1. INTRODUCTION

1.1 PREAMBLE
Visual information is of vital importance for mankind to perceive, receive and understand the surrounding world. The Chinese proverb ‘One picture is worth more than thousand words’ expresses correctly the amount of information contained in a single picture or image. Newspapers, High definition TV and Video Conferencing use still or moving images as information carriers. The need for processing and transmission of these images led to image processing. The art of transforming an image to a digital format and its processing by digital computers is called digital image processing.

1.2 STATEMENT OF THE PROBLEM
New image compression technologies which yield low bit rate while maintaining the fidelity of raw images are sought everywhere. When an image is compressed, it occupies less memory space. Hence, in long distance communication, transmission of compressed images saves time, cost and space.

The objective of this research is to compress grayscale images using Bandelets and Neuro-Statistical methods. Image compression is achieved through the transformation, quantization and encoding procedures. This research work proposes the use of Bandelets for transformation and Neuro-Statistical methods for quantization which employs the well known Huffman encoder for encoding the quantized coefficients.
1.3 MOTIVATION

Image transmission and storage are primarily motivated by the ease and flexibility of handling digital image information instead of the analog information. Transmission and storage capabilities are limited and expensive in nature. Image storage is required most commonly for educational and business documents, medical images used in patient monitoring systems and the like. Images consume more amount of storage. As the transmission and storage of every single bit incurs a cost, the advancement of cost effective image compression techniques is of high significance. In addition to the extremely high storage and bandwidth requirements, use of uncompressed digital images will add significant cost to the hardware system that processes the images.

1.4 IMAGE PROCESSING BASICS

The commonly used terms and the fundamentals in image processing are briefly explained here.

1.4.1 Definitions

This section provides definitions related to image processing. It provides background knowledge required for the image compression methods.

Image

Image is basically an analog signal carrying the continuous variations of light intensity along a horizontal line in the original scene. Images can be classified into several types based upon their form or methods of generation.

- Images that can be seen and perceived by eyes are called visible images.
Images that are formed with lenses like gratings and holograms are called optical images.

Images that are distributions of measurable physical properties like temperature, pressure distribution, distance from observer, etc. are called physical images.

Images of mathematics, the continuous and discrete functions or digital images are called abstract images.

Picture is a representation made by painting, drawing or photography. It is a restricted type of image.

**Digital Image**
A digital image is an array of pixels represented by a finite number of bits. Computers can process only digital images.

**Pixel**
A pixel is defined as a two-dimensional picture element that is the smallest non-divisible element of the digital image.

**Image Resolution**
Image resolution represents the number of pixels in the image.

**Image Digitization**
An image captured by a sensor is expressed as a continuous function $f(x, y)$ of the two co-ordinates in the plane. Image digitization means that the function $f(x, y)$ is sampled into a matrix with M rows and N columns. Sampling and Quantization are the two sub-processes of the image digitization process.
**Sampling**

The process of converting a continuous space / time signal into a discrete space / time signal is known as Sampling.

**Quantization**

The process of converting the continuous range of intensity values of the sampled image into a discrete range is known as Quantization. In most applications, the gray scale images are quantized at 256 levels and require 1 byte (8 bits) for the representation of each pixel.

**Image File Formats**

Image File formats are standardized means of organizing and storing digital images. The header information associated with each file format contains the image attributes. Some *.bmp files and *.jpg files are taken as samples in this work. The commonly used formats are the Joint Photographic Experts Group (JPEG), Tagged Image File Format (TIFF), Graphics Interchange Format (GIF), Windows Bitmap (BMP) and Portable Network Graphics (PNG).

