefficients are called the DWT [45].

The DWT in one – dimension (1-D) is given by:

$$W_{\emptyset}(s_0,\tau) = \frac{1}{\sqrt{M}}\sum_{x} f(x)\emptyset_{s_0,\tau}(x) \qquad (2.4.2.1)$$

called the ***approximation*** or scaling coefficients, and

$$W_{\psi}(s,\tau) = \frac{1}{\sqrt{M}}\sum_{x} f(x)\psi_{s,\tau}(x) \ , For\ s \geq s_0 \qquad (2.4.2.2)$$

called the ***detail*** or wavelet coefficients.

The inverse of DWT in (1-D) is given by:

$$f(x) = \frac{1}{\sqrt{M}} \sum_{\tau} W_{\emptyset}(s_0, \tau) \emptyset_{s_0,\tau}(x) + \frac{1}{\sqrt{M}} \sum_{s=s_0}^{\infty} \sum_{\tau} W_{\psi}(s, \tau) \psi_{s,\tau}(x) \qquad (2.4.2.3)$$

Here $f(x)$, $\emptyset_{s_0,\tau}(x)$, and $\psi_{s,\tau}(x)$ are functions of discrete variable *x*, for example $f(x) = f(x_0 + k\Delta x)$ and $k = 0,1,2,\dots,M-1$. The factor $\frac{1}{\sqrt{M}}$ is normalizing factor.

Normally, we let $s_0 = 0$ and select *M* to be a power of 2 (i.e. $M = 2^s$) so that the summations are performed over $k = 0,1,2,\dots,M-1; s = 0,1,2,\dots,s-1$ and $\tau = 0,1,2,\dots,2^s - 1$ [45].

The transform itself is composed of *M* coefficients, the minimum scale is *0* and the maximum scale is (s – 1).

The $W_{\emptyset}(s_0, \tau)$ and $W_{\psi}(s, \tau)$ of Eqs.(2.4.2.1) and (2.4.2.2) correspond to the $c_{s_0}(\tau)$'s and $d_s(\tau)$'s of the ***wavelet series expansion*** that explained in the previous section, that is Eqs. (2.4.1.12) and (2.4.1.13).

Example 2.4.2.1: To compute the 1-D DWT coefficients, consider the discrete function of four points: *f*(0) = 1, *f*(1) = 4, *f*(2) = -3 and *f*(3) = 0. *M* = 4, *s* = 2 and $s_0 = 0$. The summations are performed over *x* = 0,1,2,3 ; *s* = 0,1 and τ = 0 for *s* = 0 or τ =0,1 for *s* = 1. We will use the Haar scaling and wavelet functions and assume that the four sampling of $f(x)$are distributed over the support of the basis functions, which is 1. Substituting the four samples into Eqs. (2.4.2.1) and (2.4.2.2), we find :

$$W_{\emptyset}(0,0) = \frac{1}{2}\sum_{x=0}^{3} f(x)\emptyset_{0,0}(x)$$

$$= \tfrac{1}{2}[1.1 + 4.1 - 3.1 + 0.1] = 1 \text{ (since } \emptyset_{0,0}(x) = 1 \text{ for } x = 0,1,2,3)$$

$$W_{\psi}(s,\tau) = \frac{1}{\sqrt{M}}\sum_{x} f(x)\psi_{s,\tau}(x)$$

$$W_{\psi}(0,0) = \frac{1}{2}[1.1 + 4.1 - 3.(-1) + 0.(-1)] = 4$$

$$W_{\psi}(1,0) = \frac{1}{2}\left[1.\sqrt{2} + 4.(-\sqrt{2}) - 3.0 + 0.0\right] = -1.5\sqrt{2}$$

$$W_{\psi}(1,1) = \frac{1}{2}\left[1.0 + 4.0 - 3.\sqrt{2} + 0.(-\sqrt{2})\right] = -1.5\sqrt{2}$$

Thus, the DWT of the given four sample function relative to the Haar wavelet and scaling function is $[1,4, -1.5\sqrt{2}, -1.5\sqrt{2}]$, where the transform coefficients have been arranged in the order in which they were calculated.

To reconstruct the original function from its transform, using the Eq.(2.4.2.3):

$$f(x) = \frac{1}{\sqrt{M}}\sum_{\tau} W_{\emptyset}(s_0,\tau)\emptyset_{s_0,\tau}(x) + \frac{1}{\sqrt{M}}\sum_{s=s_0}^{\infty}\sum_{\tau} W_{\psi}(s,\tau)\psi_{s,\tau}(x)$$

Then;

$$f(x) = \frac{1}{2}\left[W_{\emptyset}(0,0)\emptyset_{0,0}(x) + W_{\psi}(0,0)\psi_{0,0}(x) + W_{\psi}(1,0)\psi_{1,0}(x) + W_{\psi}(1,1)\psi_{1,1}(x)\right]$$

For x = 0,1,2,3. If x = 0,

$$f(0) = \tfrac{1}{2}\left[1.1 + 4.1 - \left(1.5\sqrt{2}\right).\left(\sqrt{2}\right) - \left(1.5\sqrt{2}\right).0\right] = 1\ .$$

The one dimensional transforms (1-D) are extended easily to two dimensional transform (2-D) by using a 2-D scaling function $\phi(x,y)$, and three 2-D wavelets, $\psi^H(x,y), \psi^V(x,y), and\ \psi^D(x,y)$, so to reconstruct the signal $f(x,y)$ of size $M \times N$ in form the 2-D discrete wavelet coefficient$\left(W_f(s,m,n)\right)$, we found 2-D DWT pair becomes as [45]:

$$W_{\emptyset}(s_0,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \emptyset_{s_0,m,n}(x,y) \qquad (2.4.2.4)$$

$$W_{\psi}^{i}(s,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \psi_{s,m,n}^{i}(x,y) \qquad (2.4.2.5)$$

where,

$$\emptyset_{s_0,m,n}(x,y) = 2^{\frac{s_0}{2}} \phi(2^{s_0}x - m, 2^{s_0}y - n) \qquad (2.4.2.6)$$

$$\psi_{s,m,n}^{i}(x,y) = 2^{\frac{s}{2}} \psi^{i}(2^{s}x - m, 2^{s}y - n) \qquad (2.4.2.7)$$

For $s > s_0$ and $i = \{H,V,D\}$.

As in 1-D case, s_0 is an arbitrary starting scale and the $W_{\emptyset}(s_0,m,n)$ coefficients define an approximation of $f(x,y)$ at scale s_0. The $W_{\psi}^{i}(s,m,n)$ coefficients add horizontal, vertical, and diagonal details for scales; $s > s_0$.

Usually, let $s_0 = 0$ and $N = M = 2^s$ so that $s = 0,1,2,\dots,s-1$ and $M = N = 0,1,2,\dots,2^s - 1$. The inverse discrete wavelet ransform IDWT for 2-D is given by [45] :

$$f(x,y) = \frac{1}{\sqrt{MN}} \sum_{m} \sum_{n} W_{\phi}(s_0,m,n) \phi_{s_0,m,n}(x,y)$$

$$+ \frac{1}{\sqrt{MN}} \sum_{i-H,V,D} \sum_{s=s_0}^{\infty} \sum_{m} \sum_{n} W_{\psi}^{i}(s,m,n) \psi_{s,m,n}^{i}(x,y) \qquad (2.4.2.8)$$

Definition 2.4.2.1 [45] :

A two dimensional scaling function, $\phi(x,y)$, and three of two dimensional wavelet, $\psi^H(x,y), \psi^V(x,y), \psi^D(x,y)$, each is isolated by using the separable property to simplify the complexity computations. Where a two dimensional scaling function products of one dimensional scaling function as :

$$\phi(x,y) = \phi(x)\,\phi(y) \qquad (2.4.2.1.1)$$

and three with two dimensional wavelet, $\psi^H(x,y), \psi^V(x,y), \psi^D(x,y)$, each is the products of one dimensional wavelet function with respect to directionally sensitive wavelet as :

$$\psi^H(x,y) = \psi(x)\,\phi(y) \qquad (2.4.2.1.2)$$

$$\psi^V(x,y) = \phi(x)\,\psi(y) \qquad (2.4.2.1.3)$$

$$\psi^D(x,y) = \psi(x)\,\psi(y) \qquad (2.4.2.1.4)$$

These wavelet measure practical variations intensity or gray level variations for images along different directions as :

ψ^H : measures variations along columns (horizontal edges).

ψ^V : measures variations along rows (vertical edges).

ψ^D : measures variations along diagonals.

Remark 2.4.2.1 :

The component that defined in Eq.(2.4.2.4) is called the ***approximation*** or the low pass component $[W_\phi]$ and characterizes the image's low frequency information; the component that defined in Eq.(2.4.2.5) is called the ***detail*** or high pass component $[W_\psi^H, W_\psi^V, W_\psi^D]$ and contains its high frequency information with various orientation.

2.5 How the Wavelets Transform Works

In order to explain how the wavelets transform works, we take the Haar wavelet transform, which is one of the simplest and basic transformations from the space domain to a local frequency domain. A Haar transform decomposes each signal into two components, one is called ***average*** (approximation) and the other is known as ***difference*** (detail) [1]. The following steps illustrate how the Haar transform can be used to calculate of a matrix of $m \times n$ size.

Step 1: Find the average of each pair of elements.

Step 2: Find the difference between each average and the elements it was calculated from.

Step 3: Fill the first half of the matrix with averages.

Step 4: Fill the second half of the matrix with differences.

Step 5: Repeat the process on the first half of the matrix.

Step 6: If the dimension is odd number, we can add row(coulum) of zero elements.

In order to give an idea of its implementation, the procedure of its application may be explained with the help of a simple example as shown below.

Example 2.5.1: Consider the 8x8 matrix ,

$$M = \begin{bmatrix} 64 & 2 & 3 & 61 & 60 & 6 & 7 & 57 \\ 9 & 55 & 54 & 12 & 13 & 51 & 50 & 16 \\ 17 & 47 & 46 & 20 & 21 & 43 & 42 & 24 \\ 40 & 26 & 27 & 37 & 36 & 30 & 31 & 33 \\ 32 & 34 & 35 & 29 & 28 & 38 & 39 & 25 \\ 41 & 23 & 22 & 44 & 45 & 19 & 18 & 48 \\ 49 & 15 & 14 & 52 & 53 & 11 & 10 & 56 \\ 8 & 58 & 59 & 5 & 4 & 62 & 63 & 1 \end{bmatrix}$$

We start with an arbitrary vector representing one row of an 8×8 matrix.

Step 1: Average

$$y = \begin{bmatrix} 64 & 2 & 3 & 61 & 60 & 6 & 7 & 57 \end{bmatrix}$$

$$\frac{64+2}{2} \quad \frac{3+61}{2} \quad \frac{60+6}{2} \quad \frac{7+57}{2} \Rightarrow \quad B = \begin{bmatrix} 33 & 32 & 33 & 32 \end{bmatrix}$$

Where B is called ***approximation coefficients***, and the results become the first four entries of our modified string y_1 .

Differencing

$$64-33 \quad 3-32 \quad 60-33 \quad 7-32 \Rightarrow \quad C = \begin{bmatrix} 31 & -29 & 27 & -25 \end{bmatrix}$$

Where C is called ***detail coefficients***, and the results become the last four entries of y_1 .

$$y_1 = \begin{bmatrix} B & C \end{bmatrix}$$

$$y_1 = \begin{bmatrix} 33 & 32 & 33 & 32 & 31 & -29 & 27 & -25 \end{bmatrix}$$

Step 2: Average

$$y_1 = [\mathbf{33} \quad \mathbf{32} \quad \mathbf{33} \quad \mathbf{32} \quad 31 \quad -29 \quad 27 \quad -25]$$

$$\frac{33+32}{2} \quad \frac{33+32}{2} \quad \Rightarrow \quad \mathrm{B} = [32.5 \quad 32.5]$$

Differencing

$$33-32.5 \quad 33-32.5 \quad \Rightarrow \quad \mathrm{C} = [0.5 \quad 0.5]$$

$$y_2 = [\mathbf{32.5} \quad \mathbf{32.5} \quad 0.5 \quad 0.5 \quad 31 \quad -29 \quad 27 \quad -25]$$

Step 3: Average

$$y_2 = [\mathbf{32.5} \quad \mathbf{32.5} \quad 0.5 \quad 0.5 \quad 31 \quad -29 \quad 27 \quad -25]$$

$$\frac{32.5+32.5}{2} \quad \Rightarrow \quad \mathrm{B} = [32.5]$$

Differencing

$$32.5-32.5 \quad \Rightarrow \quad \mathrm{C} = [0]$$

$$y_3 = [\mathbf{32.5} \quad \mathbf{0} \quad 0.5 \quad 0.5 \quad 31 \quad -29 \quad 27 \quad -25]$$

The Haar wavelet does this transformation to each row of the image matrix, and then again to every column in the matrix. The resulting matrix is known as the Haar wavelet transform of the original matrix. It is important to note at this point that this process is entirely reversible. It is this fact that makes it possible to retrieve the original matrix from the Haar wavelet transform of the matrix.