**1.4.2 Digital Image Representation**

Digital image is a discrete two-dimensional function, \( f(x, y) \), where \( x \) and \( y \) are the spatial coordinates and the amplitude value of \( f \) at any pair of coordinates \((x, y)\) is called the intensity or gray level of the image at that point. The values of \( f \) are all finite and discrete for a digital image. The digital image can be conveniently represented by a rectangular matrix \( f(x, y) \) consisting of \( M \) rows and \( N \) columns as shown in Figure 1.1. The resolution of such an image is written as \( M \times N \). By convention, \( f(0, 0) \), is taken to be the bottom left corner of the image, and \( f(M-1, N-1) \), the top right corner.
In image compression, color images are usually represented in the Red, Green and Blue (RGB) format or the Luminance, Chrominance and Saturation (YUV) format. YUV is an abstract form of the RGB format that is more useful for image compression. There are equations that linearly transform RGB images to YUV images and vice versa. If an image compression algorithm is used on the separate color components of an image it would triple the information needed to compress the color image as compared to that of a gray scale image. Therefore this research uses only gray scale images avoiding similar repeated computations for the color components. The compression techniques described in this work can be applied to color images also.

1.4.3 Steps in Image Processing

The first step in image processing is image acquisition. It is the process of acquiring a digital image. The next step is pre-processing of the image. Pre-processing enhances the contrast and removes noise.
The third stage deals with segmentation; that is, partitioning an input image into its constituent parts (or) objects. The fourth stage called representation and description; deals with extracting features that result in some quantitative information of interest or features that are basic for differentiating one class of objects from another.

The fifth stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to recognized objects [1].

1.4.4 Elements of Image Processing Systems

The elements of a Digital image processing system include digitizer, storage devices, processor, communication devices and display devices.

- Digitizer is used for converting analog image into digital image
- Digital storage required for image processing application falls into three principal categories:
  - Short-term storage for use during processing (e.g. main memory)
  - On-line storage for fast recall (e.g. magnetic disks)
  - Archival storage (e.g. magnetic tapes and optical disks)
- Processor is used to process procedures that are usually expressed in algorithmic form
- Communication primarily involves local and remote communication, typically in connection with the transmission of image data
- Monochrome and color monitors are the principal display devices used in image processing systems
1.4.5 Research Areas in Image Processing
The research areas in digital image processing include image enhancement, image compression, image restoration, image segmentation, image representation, image analysis and modeling.

Digital image processing has a broad spectrum of applications such as remote sensing via satellites and storage for business applications, medical processing, radar, sonar, robotics and automated inspection of industrial parts.

1.5 IMAGE COMPRESSION BASICS
This section presents an introduction to image compression, the various phases of an image compression system and the standard performance measures for an image compression algorithm.

1.5.1 Definition
Image Compression is the process of reducing the number of bits needed to represent an image by removing the redundancies as much as possible.

1.5.2 Need for Image Compression
Communication and storage capabilities are limited and expensive in nature. Raw multimedia data contains an immense amount of data that requires certain amount of storage space. For example Land sat D satellite broadcasts $85 \times 10^6$ bits of data every second and typical image from one pass consists of $6100 \times 6100$ pixels in seven spectral bands (i.e.) 260 Megabytes of image data. The data given in Table 1.1 show the qualitative transition from simple text to video data and their uncompressed data storage capacity.
**Table 1.1 Multimedia Data Types and Uncompressed Storage Space Required**

<table>
<thead>
<tr>
<th>IMAGE TYPE</th>
<th>IMAGE SIZE</th>
<th>PIXEL RESOLUTION (bits)</th>
<th>UNCOMPRESSED IMAGE SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black and White</td>
<td>512 x 512</td>
<td>1</td>
<td>262 Kb</td>
</tr>
<tr>
<td>Gray scale Image</td>
<td>512 x 512</td>
<td>8</td>
<td>2.1 Mb</td>
</tr>
<tr>
<td>Color Image</td>
<td>512 x 512</td>
<td>24</td>
<td>6.3 Mb</td>
</tr>
<tr>
<td>Computerized Tomography</td>
<td>512 x 512</td>
<td>12</td>
<td>3.15 Mb</td>
</tr>
<tr>
<td>Computerized Radiography</td>
<td>2048 x 2048</td>
<td>12</td>
<td>50.33 Mb</td>
</tr>
<tr>
<td>A page of text</td>
<td>8.27’ x 11.69’</td>
<td>7 bits/character</td>
<td>14 Kb</td>
</tr>
<tr>
<td>Digitized Video</td>
<td>720 x 576 pixels per frame</td>
<td>24 bits/pixel x 30 frames/sec</td>
<td>298.59 Mb/sec</td>
</tr>
<tr>
<td>Stereo Audio (20Hz -20KHz)</td>
<td>44000 samples per sec</td>
<td>2 channels x 16 bit/sample</td>
<td>1.41 Mb/sec</td>
</tr>
</tbody>
</table>

**Table 1.2 Access Speed and Capacity of Various Storage Devices**

<table>
<thead>
<tr>
<th>STORAGE DEVICE</th>
<th>ACCESS SPEED PER SECOND</th>
<th>STORAGE CAPACITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Brain</td>
<td>100 million MIPS (Million computer Instructions Per Second)</td>
<td>500 to 1000 TB</td>
</tr>
<tr>
<td>Magnetic Disk</td>
<td>70 MB/sec</td>
<td>760 GB</td>
</tr>
<tr>
<td>Flash memory</td>
<td>25 MB/sec</td>
<td>16 GB</td>
</tr>
<tr>
<td>Optical Disk Memory CD 1x speed</td>
<td>150 KB/s</td>
<td>700 MB</td>
</tr>
<tr>
<td>DVD 1x speed</td>
<td>1.385 MB/s</td>
<td>4.70 GB</td>
</tr>
<tr>
<td>Floppy Disk (3½-inch HD)</td>
<td>125 KB/s</td>
<td>1.44 MB</td>
</tr>
</tbody>
</table>
Digital image compression is thus necessary even with exponentially increasing bandwidth and storage capacities. A digital computer represents data using the binary number system. Text, numbers, pictures, audio, and nearly every other form of information can be converted into a string of bits or binary digits, each of which has a value of 1 or 0. The most common unit of storage is the byte, equal to 8 bits. Computers can store and process anything that is represented in binary digits. For example, using eight million bits, or about one megabyte, a typical computer could store a short novel.

Table 1.1 clearly illustrates the need for large storage space for digital image, audio and video data. Although the capacity of several storage media is substantial as given in Table 1.2, their access speeds are usually slow [2]. So at the present state of technology, the only solution is to compress multimedia data before its storage and transmission and to decompress it at the receiver end for play back.

1.5.3 Background for Image Compression

Digital images have significant redundancies. There are two basic data redundancies which can be identified and exploited in images: Statistical redundancy and Redundancy using predictability. Eliminating or reducing these redundancies results in compression.

Compression can be lossy or lossless. Lossless image compression reproduces an identical image after decompression. Lossy compression yields perceptually equivalent, but not identical image compared to the uncompressed image. Statistical redundancy removal and redundancy removal using predictability are the two fundamental principles used in
image compression. Further, image compression can be achieved by representing images in the transform domain rather than in the spatial domain representation.

1.5.3.1 Statistical Redundancy (Lossless)
Statistical redundancy is directly related to the image data probability distribution. There are two kinds of Statistical redundancies namely the coding redundancy and the interpixel redundancy.

- **Coding Redundancy** represents the length of code words assigned to various gray levels of an image. Huffman coding and arithmetic coding schemes aim at assigning fewer bits to the more probable gray levels than the less probable gray levels.

- **Interpixel Redundancy** represents the correlation between neighboring pixels of an image. Run-length coding is a technique that aims at removing spatial redundancy.

1.5.3.2 Redundancy using Predictability (Lossy)
Image redundancy can be described by using predictability in local image neighborhood correlation or significance of the pixel intensity. This type of redundancy is categorized into Psycho-visual redundancy and Predictive Neighborhood redundancy.