Apply average and differencing to the entire matrix *M*

First: the rows of matrix M is:

$$M=\begin{bmatrix}64&2&3&61&60&6&7&57\\9&55&54&12&13&51&50&16\\17&47&46&20&21&43&42&24\\40&26&27&37&36&30&31&33\\32&34&35&29&28&38&39&25\\41&23&22&44&45&19&18&48\\49&15&14&52&53&11&10&56\\8&58&59&5&4&6&63&1\end{bmatrix}\Rightarrow\begin{bmatrix}32.5&0&.5&.5&31&-29&27&-25\\32.5&0&-.5&-.5&-23&21&-19&17\\32.5&0&-.5&-.5&-15&13&-11&9\\32.5&0&.5&.5&7&-5&3&-1\\32.5&0&.5&.5&-1&3&-5&7\\32.5&0&-.5&-.5&9&-11&13&-15\\32.5&0&-.5&-.5&17&-19&21&-23\\32.5&7&.5&-13.5&-25&27&-1&31\end{bmatrix}$$

Second: the columns of matrix M is:

$$\begin{bmatrix}32.5&0&.5&.5&31&-29&27&-25\\32.5&0&-.5&-.5&-23&21&-19&17\\32.5&0&-.5&-.5&-15&13&-11&9\\32.5&0&.5&.5&7&-5&3&-1\\32.5&0&.5&.5&-1&3&-5&7\\32.5&0&-.5&-.5&9&-11&13&-15\\32.5&0&-.5&-.5&17&-19&21&-23\\32.5&7&.5&-13.5&-25&27&-1&31\end{bmatrix}\Rightarrow\begin{bmatrix}32.5&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&4&-4&4&-4\\0&0&0&0&4&-4&4&-4\\0&0&.5&.5&27&-25&23&-21\\0&0&-.5&-.5&-11&9&-7&5\\0&0&.5&.5&-5&7&-9&11\\0&0&-.5&-.5&21&-23&25&-27\end{bmatrix}=\text{N}$$

A matrix N can be represented as a more concise manner, with one overall average in the upper left-hand corner of the matrix is called ***approximation coefficients***. The remaining components are all ***detail coefficients*** that now represent the amount of detail in that area of the matrix. Because we know this, we can eliminate some information from the given matrix and still be capable of attaining a fairly good approximation of the original matrix. Doing this it can choose some number δ and set equal to zero all elements with a magnitude less than δ.

Choosing $\delta = 5$, then eighteen of the detail coefficients (bold) are eliminated.

$$\begin{bmatrix} 32.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & \mathbf{0} & \mathbf{0} & 27 & -25 & 23 & -21 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & -11 & 9 & -7 & \mathbf{0} \\ 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 7 & -9 & 11 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & 21 & -23 & 25 & -27 \end{bmatrix}$$

Remark 2.5.1:

The averaging and differencing method that we just discussed is very effective. Using linear algebra, we can use three matrices (A_1, A_2, A_3) that perform each of the three steps of the averaging and differencing process. In our previous Example (2.5.1), the transformation of y to y_1 can describe as

$$y_1 = yA_1$$

$$= [64 \quad 2 \quad 3 \quad 61 \quad 60 \quad 6 \quad 7 \quad 57] \begin{bmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 \end{bmatrix}$$

$$= [33 \quad 32 \quad 33 \quad 32 \quad 31 \quad -29 \quad 27 \quad -25]$$

It can next be shown that the transformation from y_1 to y_2 can be written as $y_2 = y_1 A_2$

$$= [\,33 \;\; 32 \;\; 33 \;\; 32 \;\; 31 \;\; -29 \;\; 27 \;\; -25] \begin{bmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [\,32.5 \;\; 32.5 \;\; 0.5 \;\; 0.5 \;\; 31 \;\; -29 \;\; 27 \;\; -25]$$

and lastly we can show that $y_3 = y_2 A_3$

$$= [\,32.5 \;\; 32.5 \;\; 0.5 \;\; 0.5 \;\; 31 - 29 \;\; 27 - 25] \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= [\,32.5 \;\; 0 \;\; 0.5 \;\; 0.5 \;\; 31 \;\; -29 \;\; 27 \;\; -25]$$

This whole transformation can be done in one step by multiplying these three matrices together to obtain a single transform matrix $= A_1 A_2 A_3$. We can now multiply the original string by just one transform matrix to go directly from the original string to the final results of step 3.where W is,

$$W = [64 \quad 2 \quad 3 \quad 61 \quad 60 \quad 6 \quad 7 \quad 57] \begin{bmatrix} 1/8 & 1/8 & 1/4 & 0 & 1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & 1/4 & 0 & -1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & 1/2 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & -1/2 & 0 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & 1/2 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & -1/2 & 0 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & 1/2 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & -1/2 \end{bmatrix}$$

$$= [\,32.5 \quad 0 \quad 0.5 \quad 0.5 \quad 31 \quad -29 \quad 27 \quad -25]$$

The following equations simplify this process of matrix multiplication of the averaging and differencing.

$$T = ((MW)^T W)^T$$

$$T = (W^T M^T W)^T$$

$$T = W^T (M^T)^T (W^T)^T$$

$$T = W^T M W$$

It is also important to note that since every column of the individual matrices that comprise W is orthogonal to every other column, the matrices are invertible. Thus,

$$W^{-1} = A_3^{-1} A_2^{-1} A_1^{-1}, \text{ then}$$

$$N = W^T M W$$

$$M = (W^T)^{-1} N W^{-1}$$

Where, M is original matrix

W is transforming matrix

N is compressed matrix

2.6 Singular Value Decomposition (SVD)

Singular value decomposition (SVD), in a least square sense, is the optimal matrix decomposition that it packages the maximum signal energy into as few coefficients as possible. In numerical analysis, SVD is a numerical technique used to diagonalize matrices [48]. It is well-known method in linear algebra [2] to diagonalise a rectangular $m \times n$ matrix A by factorizing it into three matrices U, S, and V, such that,

$$A = USV^T \tag{2.6.1}$$

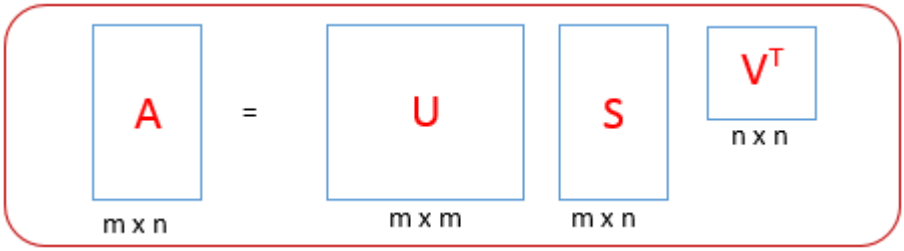


Figure 2.6.1: Illustration of Factoring A to USV^T

Where S is a diagonal $m \times n$ matrix (the same dimensions as A) with elements s_i along the diagonal and zeros everywhere else. U and V are orthonormal matrices with sizes $m \times m$ and $n \times n$, respectively. The matrix U is called the left singular matrix, V is called the right singular matrix, and the diagonal matrix S is the singular values matrix. The singular vectors form orthonormal bases and lead to the following relationship:

$$Av_i = s_i u_i \tag{2.6.2}$$

The SVD is an approximation method which effectively reduces any matrix into a smaller invertible and square matrix. Thus, one special feature of SVD is that it can be performed on any real $m \times n$ matrix. Eq. (2.6.2) can be expressed as [49] :

$$A = \sum_{i=1}^{p} u_i s_i v_i^T \tag{2.6.3}$$

Where u_i and v_i are the i^{th} column vectors of *U* and *V* respectively, s_i are the singular values, and $p = min\{m, n\}$. If the singular values are ordered so that $s_1 \geq s_2 \ldots \geq s_p$, and if the matrix *A* has a rank $r < p$, then the last $p - r$ singular values are equal to zero, and the matrix *A* can be approximated by a matrix A^* with rank *r* (i.e. the SVD becomes A^*) as the following equation:

$$A^* = \sum_{i=1}^{r} u_i s_i v_i^T \tag{2.6.4}$$

Hence, the approximation error matrix E_r is dependent on the performance accuracy of the quantization and/or truncation by parameter *r*, which can be described as $E_r = A - A^*$. The 2-norm of a matrix may be calculated from the singular values. The 2-norm of approximation error is calculated by

$$E_r^2 = \|A - A^*\|_2 = \left\| \sum_{i=1}^{p} u_i s_i v_i^T - \sum_{i=1}^{r} u_i s_i v_i^T \right\|_2 = \left\| \sum_{i=r+1}^{p} u_i s_i v_i^T \right\|_2$$

$$= \sum_{i=r+1}^{p} (s_i)^2 \tag{2.6.5}$$

As the singular values are in descending order, it can be seen that the error decreases towards zero in the 2- norm sense. The property of SVD to provide the closest rank *r* approximation for a matrix *A* as shown in Eq. (2.6.4) can be used in image processing for compression and noise reduction. By setting the small singular values to zero, matrix approximations whose rank equals the number of remaining singular values can be obtained [49].

CHAPTER THREE

CONCEPTS OF IMAGE COMPRESSION

3.1 Introduction

Images are vastly used in computer applications and storing these images in less memory leads to reduction directly in storage cost and faster data transmission.

In this chapter we shed light on the some definitions and concepts that are concerning to image compression in order to decrease requirements of data storage and hence communication costs. Thus the development of efficient compression techniques will continue to be a design challenge for future communication systems and advanced multimedia applications.

Data compression may be the fundamental expression of Information Theory. Information Theory is a branch of mathematics that had its beginning in the late 1940s with the work of *Claude Shannon*. Data compression enters into the field of Information Theory because of its concern with ways of storing and redundancy. Redundant information takes extra bit to encode, and if it can get rid of that extra information, the size will have reduced [50].

3.2 Some Definitions

- **Digital Image [45]:**

An image is defined as a two-dimensional function, $f(x,y)$, contains of small dots called "pixels". Each one of pixel can be either one bit, points to a black or a white dot; or several bits, points to one of several colors or shades of gray. Each pixel is identified with unique positional *(x,y)*. The elements of array $f(x,y)$, of image having *M* rows and *N* columns and (*M* x *N*) sets the resolution of the image. In other meaning, image is stored as a two-dimensional signal and represented by function $f(x,y)$, where x and y are spatial coordinates of a pixel and the value of a function at any pair of

coordinate is called the gray level of the image or "the intensity" of a pixel at that point.

The representation of the function $f(x,y)$ is:

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (3.2.1)$$

Digital image has a finite number of elements; each of these elements has a particular value and location. There are different types of images and in this thesis we depend on a ***gray-scale*** image which contains the brightness information. This image contains 8 bits per pixel; so, there are 256 different possible gray-level or intensity values from 0 to 255.

- **Image Compression [50]:**

Image compression is a branch of the data compression that searches for reduce the number of bits used to store or transmit information in images. Data is represented as a combination of ***information*** and ***redundancy***. Information is the portion of data that must be preserved always in its original form in order to correctly understand the meaning or purpose of the image. Redundancy is that portion of data that can be removed when it is not needed. A technique to reduce the redundancy of data is defined as ***data compression***.

From a mathematical viewpoint, the compression process is a transformation of a 2-D pixel array into a statistically uncorrelated data set .The transformation is applied prior to storage or transmission of the image. Therefore, the basic objective of image compression is to find an image representation in which pixels are less correlated.

There are two types of image compression methods. **The first** type that preserve the data, is called "***lossless***" methods, since no data are lost and the original image can be exactly recreated from the compressed data. **The second** type of compression methods that allow some loss of data, are called "***lossy***" methods, since they allow a loss in the actual image data, so the original uncompressed image cannot be exactly created from the compressed image.

- **Edge Detecting [45]:**

Edges are formed from pixels with derivative values that override a preset threshold. Thus, the idea of an edge is a "local" concept that is based on an amount of intensity-level ***discontinuity*** at a point. That is, the basic idea behind edge detection is to detect places in an image where the intensity ***changes rapidly***, using one of two generic criteria:

1. Find places where the 1^{st} derivative of the intensity is greater in magnitude than a specified threshold.
2. Find places where the 2^{ed} derivative of the intensity has a zero crossing.

- **Distortion Measures [1]:**

A distortion measure is a criterion that appropriates a "quality number" to an image. It caused in the recovered image by the image compression process and can be measured in several ways. There are different kinds of distortion measures with different properties and the most important examples for mathematical distortion measures are the ***Mean Squared Error*** (*MSE*), the ***Peak Signal to Noise Ratio*** (*PSNR*). They work by comparing the squared error (power) between the original and the recovered digital image:

$$MSE = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[f(x,y) - f^*(x,y)]^2 \quad (3.2.2)$$

$$PSNR = 10log_{10}\left(\frac{(L-1)^2}{MSE}\right) \quad (3.2.3)$$

Where,

L : is the number of gray levels (e.g., for 8 bits L = 256).

$f(x,y)$: the original image, $f^*(x,y)$: the decompressed image.

x , y : row and column of the image f.

- **Structural Similarity [51]:**

The structural similarity (*SSIM*) index is a method used for measuring the similarity between two images. It is a full reference metric; in other meaning, SSIM is contagious to improve on classical methods such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proven to be inconsistent with human visual perception. It has been defined as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.2.4)$$

Where,

$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$: the average of x,

$\mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$: the average of y,

$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2$: the variance of x,

$\sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2$: the variance of y,

$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(x_i - \mu_y)$: the covariance of x and y,

$\left.\begin{matrix} C_1 = (K_1 L)^2 \\ C_2 = (K_2 L)^2 \end{matrix}\right\}$: two variables to stabilize the division with weak denominator,

L : is the dynamic range of the pixel values (256 for 8-bit images)

K1 = 0.01 and K2 = 0.03 by default.

The overall image quality is measured by the Mean SSIM (*MSSIM*) index which is given by

$$MSSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j) \qquad (3.2.5)$$

➢ Compression Performance

Various measures are commonly used to express the performance of a compression method:

1. The Compression Ratio (CR) [1]: It is defined as

$$Compression\ ratio = \frac{size\ of\ the\ output\ stream}{size\ of\ the\ input\ stream} \qquad (3.2.6)$$

The compression ratio can also be called ***bit per pixel*** "bpp". It is the number of bits necessary to describe one pixel of the image. It is defined as

$$bpp = \frac{Number\ of\ bits}{Number\ of\ pixels} = \frac{8(Number\ of\ bytes)}{M \times N} \qquad (3.2.7)$$

2. The Compression Factor (CF) [1]: It is the inverse of the compression ratio defined as:

$$Compression\ factor = \frac{size\ of\ input\ stream}{size\ of\ output\ stream} \qquad (3.2.8)$$

In this state, values greater than (1) indicate compression and values less than (1) imply expansion. This measure seems natural to many people, since the bigger the factor, the better the compression.

3.3 Advantages and Disadvantages of Data Compression

Compression of data offer many advantages:

- Reduce disk space required.
- Unused disk space can be used for shadowing to increase reliability.
- Compressed data can be transferred faster to and from disk.
- Faster transfer rates across the network.

Although data compression offers many advantages, it has some disadvantage too, depending on sensitivity of the data and the application area. Some of these disadvantages are:

- Data compression generally reduces the reliability of the compression files.
- the extra overhead incurred by encoding and decoding processes is one of the most serious drawbacks of data compression
- Transmission of very sensitive compressed data through a noisy communication channel is risky because the burst errors introduced by the noisy can destroy the transmitted data.
- Disruption of some data properties of a compressed data, it will result when the compressed data largely different from the original data.

3.4 Compression System Model [52]

The compression system model consists of two parts: "***the compressor***", and "the ***de-compressor***", as illustrated in Figure 3.4.1.

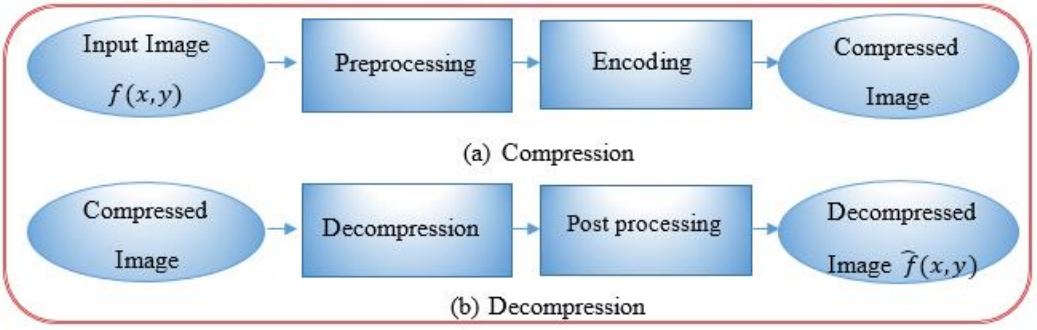


Figure 3.4.1: Compression System Model [52]

The compressor classifies into a pre-processing stage and encoding stage:

- The first stage in pre-processing is data reduction: here, the image data can be reduced by gray-level and/or spatial quantization, or they can undergo any desired image improvement (for example, noise removal) process.
- The second step in pre-processing is the mapping process, which maps the original image data into another mathematical space where it is easier to compress the data.
- Next, as part of the encoding process, is the quantization stage, which takes the possibly continuous data from the mapping stage and puts it in discrete form.
- The final stage of encoding includes coding the resulting data, which maps the discrete data from the quantizer stage onto a code in an optimal mode. All the compressor stages illustrated in Figure 3.4.2.

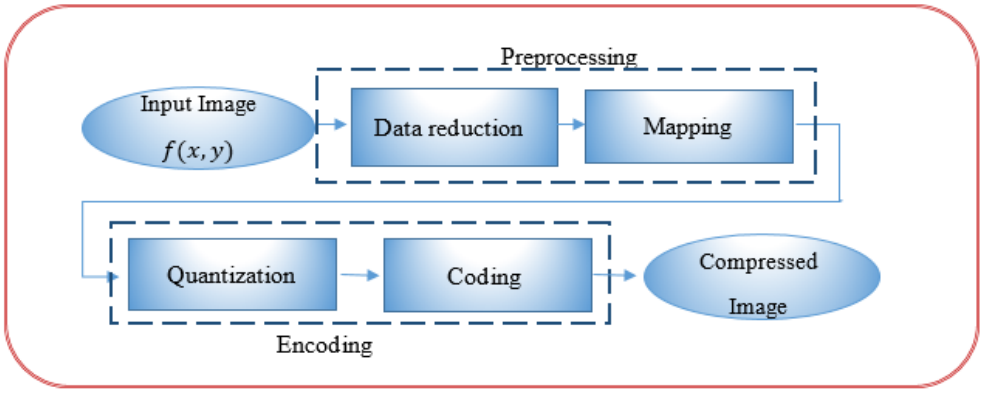


Figure 3.4.2: The Compressor Stages [52]

The de-compressor can be further broken down into the two-stages: decoding stage followed by a post processing stage, as illustrated in Figure3.4.3.

- The first, the decoding stage, takes the compressed file and reverses the original coding by mapping the codes to the original (quantized values).
- Then, these values (de-quantized values) are processed by a stage that performs an inverse mapping to reverse the original mapping process.
- Ultimately, the image may be post processed to enhance the final image.

Before encoding stage, pre-processing is performed to prepare the image for the encoding process, and consists of a number of operations that are application specific. After the compressed file has been decoded, post processing can be performed to eliminate some of the possibly undesirable artifacts brought about by the compression process. Oftentimes, many practical compression algorithms are a combination of a number of different individual compression techniques. A compression algorithm may consist of all these stages, or it may consist of only one or two of these stages.

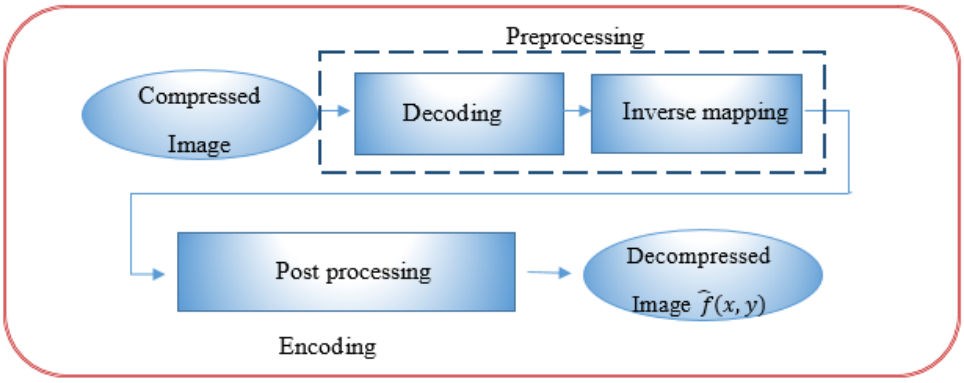


Figure 3.4.3: The Decompression Stages [52]

3.5 Classification of Compression Algorithms

As mentioned previously in section 3.2, the data compression algorithms can be classified in two categories-***lossless*** and ***lossy***. Lossless compression can only achieve a modest amount of compression [1]. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. Therefore, it is impossible to produce an exact replica of the original image when the image is reconstructed. However, lossy schemes are capable of achieving much higher compression.

Transform coding, such as DFT and its invariants DCT, DST and DWT were used effectively in lossy image compression scheme. These transformations transform the image from its spatial domain representation in to a different type of representation then code the transformed values (coefficients). On the other hand, the most robust and quantization technique used for the image compression is ***vector quantization*** (*VQ*). The following subsection describes the concept of VQ scheme and its application in the image compression.

3.5.1 Vector Quantization (VQ)

Vector quantization is one form of lossy compression technique because the quantization is a many-to-one mapping that replaces a set of values with only one representative value [53]. By definition, after this mapping the original value cannot be recovered exactly.

Mathematically, a vector quantizer (*VQ*) of dimension *k* and size *N* is a mapping of a vector in *k*-dimensional Euclidean space, $\mathbb{R}^k$, in to a finite subset Y of $\mathbb{R}^k$ containing *N* reproduction points, thus [6],

$$Q: \mathbb{R}^k \rightarrow Y \qquad (3.5.1.1)$$

The most robust and quantization technique used for the image compression is vector quantization (*VQ*). The idea is similar to that of "rounding-off" (the nearest integer). It is an approximator.

Vector Quantization (*VQ*) includes four stages: vector formation, Training set selection, codebook generation (partition) and quantization. The first step is to split the input image into set of vectors. Taken subset of vectors in this set later and chosen as a training sequence. The codebook of code words is acquired by an iterative clustering algorithm. Lastly, in quantizing an input vector, closest code words in the codebook is specified and corresponding label of this code word is transmitted [54]. Figure (3.5.1.1) depicts the basis processing steps of *VQ* system.

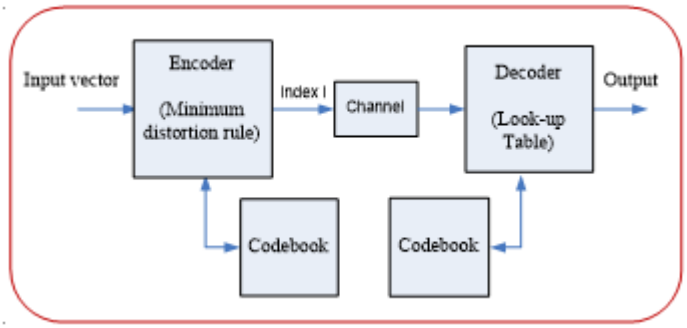


Figure 3.5.1.1: Basic Processing Steps of VQ System.

In the vector space $\mathbb{R}^k$, a vector quantizer maps k-dimensional vectors into a finite set of vectors $Y = \{y_i : i = 1, 2, \ldots, N\}$. Each vector y_i is called a code vector (or a codeword), and the set of all the code vectors (or codewords) is called a ***codebook***. Related with each codeword y, is a nearest neighbor region called encoding region (or Voronoi region) and it is defined by:

$$V_i = \left\{x \in R^k : ||x - y_i|| \leq \left||x - y_j|\right|, for all j \neq i\right\} \qquad (3.5.1.2)$$

The set of encoding regions division the entire space $\mathbb{R}^k$ such that [55]:

$$\bigcup_{i=1}^{N} V_i = R^k \;\; ; \;\; V_i \cap V_j = \emptyset \quad for\ all\ i \neq j \qquad (3.5.1.3)$$

Thus the set of all encoding regions is called the partition of the space.

In Figure 3.5.1.2, we take vectors in the two-dimensional case without loss of generality. Input vectors are marked with green dots, codewords are marked with red stars, and the Voronoi regions are divided with boundary lines. The figure shows some vectors in space. Related with each cluster of vectors is a representative codeword, and each codeword resides in its own Voronoi region which is separated with imaginary boundary lines. When given an input vector, the codeword that is selection to represent it is the one

in the same Voronoi region. The representative codeword is specified to be the closest in Euclidean distance from the input vector [55].

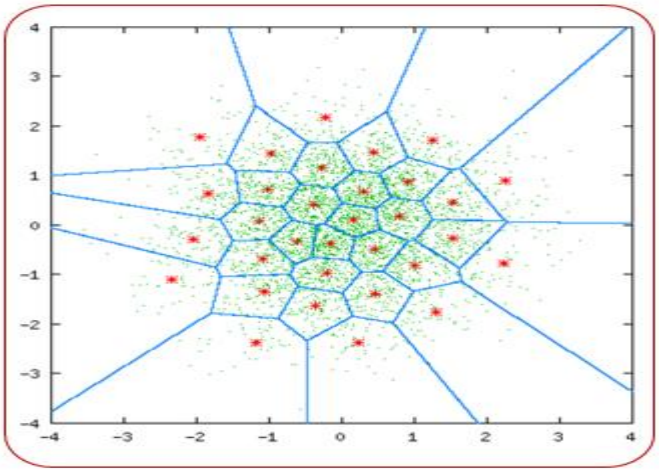


Figure 3.5.1.2 : Codewords in 2-Dimentional Space [55]

The best-matched codevector is chosen using a minimum distortion rule. The most widely used distortion measure in *VQ* is the squared Euclidean distance between the input vector x and its corresponding codevector y. This measure is given by:

$$d(x,y) = \sqrt{\sum_{j=1}^{k}(x_j - y_j)^2} \qquad (3.5.1.4)$$

where x_j, and y_j are the j^{th} elements of the vectors x and y [55].

3.6 An Overview of Image Compression Standard

This section explains the most important compression method which its concepts are used in the proposed model system in this thesis. This method is the well-known image compression standard JPEG2000 (Joint Photographic Experts Group 2000).

JPEG2000 [1]:

In 1997, the new standard for image compression was introduced named JPEG2000 that uses the DWT because of their data reduction capacity. The JPEG 2000 outperforms the other standard compression technique called JPEG in several aspects. The JPEG standard is lossy still image compression method that uses Discrete Cosine Transform DCT.

In a wavelet compression system, the entire image is transformed and compressed as a single matrix rather than block by block as in a DCT-based compression system. The DWT offers adaptive spatial-frequency resolution (better spatial resolution at high frequencies and better frequency resolution at low frequencies) that is well suited to the properties of a Human Visual System (HVS). It can provide better image quality than DCT, especially on a higher compression ratio.

JPEG2000 depends on three fundamental stages which are ***transformation***, ***quantization*** and ***coding***. The first stage explains the using of DWT to transform the data from time domain to mathematical space as explained before in chapter (2). Wavelet compression technique uses the wavelet filters for image decomposition. The image is divided into ***approximation*** and ***detail*** sub images. The forward discrete wavelet transform graphically represented in Figure 3.6.1 involves two filters, one is

corresponding to the ***scaling*** filter or low-pass filter, and the other is corresponding to ***wavelet*** filter or high-pass filter. Moreover, it is noticed that low-pass filter leads to the ***scaling*** function and the high-pass filter leads to ***wavelet*** function.

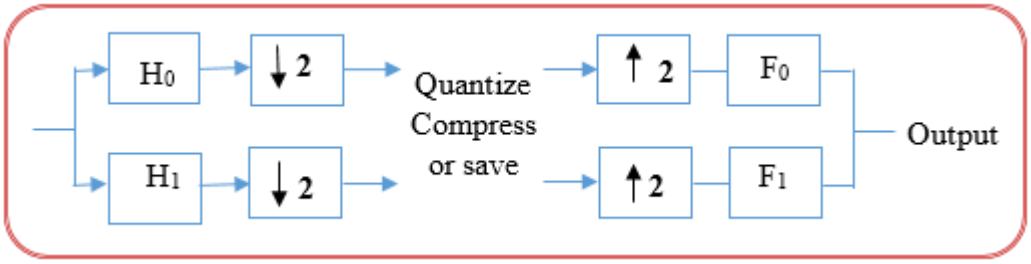


Figure 3.6.1: JPEG2000 Stages Graph [1]

The basic idea of a filter is that it shows an analysis consisting of two filters, a low-pass filter H_0 and a high-pass filter H_1. The low-pass filter works convolution to remove the high frequencies from the input signal *f(x,y)* and let the low frequencies through. The high-pass filter does the opposite. Together, they separate the input into frequency bands (levels), so the outputs of the analysis are called subband coefficients. The filters divide the input image into four non– intersecting multi-resolution coefficient sets, a lower resolution approximation image (LL_1) as well as horizontal (HL_1), vertical (LH_1) and diagonal (HH_1) detail components. The subband LL_1 represents the ***scaling*** function where the scaling function for 2-D DWT can be obtained by multiplying two 1-D scaling functions defined in Eq. (2.4.2.1.1). However, the coefficient sets HL_1, LH_1 and HH_1 represent the ***wavelet*** functions where the wavelet functions for 2-D DWT coefficients can be obtained by multiplying two wavelet functions or wavelet and scaling function. It follows that for 2-D case there exist three wavelet functions that analysis details in ***horizontal***, ***vertical*** and ***diagonal*** as defined in Eqs. (2.4.2.1.2), (2.4.2.1.3), and

(2.4.2.1.4) respectively. The Esq. (2.4.2.1.2 – 2.4.2.1.4) are used efficiently in the proposed first model in this thesis, for determining the orientations of discontinuities.

Figure (3.6.2-a) explains the one-Level 2D wavelet decomposition. $H_0(x)$ and $H_1(x)$ are horizontal low-pass and high-pass filter functions whereas $H_0(y)$ and $H_1(y)$ are vertical low-pass and high-pass filter functions respectively. $f_L(x, y)$ and $f_H(x, y)$ are horizontal low-pass and high-pass wavelet coefficients respectively; approximation, horizontal, vertical and diagonal details are respectively represented by $f_{LL}(x,y)$, $f_{LH}(x,y)$, $f_{HL}(x,y)$ and $f_{HH}(x,y)$.These operations can be reversed with the same filters to get an image reconstruction. This is illustrated in Figure (3.6.2-b).

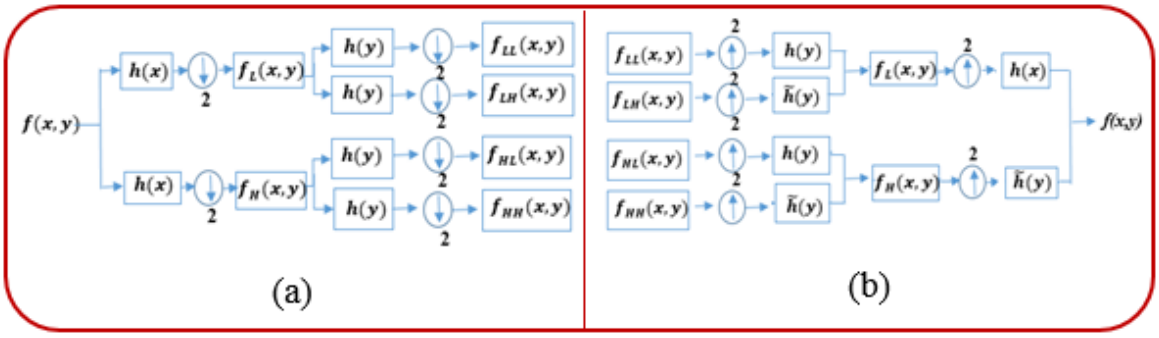


Figure 3.6.2: Two–Dimensional Wavelet Decomposition and Reconstruction[1]
(a)One – Level, Two–Dimensional Wavelet Decomposition.
(b)One – Level, Two–Dimensional Wavelet Reconstruction.

The outputs of the low-pass filter H_0 are normally passed through the analysis filter several times, creating shorter and shorter outputs that show in the Figure 3.6.3 [56]. They can be quantized by using threshold. It is the second stage in which each subband can have a different quantization step size [1]. Figure 3.6.7 shows the two-level transform of the original "Lena" image.

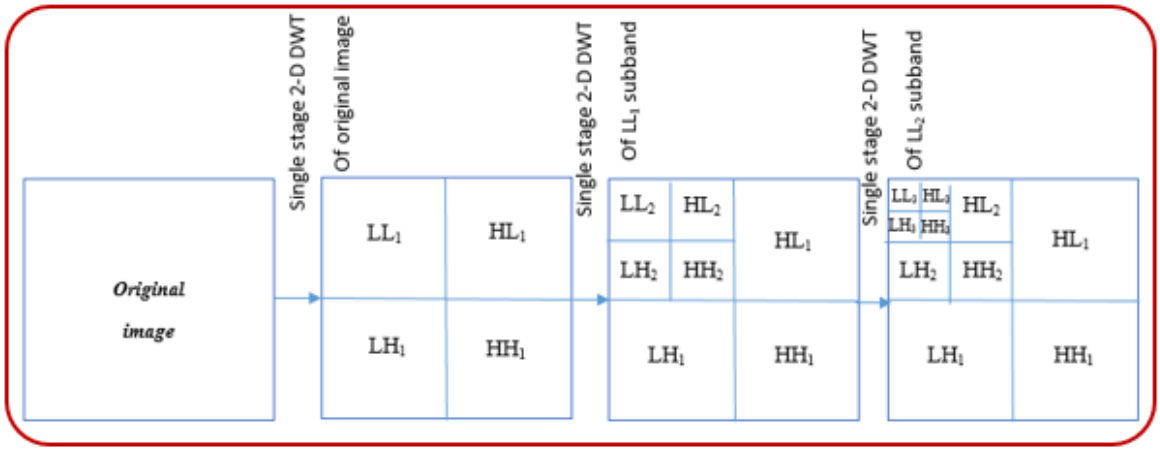


Figure 3.6.3: Three- Bands (Level) 2-D DWT Decomposition of an Image [56]

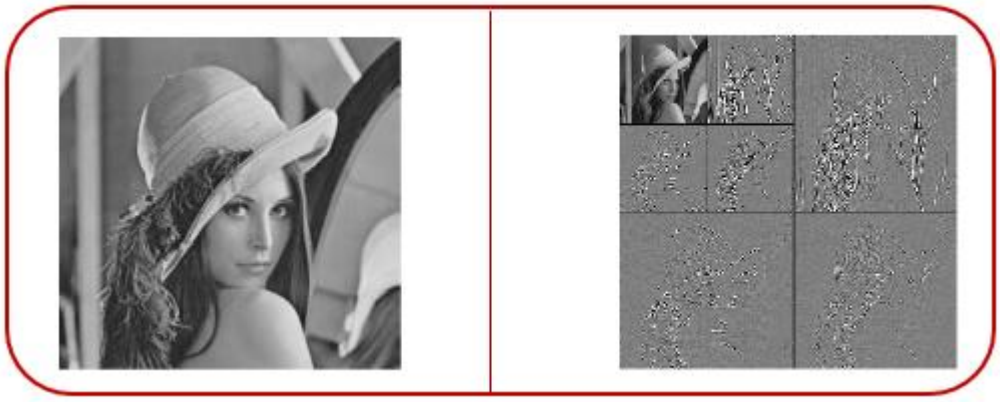


Figure 3.6.4: Two-Level 2-D Wavelet Transform of Lena Image [56]

Each wavelet coefficient in the subband is divided by the quantization step size and the result is truncated. The quantization step size may be determined iteratively in order to achieve a goal bitrates (i.e., the compression factor may be specified in advance by the user) or in order to achieve a predetermined level of image quality. In the third stage, the output quantization coefficients can be coded by Huffman coding, for example.

CHAPTER FOUR

PROPOSED COMPRESSED MODELS

4.1 Introduction

This chapter contains the proposed compressed models, the experimental results of the application, discussions, conclusions and future works. Two models were proposed. The first model consists of using the *DWT* and *SVD* based classified *VQ (CVQ)*, while the second model consists of implementing the *DST* and *SVD* based *CVQ*. Five grey scale images such as (Barbara, Peppers, Lena, Baboon and Goldhill) were used to evaluate the efficiency of the proposed models. These images have different size and details, gray (8-bit) and BMP format. We used mathematical software MATLAB to compress the image data.

4.2 First Proposed Compression Model

An efficient image compression technique using *SVD* based *CVQ* and *DWT* in both the spatial and frequency domains for the efficient representation of still images. The proposed model combines the properties of *SVD, CVQ,* and *DWT*; while avoiding some of their limitations. A simple but efficient classifier based gradient method in the spatial domain, which employs only one threshold to determine the class of the input image block into one of finite number of classes, and uses only the first level of the *DWT* coefficients to determine the orientation of the block without employing any threshold that results in a good image quality was utilized. *SVD* method was used for efficient construction of the classified codebooks. The proposed technique was benchmarked with the conventional approach based *VQ*, existing methods using *CVQ*; and JPEG-2000 image compression techniques. Simulation results indicated that the proposed approach alleviates edge degradation and can

reconstruct good visual quality images with higher *PSNR* than the benchmarked techniques.

Like any compression system, the new proposed technique consists of an encoder and a decoder. At the encoder, a block classification process will be performed to classify the image blocks into shade or edge (discontinuous) blocks. *SVD* based *CVQ* method is used for the reconstructions of the various codebooks. The next sections explain the implementation of block classification and construction of the codebook, and the encoder and the decoder operations.

- **Block Classification:**

The proposed model assumes that any edge information within a small image block can be described by a straight line across the block with an abrupt change of intensity (value) in the spatial domain. However, the assumption that any edge segment within a block is a straight line starts to fail for 6×6 blocks and bigger, and therefore a block size of 4×4 was utilized. The discrete gradient of the block is used as a measure of the edge content of the block in the spatial domain. The orientation of the edge is used to further classify the edge blocks in the frequency domain. Normally, for 4×4 image blocks, the orientations are restricted to four types: horizontal, vertical and two diagonals. Only first level of *DWT* decomposition is employed to determine the orientation of the edge block.

At the outset, the block mean value is calculated and subtracted from each pixel in the block.

Let $B = \{b_{ij}\,; 1 \le i\,,\ j \le 4\}$ represents a 4x4 image block. In this case b_{ij} is the gray level pixel value corresponding to position (i, j) of row i and column j in the image block B. The discrete gradients of the block B in the x and y directions are determined as follows:

$$G_x = \frac{1}{8}\left[\sum_{i=1}^{2}\sum_{j=1}^{4} b_{ij} - \sum_{i=3}^{4}\sum_{j=1}^{4} b_{ij}\right] \tag{4.2.1}$$

$$G_y = \frac{1}{8}\left[\sum_{i=1}^{4}\sum_{j=1}^{2} b_{ij} - \sum_{i=1}^{4}\sum_{j=3}^{4} b_{ij}\right] \tag{4.2.2}$$

In general, for an even numbers $n \times m$ block size, the directional derivatives are:

$$G_x = \frac{2}{m\ x\ n}\left[\sum_{i=1}^{\frac{m}{2}}\sum_{j=1}^{n} b_{ij} - \sum_{i=\frac{m}{2}+1}^{m}\sum_{j=1}^{n} b_{ij}\right] \tag{4.2.3}$$

$$G_y = \frac{2}{m\ x\ n}\left[\sum_{i=1}^{m}\sum_{j=1}^{\frac{n}{2}} b_{ij} - \sum_{i=1}^{m}\sum_{j=\frac{n}{2}+1}^{n} b_{ij}\right] \tag{4.2.4}$$

Where $i = 1,2,\dots,m\,; j = 1,2,\dots,n.$ The gradient magnitude within each image block is defined by:

$$|G| = \sqrt{(G_x^2 + G_y^2)} \tag{4.2.5}$$

If the gradient magnitude $|G|$ of the block B is smaller than a threshold T, the block contains no significant gradient and it is classified as a ***shade*** block; otherwise, it will be classified as an ***edge*** block. Once a block is

classified as an *edge* block, the orientation of the *edge* pattern within the block will be computed using *DWT*. First level 2-D *DWT* decomposition will be used to determine the *edge* directions, namely, *HL*, *LH*, and *HH* as follows:

1) Compute each of the edge direction β_θ for $\theta \in \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\right\}$ such that

$$\left.\begin{aligned} \beta_0 &= 2|V| \\ \beta_{\frac{\pi}{4}} &= \frac{4}{3} max\{|H+V+D|, |H+V-D|\} \\ \beta_{\frac{\pi}{2}} &= 2|H| \\ \beta_{\frac{3\pi}{4}} &= \frac{4}{3} max\{|H-V+D|, |H-V-D|\} \end{aligned}\right\} \quad (4.2.6)$$

Where,

$$H = \frac{1}{2}A\,, V = \frac{1}{2}B\,, D = \frac{1}{2}C\,, A = HL\,, B = LH\,, and\ C = HH.$$

2) The θ value whose measure is the largest will be selected as the block edge orientation of the block.

Once the block classification process has been completed, five different sub-codebooks are generated, representing the different orientations of edge block information and the shade block. *SVD*-based *CVQ* is used for designing the sub-codebooks corresponding to each class. Different rank values have been used in the codebook generating process according to the type of the codebook. Figure 4.2.1 shows the classification process.

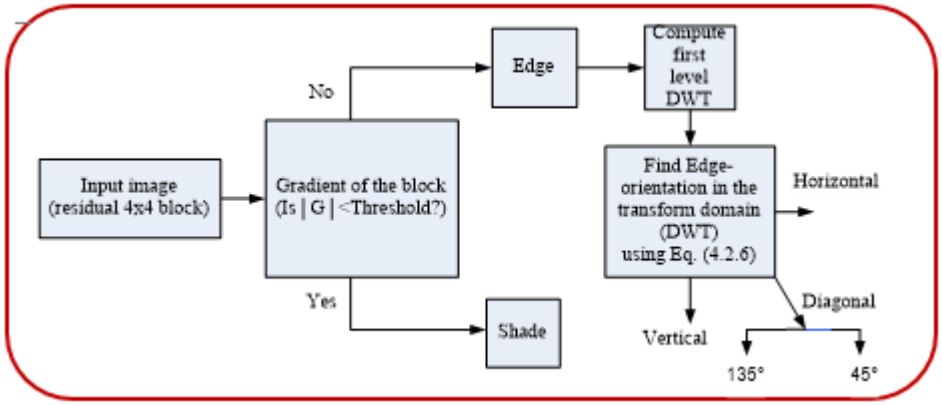


Figure 4.2.1: Block Diagram of the Classification Process

- **Construction of the Codebook:**

Two standard 512×512 monochromatic images, Barbara and Peppers, are used for codebook constructions. The selected training images are divided into small non-overlapping blocks of size 4×4 pixels making the vector dimension equal to 16. The block mean value is computed and subtracted from each pixel in the block and then the classification process is performed as shown above. The process is resulted in a better utilization of codevectors for encoding shade blocks because they are mapped into small regions near the origin where they can be encoded efficiently. Visually sensitive edge blocks and shade blocks are encoded with a codebook specifically designed for that class of blocks so that distortion (error) is minimized. The value of the threshold T is determined experimentally and set to 15 to obtain a reasonable percentage of the shade blocks. Once the block classification process has been completed, different sub-codebooks corresponding to the different classes are generated, using *SVD*-based *CVQ* technique with different rank values. As *VQ* scheme requires operations in multidimensional space, the utility of the *SVD* for dimensionality reduction purpose may alleviate the complexity of a

scheme based *VQ*. The different five sub-codebooks corresponding to the classified classes were generated by projecting the n-dimensional dataset of a class to the space spanned by the *m (m < n)* most significant singular vectors of the *m (m < n)* largest singular values. That is, the sub-codebooks were generated from the eigenvectors of a set of image in a corresponding class.

- **Encoder:**

The encoder of the proposed model operates on the residual blocks where the mean value of the input image block is subtracted from each pixel in the block to yield a residual block (vector). The mean values are then encoded separately. This is because the mean values are usually highly correlated with adjacent mean values. The first and most obvious way, of coding the mean in a useful way is via *scalar quantization*. By implementing this type of quantization scheme, the mean values are encoded with 6-bits value for a 256 level gray scale image based on both the desired quality and rate, using prediction method together with an improved greyscale (IGS) quantization method [45].

The next step is to encode the image by decomposing the image blocks into shade and edge blocks using Eq. (3.2.4). If a block is classified as an edge block, the orientation of the edge pattern within the block will be computed using Eq. (4.2.5). Then it encodes the two types of blocks (the shade block and the edge-orientation block) separately using *CVQ* scheme. The input image is divided into small non-overlapping blocks of size 4×4 pixels which are then processed independently. Each input image block is compared with the closest codeword in the codebook of the same type using the *MSE* as the distortion

measure. The codeword with the lowest *MSE* is selected as the block compressor. This technique reduces the computational complexity by comparing the input image block with only those of the same type to obtain the closest codeword. Figure 4.2.2 shows the block diagram of the proposed model encoder.

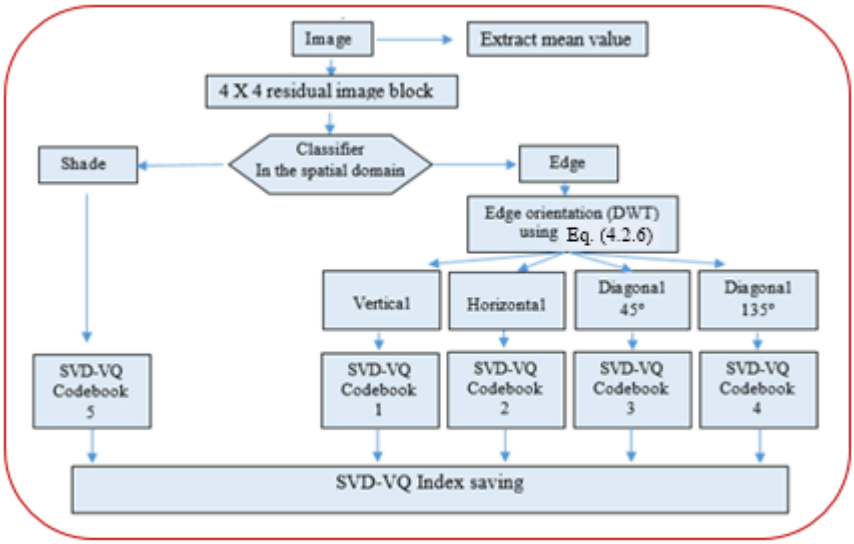


Figure 4.2.2: Block Diagram of the Proposed Model Encoder

- **Decoder:**

The decoder performs simple table look-up operations to retrieve the corresponding codeword from the same codebook as the encoder used, and computes the inverse of the used transforms. As the residual block is used in the encoding process, the block mean value is added to the reconstructed image block.

- **Simulation Results:**

A database of five grey level images was developed to systematically evaluate the application of the proposed system. MATLAB code was written for the generation of the proposed model. The training set used is obtained from two 512×512 monochromatic images of 8-bit intensity, Barbara512 and Peppers512. Three test images outside the training set: Baboon512, Lena512 and Goldhill256 as well as the two images inside the training set: Barbara512 and Peppers512 were used to evaluate the performance of the proposed model. These test images were coded by the proposed model employing the same codebooks that were used for coding the images Barbara512 and Peppers512. The performance of this model is usually characterized using the MSE and the PSNR as image quality metrics based error, as well as the SSIM defined in Eqs. (3.2.2), (3.2.3), and (3.2.4) respectively.

Tables 4.2.1 and 4.2.2 show the performance of the proposed model characterized by the PSNR and the MSSIM based approaches respectively.

Table 4.2.1: Reconstruction Performance for the Proposed Model Trained on the Top Two Images (Barbara512 and Peppers512) and Generated to the Rest of the Images

	Proposed Model		VQ		JPEG-2000	
Image	**Bitrate (bpp)**	**PSNR (dB)**	**Bitrate (bpp)**	**PSNR (dB)**	**Bitrate (bpp)**	**PSNR (dB)**
Barbara512	0.5639	**33.2273**	0.5604	27.3250	0.5654	33.1516
Peppers512	0.5434	**35.8529**	0.5446	28.5478	0.5333	35.8340
Lena512	0.5270	**36.1468**	0.5282	28.7617	0.5114	36.7676
Baboon512	0.6378	**30.8584**	0.6229	25.5914	0.6534	30.2985
Goldhill256	0.7332	**32.7935**	0.7124	26.7216	0.7554	31.4860

Table 4.2.2: Reconstruction Performance for the Proposed Model Trained on the Top Two Images (Barbara512 and Peppers512) and Generated to the Rest of the Images.

	The Proposed Model		VQ		JPEG-2000	
Image	**Bitrate (bpp)**	**MSSIM**	**Bitrate (bpp)**	**MSSIM**	**Bitrate (bpp)**	MSSIM
Barbara512	0.5639	**0.8664**	0.5604	0.1973	0.5654	0.9087
Peppers512	0.5434	**0.9229**	0.5446	0.2761	0.5333	0.9208
Lena512	0.5270	**0.9300**	0.5282	0.4742	0.5114	0.9427
Baboon512	0.6378	**0.7907**	0.6229	0.1223	0.6534	0.7836
Goldhill256	0.7332	**0.8510**	0.7124	0.1429	0.7554	0.8354

The results in Table 4.2.1 and 4.2.2 show that, in all cases, the proposed model outperformed ordinary *VQ* and show competitive results in comparison to JPEG-2000 standard which was generated using MATLAB [45] in terms of both the *PSNR* and the *MSSIM*. However, the proposed model outperforms JPEG-2000 in some cases where the test images were of high detail type (the images Baboon512 and Goldhill256). This is because of the good approximation of the edge blocks which lie far away from the densely region near the origin. On the other hand, popular transform-based lossy compression techniques tend to introduce artifacts at high frequency signal components

since such details often represent high frequency components in frequency domain.

The simulation results also indicated that the proposed technique needs shorter encoding CPU time (in seconds) than the ordinary *VQ* while the standard JPEG-2000 needs shorter encoding CPU time than the proposed model as indicated in Table 4.2.3 :

Table 4.2.3: Comparison of Computing Time Between the Proposed Model and Each of the Ordinary VQ and JPEG-2000 Technique at Fixed Bitrate.

Image	Bitrate (bpp)	Ordinary VQ using k-means Time (sec.)	proposed model Time (sec.)	JPEG-2000 Time (sec.)
Barbara 512	0.5639	103.31	**16.9573**	5.32
Peppers 12	0.5434	104.65	**9.5317**	5.17
Lena 512	0.5270	103.08	**11.5597**	5.66
Baboon 512	0.6378	104.05	**24.5234**	5.42
Goldhill 256	0.7332	26.39	**6.3960**	1.78

Figure (4.2.3) shows some of the reconstructed compressed images of the proposed model and their corresponding reconstructed error.

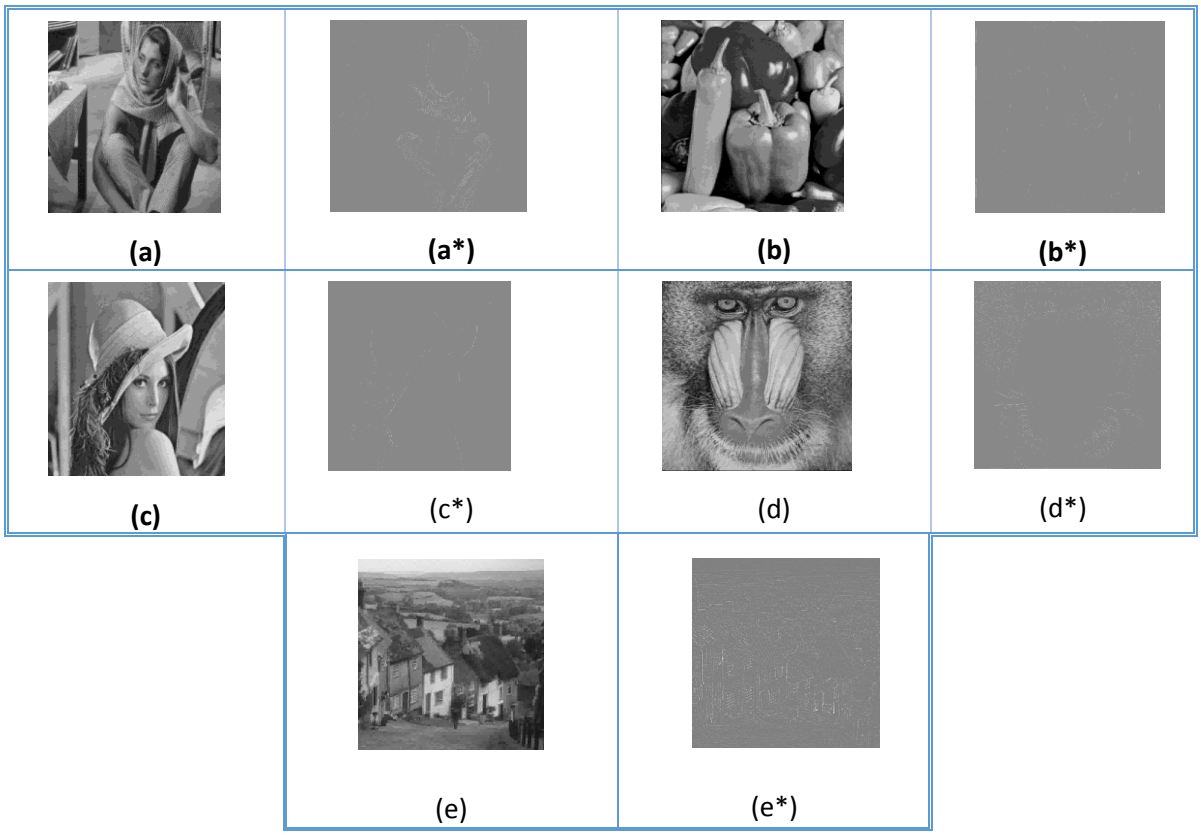


Figure 4.3.1. Some of the Reconstructed Compressed Images by the Proposed Model.

(a)512×512reconstructed image Barbara512 at bitrate 0.5639bpp and psnr= 33.2273dB.

(b)512×512reconstructed image Peppers512 at bitrate 0.5434bpp and psnr= 35.8529dB.

(c)512×512 reconstructed image Lena512 at bitrate 0.5270bpp and psnr=36.1468dB.

(d)512×512reconstructed image Baboon512 at bitrate 0.6378bpp and psnr = 30.8584dB.

(e)256×256reconstructed image Goldhill256 at bitrate 0.7332bpp and psnr = 32.7935dB.

The images a*, b*, c*, d*, e* are the differences between the original images and their reconstructed images a, b, c, d and e plus 128, respectively.

By comparing the proposed scheme with more conventional *CVQ* methods [7], [57], [8], [58], and [59]. It has been noted that the proposed model maintains higher *PSNR* values for the same images at the same bitrate. For example, for the image Lena512, the comparisons are summarized in Table 4.2.4

Table 4.2.4: Comparison Results Between the Proposed Model and More Conventional CVQ for the Image Lena512.This Comparison is Calculated by Using Approximately the Same Bitrate.

	PSNR (dB) values for the image Lena512 by different methods This comparison is calculated by using approximately the same bitrate.					
Bitrate (bpp)	**Reference [7]**	**Reference [57]**	**Reference [8]**	**Reference [58]**	**Reference [59]**	**Proposed Model**
0.625	-	31.26	-	32.65	36.7972	**37.0248**
0.688	-	31.79	-	33.27	37.1509	**37.3145**
0.70	29.79	-	-	-	37.1885	**37.3581**
0.750	-	32.23	-	33.80	37.4166	**37.4639**
0.530	-	-	34.14	-	36.0838	**36.3964**
0.572	-	-	34.49	-	36.4723	**36.7124**
0.600	-	-	34.74	-	36.6557	**36.7943**

As a conclusion, the combined mathematical properties of each of *SVD* based method and *DWT* result in building an efficient coding method for accurate reconstruction of still images at low bit-rates. The quality of the reconstructed images by the proposed scheme is preserved. Edges are reproduced faithfully and their jaggedness is greatly reduced. This method also shows an advantage in *PSNR* and *MSSIM* over the standard *VQ* method using the *k*-means algorithm and the existing methods using *CVQ* scheme; and competitive to the JPEG-2000, for similar values of the bit-rate.

4.3 Second Proposed Compression Model:

Here, the same approach as the first one in section 4.2 is adopted, except, this model makes use *DST* instead of *DWT* to determine the orientation of the edge blocks. Only three AC coefficients of *DST* coefficients are employed. Three AC coefficients of *DST* coefficients will be used to determine the orientation of the edge block, namely S(0,1), S(1,0), and S(1,1) as follows:

1) Compute each of the edge direction β_θ for $\theta \in \left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\right\}$ such that

$$\left.\begin{aligned} \beta_0 &= 2|V| \\ \beta_{\frac{\pi}{4}} &= \frac{4}{3} max\{|H+V+D|, |H+V-D|\} \\ \beta_{\frac{\pi}{2}} &= 2|H| \\ \beta_{\frac{3\pi}{4}} &= \frac{4}{3} max\{|H-V+D|, |H-V-D|\} \end{aligned}\right\} \quad (4.3.1)$$

Where, $H = \frac{1}{2}S(0,1)$, , $V = \frac{1}{2}S(1,0)$,, $D = \frac{1}{2}S(1,1)$

2) The θ value whose measure is the largest will be selected as the block edge orientation of the block.

Figures 4.3.1 and 4.3.2 show the block diagrams of the classification process and the proposed model encoder, respectively.

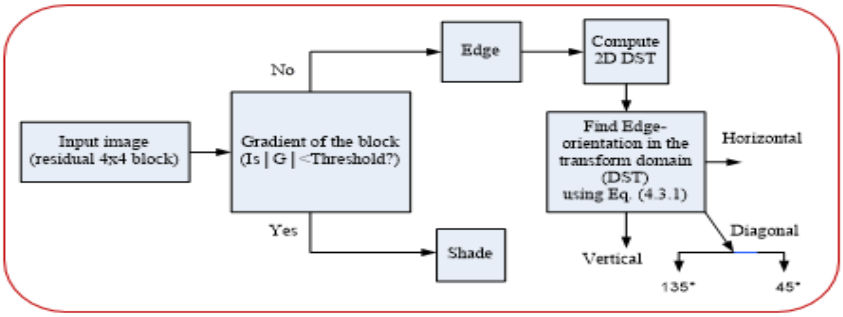


Figure 4.3.1: Block Diagram of the Classification Process

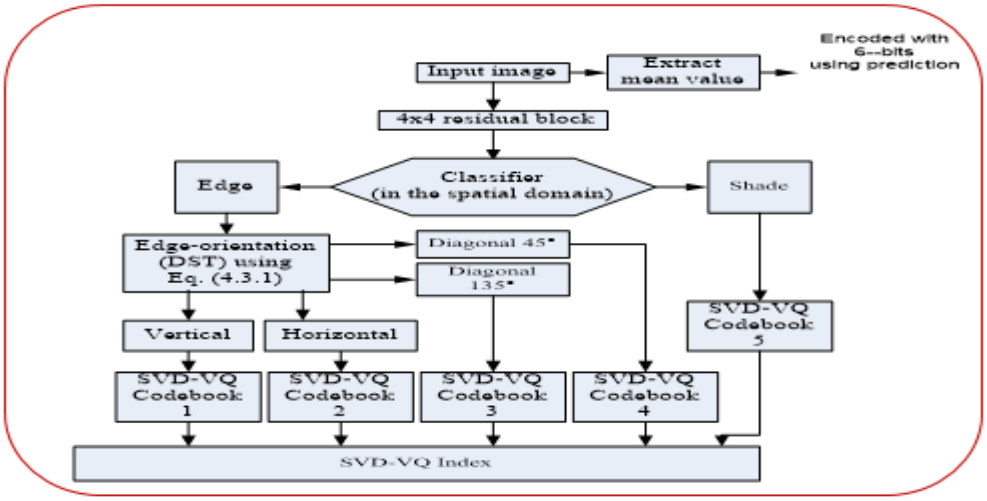


Figure 4.3.2: Block Diagram of the Proposed Model Encoder.

Tables 4.3.1 and 4.3.2 show the performance of the proposed model characterized by the *PSNR* and the *MSSIM* based approaches respectively.

Table 4.3.1: Reconstruction Performance for the Proposed Model Trained on the Top Two Images (Barbara512 and Peppers512) and Generated to the Rest of the Images.

	Proposed Model		VQ		JPEG-2000	
Image	**Bitrate (bpp)**	**PSNR (dB)**	**Bitrate (bpp)**	**PSNR (dB)**	**Bitrate (bpp)**	**PSNR (dB)**
Barbara512	0.5625	**33.3686**	0.5604	27.3250	0.5654	33.1516
Peppers512	0.5427	**35.8856**	0.5446	28.5478	0.5333	35.8340
Lena512	0.5257	**36.1862**	0.5282	28.7617	0.5114	36.7676
Baboon512	0.6526	**30.8762**	0.6229	25.5914	0.6534	30.2985
Goldhill256	0.7511	**32.8406**	0.7124	26.7216	0.7554	31.4860

Table 4.3.2: Reconstruction Performance for the Proposed Model Trained on the Top Two Images (Barbara512 and Peppers512) and Generated to the Rest of the Images.

	The Proposed Model		VQ		JPEG-2000	
Image	Bitrate (bpp)	MSSIM	Bitrate (bpp)	MSSIM	Bitrate (bpp)	**MSSIM**
Barbara512	0.5625	0.8664	0.5604	0.1973	0.5654	**0.9100**
Peppers512	0.5427	0.9230	0.5446	0.2761	0.5333	**0.9208**
Lena512	0.5257	0.9312	0.5282	0.4742	0.5114	**0.9427**
Baboon512	0.6526	0.7897	0.6229	0.1223	0.6534	**0.7836**
Goldhill256	0.7511	0.8549	0.7124	0.1429	0.7554	**0.8207**

The results in Tables 4.3.1 and 4.3.2 show that, in all cases, the proposed model outperformed ordinary *VQ* using the *k*-means algorithm and show competitive results in comparison to JPEG-2000 standard in terms of both the *PSNR* and the *MSSIM*. However, the proposed model outperforms JPEG-2000 in some cases where the test images were of high detail type (the images Baboon512 and Goldhill256). This is because of the good approximation of the edge blocks which lie far away from the densely region near the origin, that result from the mean-removed process, so that the high amplitude vectors may be adequately represented. On the other hand, popular transform-based lossy compression techniques tend to introduce artifacts at high frequency signal components since such details often represent high frequency components in frequency domain.

It is noted that the edge direction is highly correlated with the corresponding DST coefficients. This means that *DST* coefficients appear along the direction perpendicular to the edge direction. This is together with the excellent energy-compaction property of *DST*, the used classifier provides high accuracy in terms of determining each of the strength and the edge-orientation of a given input image block. The properties of *DST* together with the compact energy and low-rank approximation properties provided by *SVD* transform led to high image quality for the proposed model at reasonable bitrate.

Figure (4.3.3) shows some of the reconstructed compressed images of the proposed model and their corresponding reconstructed error.

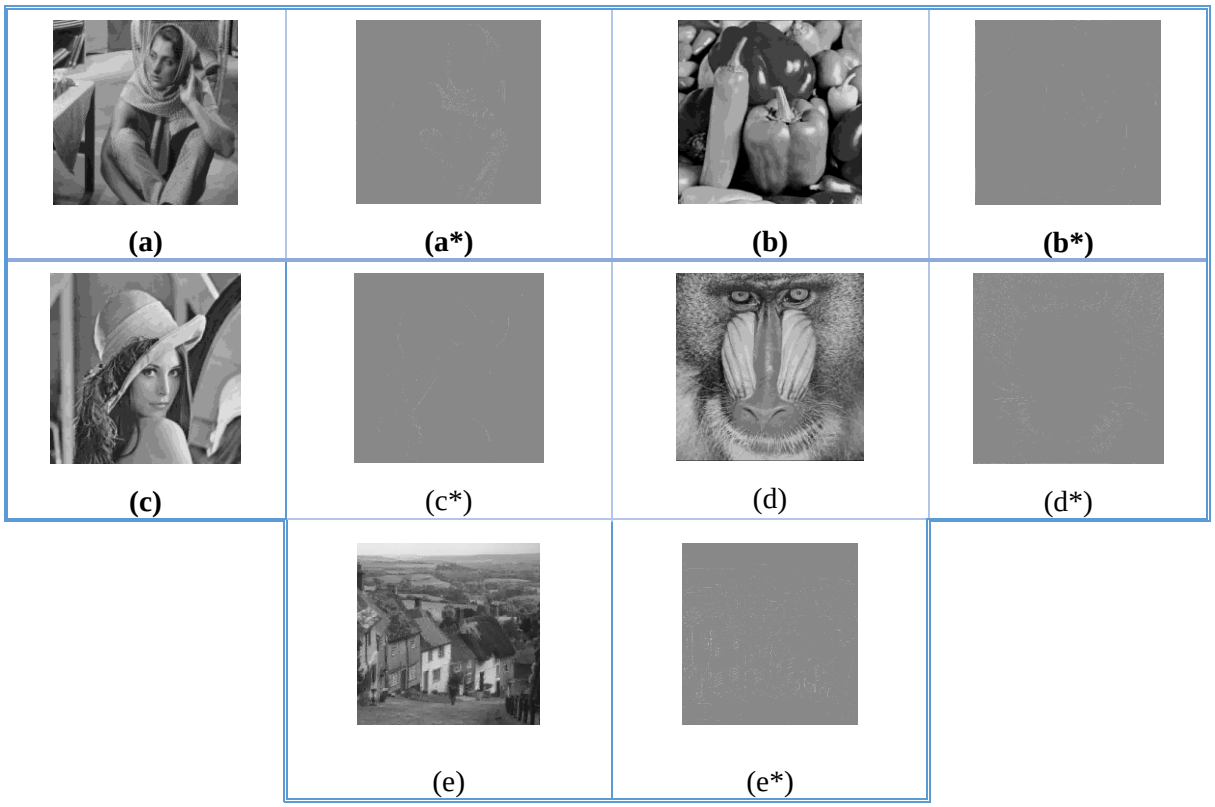


Figure 4.3.3: Some of the Reconstructed Compressed Images by the Proposed Model.

(a)512×512reconstructed image Barbara512 at bitrate 0.5625bpp and psnr = 33.3686dB.

(b)512×512 reconstructed image Peppers512 at bitrate 0.5427bpp and psnr = 35.8856dB

(c)512×512 reconstructed image Lena512 at bitrate 0.5257bpp and psnr=36.1846dB

(d)512× 512 reconstructed image Baboon512 at bitrate 0.6526bpp and psnr = 30.8762dB

(e)256×256reconstructed image Goldhill256 at bitrate 0.7511bpp and psnr = 32.8406dB.

The images a*, b*, c*, d*, e* are the differences between the original images and their reconstructed images a, b, c, d and e plus 128, respectively.

Again, the quality of the reconstructed images by the proposed scheme is preserved. Additionally, the proposed scheme was compared with more conventional CVQ methods [7], [57], [8], [58], and [59]. It has been noted that the proposed model maintains higher *PSNR* values for the same images at the same bitrate. For example, for the image Lena512, the comparisons are summarized in Table 4.3.3.

Table 4.3.3: Comparison Results Between the Proposed Model and More Conventional CVQ for the Image Lena512.This Comparison is Calculated by Using Approximately the Same Bitrate

	PSNR (dB) values for the image Lena512 by different methods. This comparison is calculated by using approximately the same bitrate.					
Bitrate (bpp)	Reference [7]	Reference [57]	Reference [8]	Reference [58]	Reference [59]	**Proposed Model**
0.625	-	31.26	-	32.65	36.7972	**36.8793**
0.688	-	31.79	-	33.27	37.1509	**36.9019**
0.70	29.79	-	-	-	37.1885	**37.3191**
0.750	-	32.23	-	33.80	37.4166	**37.4463**
0.530	-	-	34.14	-	36.0838	**36.3245**
0.572	-	-	34.49	-	36.4723	**36.6729**
0.600	-	-	34.74	-	36.6557	**36.8693**

CHAPTER FIVE

CONCLUSIONS AND SUGGESTIONS

5.1 Introduction

This chapter contains the conclusions and possible future works of the two models were proposed.

5.2 Conclusions

Efficient coding models for accurate reconstruction of still images at low bit-rates were presented in this research work. These models combine *SVD* based *CVQ* as well as the *DWT* and the *DST* in both the spatial and transform domains. The capability of dimensionality reduction of the *SVD* method and the multiresolution analysis property of each of the *DWT* and *DST* were utilized that result in constructing efficient mathematical models for still image compression. The models also show an advantage in *PSNR* and *MSSIM* over the standard *VQ* method and the existing methods using *CVQ* scheme; and competitive to the JPEG-2000, for similar values of the bit-rate.

Different images that have different spatial (time) and frequency characteristics have been selected for measuring the performance of the proposed models. It is noted that the image statistical nature being viewed influences the performance of the compression model. The test image Baboon512 has a lot of details, it contains components in high frequency part, so it presents low redundant image and thus difficult for compression. On the other hand, the test image Lena512 and Goldhill256 are images with less detail than Baboon512. The test image Lena512 does not contain large amounts of high frequency whereas the test image Goldhill256 has components in high frequency area more than the image Lena512.

The results in Tables 4.2.1 , 4.2.2 , 4.3.1 and 4.3.2 show that, in all cases, the proposed models outperformed ordinary *VQ* and show competitive

results in comparison to JPEG-2000 standard which was generated using MATLAB [45] in terms of both the *PSNR* and the *MSSIM*. However, the proposed model outperforms JPEG-2000 in some cases where the test images were of high detail type (the images Baboon512 and Goldhill256). This is because of the good approximation of the edge blocks which lie far away from the densely region near the origin. On the other hand, popular transform-based lossy compression techniques tend to introduce artifacts at high frequency signal components since such details often represent high frequency components in frequency domain. In the case of the test image Lena512, the proposed models show competitive results in comparison to JPEG2000 because this image contains fewer details than the other test images.

The quality of the reconstructed images by the proposed models is preserved. Edges (discontinuities) are reproduced faithfully and their jaggedness is greatly reduced. The good visual quality of the reconstructed images is attributed to the grouping of perceptually similar blocks (matrices) by the classifier and the subsequent codebook design of each class.

The simulation results also indicated that the proposed models need shorter encoding CPU time (in seconds) than the ordinary *VQ* which is time consuming. The faster training and shorter encoding time of the proposed models than the standard *VQ* is due to the design of the encoding part that utilizes the properties of the *SVD* method. However, the standard JPEG-2000 needs shorter encoding CPU time than the proposed models as indicated in Table 4.2.3 since it works direct in frequency domain.

Additionally, Tables 4.2.4 and 4.3.3 show that the performance of the proposed schemes were compared with more conventional CVQ methods [7],

[57], [8], [58], and [59]. It has been noted that the proposed models maintain higher *PSNR* values for the same images at the same bitrate.

Finally the performance of the second proposed model in which the *DST* made use instead of *DWT* for determining the orientation of the discontinuities blocks showed some gain in reconstructed image quality than the first model. This is due to the good utilization of the multiresolution property of the *DST* as shown in Tables 4.2.1 and 4.3.1.

5.3 Suggestions:

The classification process can be improved by using more effective mathematical methods relating to gradient measure. On the other hand, the type of visible error such as the staircase and the blurring may decrease by studying the boundary properties of sub images (matrices).

REFERENCES

References

[1]. David S., "**Data Compress: The Complete Reference**", 4th Edition, Springer-Verlag London Limited, 2007 .

[2]. Gilbert S., "**Introduction to Linear Algebra**", 3ed Edition, 2003.

[3]. Tian M., Si-Wei L., and Ling-Zhi L., "**An Investigation into Using Singular Value Decomposition As A Method of Image Compression**", Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference, Vol. 8, 5200-5204, 18-21 Aug.2005.

[4]. Dapena A. and Ahalt S., "**A Hybrid DCT-SVD Image-Coding Algorithm**", IEEE Trans. Circuits Syst. Video Technol., Vol. 12, No. 2, 114-121, Feb. 2002.

[5]. Aase S.O., Husoy J. H. and Waldemar P., "**A Critique of SVD-Based Image Coding Systems**", IEEE International Symposium on Circuits and Systems VLSI, Vol. 4,13-16, Orlando FL, May 1999.

[6]. Gersho A. and Gray R. M., "**Vector Quantization and Signal Compression**", Kluwer Academic Publishers, 1992.

[7]. Ramamurthi B. and Gersho A., "**Classified Vector Quantization of Images"**, IEEE Trans. Commun. COM-34, 1105-1115, 1986.

[8]. Kim J. W. and Lee S. U., "**A Transform Domain Classified Vector Quantizer for Image Coding"**, IEEE Trans. Circuits and Systems for Video Technology, Vol. 2, No. 1, 3-14, 1992.

[9]. Mirko C., Anita G., and Zvonimir L., " **Classified Vector Quantization and its Application on Compression of IRIS Images in the Safety of Marine Systems**", Information and Communication Technology Preliminary Communication, Vol. 28, No. 2, 125-131,2016.

[10]. Amara G., "**An Introduction to Wavelets**", 1st section, IEEE Computational Science and Engineering, vol. 2, no. 2, 1995.

[11]. Jean L. Starck, Fionn M., and Albert B., "**Image Processing and Data Analysis The Multiscale Approach**", 1988.

[12]. Hannu O., "**Discrete Wavelet Transforms : Algorithms and Applications** ", 2011.

[13]. Myung S. Song, "**Wavelet Image Compression**", Contemporary Mathematics ,Mathematics Subject Classification. Primary 42C40, 1991.

[14]. Jain Y.K. and Jain S., "**Performance Evaluation of Wavelets for Image Compression**", Asian Journal of Informational Technology 5 (10): 1104-1112, 2006.

[15]. Banti D. and Vivek A., " **Image Compression Using DCT and DWT** ", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 6, June 2015.

[16]. Harjeetpal S. and Sakhi S., "**Hybrid Image Compression Using DWT, DCT & Huffman Encoding Techniques**", International Journal of Emerging and Advanced Engnerring, Vol 2, Issue 10, 2012.

[17]. Ezhilarasan M. and Thambidurai P., "**A Hybrid Transform Technique for Video Coding**", LNCS, vol.4308, pp. 503-508, 2006.

[18]. Usama S. Mohammed, "**Highly Scalable Hybrid Image Coding Scheme**", Digital Signal Processing, Science Direct, vol.18, pp. 364-374, 2008.

[19]. Usama S. Mohammed and Abdelhafiez W. M., "**Image Coding Scheme Based on Object Extraction and Hybrid Transformation**

Technique", Int. J. of Engineering Science and Technology, vol.2, no.5, pp. 1375-1383, 2010.

[20]. Zhijun F., Yuanhua Z., and Daowen Z., "**A Scalable Video Coding Algorithm Based DCT DWT**", in Proc. 3rd IEEE Int. Symp. Signal Processing and Information Technology, 2003, pp. 247-249.

[21]. Jiang S. and Hao X., "**Hybrid Fourier-Wavelet Image Denoising**", Electronics Letters, vol.43, no.20, pp. 1081-1082, 2007.

[22]. Howard A. and Chris R., "**Elementary Linear Algebra, Application**", Version,10th Ed. John Wiley & Sons, Inc m 2010.

[23]. Serge L., "**Linear Algebra**", Springer, third edition, 1989.

[24]. Walnut D., "**An introduction to Wavelet Analysis**", Boston: Birkhauser Basel; 2002. pp.449.

[25]. Boggess A.L. and Narcowich F. J. A., "**First Course in Wavelets With Fourier Analysis**", pp.292.

[26]. Julius O. Smith III, "**Introduction to Digital Filters With Audio Applications**", Center for Computer Research in Music Acoustics (CCRMA), Stanford University, Stanford, California, 2ed priniting, 2008.

[27]. Smith S. W., "**The Scientist and Engineer's Guide to Digital Signal Processing**", California Technical Publishing San Diego, California, 2nd Edition, 1999.

[28]. Körner T. W., "**Fourier Analysis**", Cambridge University Press, Cambridge, 1996.

[29]. Alfred M., "**Signal Analysis; Wavelets, Filter Banks, Time-Frequency Transformsand Applications**", John Wilet& sons: New York; 1999. pp.53.

[30]. Shih F. Y., "**Image Processing and Pattern Recognition Fundamentals and Techniques**", A John Wiley & Sons, Inc., publication , IEEE, 2010.

[31]. Jain A. K., "**Fundamentals of Digital Image Processing**", Prentice Hall,Inc.,1989.

[32]. Joshi M. A., "**Digital Image Processing An Algorithmic Approach**", PHL Learning Private Limited, New Delhi., 2009.

[33]. Mrinal M., Amir A, "**Continuous and Discrete Time Signals and Systems**", by Content Technologies, Inc., 1st Edition, 2007.

[34]. Clarke R. J., "**Transfer Coding of Images**", Academic Press, London, 1985.

[35]. Oppenheim A.. V. and Schafer R. W., "**Discrete-Time Signal Processing**", Englewood Cliffs, NJ: Prentice-Hall, 1989.

[36]. Robert X Gao and Ruqiang Y., "**Wavelets : Theory and Applications for Manufacturing**", Springer Science and Business Media, LLC, 2011.

[37]. Zahir M. Hussain, Amin Z. Sadik, and Peter O., "**Digital Signal Processing : An Introduction with MATLAB and Applications**", Springer, 2011.

[38]. Vladimir B., Patrick C. Yip, and Rao K. R., "**Discrete Cosine and Sine Transforms ; General Properties, Fast Algorithms and Integer Approximations**", Academic Press is an imprint of Elsevier, 1st Edition, 2007.

[39]. Priyanka J., Balbir K., and Shail B. Jain, "**A General Design for One Dimensional Discrete Sine Transform**", Springer Science and Business Media, LLC, Process 61:211–214, 2009.

[40]. Koti D. Reddy, Jayanandan T. , and Vijay C.L Kumar, "**Speech Enhancement through Elimination Of Impulsive Disturbance Using Log MMSE Filtering**", International Journal Of Engineering And Computer Science, Volume 2, Issue 12, 2013.

[41]. Grossman , A., Morlet, J., "**Decomposition of Hardy functions into square integrable wavelets of constant shape**", SIAM J. Mth. Anal, 15(4) : 723-36,1984.

[42]. Mohammed G., " **Standard codes image compression to advanced video coding**", 3rd Edition, 2011.

[43]. Sven N. , " **Wavelet Based Spiht Compression For Dicom Images** ", Linnaeus University, 2011.

[44]. Sidney C. Burrus, Ramesh A. Gopinath, and Haitao G., "**Introduction to Wavelets and Wavelet Transforms**", Prentice-Hall, 1998.

[45]. Rafael C. Gonzalez and Richard E. Woods, "**Digital Image Processing**", 3rd Edition, 2008.

[46]. Mallat S., "**A Theory for Multiresolution Signal Decomposition**", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, pp. 674-693, July 1989.

[47]. Ali N. Akansu and Richard A. Haddad, "**Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets**", New Jersey Institute of Technology, 2ed Edition, 2001.

[48]. Rowayda A. Sadek, " **SVD Based Image Processing Applications: State of The Art, Contributions and Research Challenges**", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 3, No. 7, 2012.

[49]. Daniel P. Berrar, Werner D., and Martin G., " **A Practical Approach to Microarray Data Analysis** ", Kluwer Academic Publishers, 2003.

[50]. Mark N. and Jean L. Gailly, "**The Data Compression Book**" ,2ed edition, 2011.

[51]. Wang Z., Bovik A. C., Sheikh H. R., and Simoncelli E. P., "**Image Quality Assessment: From Error Visibility to Structural Similarity**", IEEE Transactions on Image Processing, Vol. 13, No. 4, 1-14, 2004.

[52]. Scott E Umbaugh, "**Digital Image Processing and Analysis : Human and Computer Vision Applications with CVIP tools**", 2ed Edition, 2011.

[53]. Binit A., and Patel A., " **Vector Quantization Based Lossy Image Compression Using Wavelets – A Review** " International Journal of Innovative Research in Science, Engineering and Technology, Vol. 3, Issue 3, March 2014

[54]. Khadtare T. D., Mayuri C., Sushma B., and Yogita R., "**A Combined Novel Approach for Image Compression Using Vector Quantization and Wavelet Transfhorm**", International Journal of Application or Innovation in Engineering and Management, Vol 3, Issue 4, April 2014.

[55]. Swarup R., Asoke K. Sen, and Nandan K. Sinha. Roy, "**VQ-DCT Based Image Compression: A New Hybrid Approach**", Assam University Journal of Science & Technology : Physical Sciences and Technology, Vol. 5, Number II, pp:73-80, 2010.

[56]. CCSDS Secretariat, "**Image Data Compression**", Report Concerning Space Data System Standards, CCSDS 120.1-G-2, Issue 2, Washington, USA, 2015.

[57]. Kim D. S. and Lee S. U., "**Image Vector Quantizer Based on A Classification in The DCT Domain**". IEEE Trans. Commun. COM-39, 549-556, 1991.

[58]. Quweider M. K. and Farison J. B., "**Classification of Image Edge Blocks by Orthogonal Projection**", Electron. Lett, Vol. 33, No. 22, 1856-1857, 23rd Oct. 1997.

[59]. Ali H. Al-Fayadh, Abir J. Hussain, Paulo L., and Dhiya A., " **Novel Hybrid Classified Vector Quantisation using Discrete Cosine Transform for Image compression"**, J. Electron. Imaging. 18(2), 023003, Doi:10.1117/1.3116564, April 21, 2009.

الملـخص

الغرض الرئيسي من هذا البحث هو دراسة وتحقيق أهم الخصائص من التحويلات التكاملية (تحويل فوريه المنفصل، تحويل الجيب منفصل، وتحويل المويجات المنفصل) وجوانبها الرياضية سواء من الناحية النظرية وتطبيقها لضغط الصورة.

وكذلك، درسنا طريقة تجزئ القيمة المفردة وتطبيقها على تمثيل الصورة.

تم تطوير نموذجين رياضيين. يتكون النموذج الأول من أسلوب التحويل المتعدد الجديد الذي يستفيد من ميزة كل من تحويل المويجات المنفصل وطريقة تجزئ القيمة المنفردة في حين يتناول النموذج الثاني الاستفادة من تحويل الجيب منفصل وطريقة تجزئ القيمة المنفردة. تم تطبيق هذه النماذج في ضغط الصورة . وتشير النتائج إلى أن هذا النهج (المعالجة) الجديد ينتج عنها تمثيل أفضل للدالة تمثيل وظيفة وإعادة بناءها في تطبيق ضغط الصورة اكثر مما هو ممكن مع استخدام تحويل واحد منفرد . النماذج المقترحة تحسّن كفاءة عملية الضغط في مجالي تحويل المويجات المنفصل وتحويل الجيب المنفصلة .

جمهوريـــــــة العـــــراق

وزارة التعليم العالـــي والبحث العلـــــمي

جامعــة النهريــن

كليـة العلـوم

قسم الرياضيات وتطبيقات الحاسوب

بعض نتائج طرق التحويلات الرياضياتية وتطبيقها في ضغط الصورة

رســالة

مقدمة الى كلية العلوم – جامعة النهرين

وهي جزء من متطلبات نيل درجة ماجستير علوم في الرياضيات

من قبل

هديل ماجد رشيد العاني

(بكالوريوس علوم رياضيات ، الجامعة التكنولوجية ، ١٩٩٩)

بـــأشـــراف

أ.م.د. علي حسن ناصر الفياض

1438 2017