- **Psycho-Visual Redundancy**: It represents the presence of irrelevant visual information in an image according to the human visual system. Quantization schemes aim at the elimination of Psycho-visually redundant information.
- **Predictive Neighborhood Redundancy:** The intensity of a pixel is predicted from the intensities of its neighboring pixels, if the image data are correlated within certain image neighborhood. Predictive coding schemes aim at reducing neighborhood redundancy by prediction.

### 1.5.3.3 Image Transforms

Image transforms pack information content present in an image by exploiting the fact that certain orthogonal transforms (e.g., Discrete Cosine transform, Karhunen-Loeve transform, Discrete Sine transform, Walsh-Hadamard transform) can pack image energy into few transform coefficients.

### 1.5.4 Benefits of Image Compression

- Reduces storage requirement and incurs less transmission cost
- Takes lesser time to load than their more cumbersome originals, making it possible to view more images in a shorter period of time
- Reduces the probability of transmission errors
- Provides a level of security against illicit monitoring

### 1.5.5 Image Compression Model

Image Compression system design consists of two parts. In the first part, image data properties like gray level histograms, image entropy, and various correlation functions are determined. The second part yields an appropriate compression technique design with respect to the measured image properties.
The most commonly used structure in image compression systems is illustrated in Figure 1.2. The compression stage compresses the original image to the form of bit stream. The decompression stage reconstructs the image after transmission (or) archiving. The first step removes information redundancy caused by the presence of high correlation in the image (i). Transform compression, predictive compression or any hybrid approach shall be used in this step. \( t = T(i) \) is the transformation function and \( i' = T^{-1}(t') \) is the reverse transform applied at the decompression stage.

The second step is to represent the transform coefficients approximately using a sequence of quantization indices. The function \( q = Q(t) \) generates the quantization indices. The number of quantization indices (q) is smaller than the number of transform coefficients (t). Thus the Quantization mapping (Q) introduces distortion and the decompression stage applies an approximate inverse function, \( t' = Q^{-1}(q') \). Finally the

---

**Figure 1.2 Basic Structure of an Image Compression System**
function \( c = C(q) \) encodes the quantization indices to form the final bit-stream, \( c \). This step is invertible and introduces no distortion so that the decompress function may recover the quantization indices as \( q' = C^{-1}(c') \).

### 1.5.6 Performance Measures

The efficacy of the image compression algorithm can be evaluated by computing the amount of compression achieved and measuring the visual quality of the reconstructed image.

#### 1.5.6.1 Compression Measurement

The purpose of image compression is to represent the image with a reduced string of bits called the compressed bit-stream, denoted by \( c \). If \( M \times N \) is the size of the original image \( (I) \) and \( B \) is the number of bits per pixel in the original image then the compression ratio is defined as,

\[
\text{compression Ratio (CR)} = \frac{M \times N \times B}{\|c\|}
\]  

(Equivalently, the compressed bit-rate, expressed in bpp (bits per pixel), is defined as,

\[
\text{Bit Rate} = \frac{\|c\|}{M \times N}
\]  

(1.2)

Also, the percentage of storage space saving is defined as,

\[
\text{Space Saving} (\%) = (1 - (1/\text{CR})) \times 100
\]  

(1.3)

#### 1.5.6.2 Visual Quality Measurement

The amount of information loss is expressed as a function of the original image \( (I) \) and the compressed bit-stream \( (c) \) and subsequently the reconstructed image \( (I') \). Mean Squared Error (\( MSE \)) and the Peak Signal to
Noise Ratio \((PSNR)\) are the most commonly used visual quality measures for image compression.

The \(MSE\) is the cumulative squared error between the reconstructed and the original image defined by,

\[
MSE = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - I'(x,y))^2
\]  

\((1.4)\)

\(PSNR\) is a measure of the peak error, defined by

\[
PSNR = 10 \log_{10} ((2^B - 1)^2 / MSE)
\]  

\((1.5)\)

where \(M, N\) are the dimensions of the image. A lower value for \(MSE\) means lesser error, and as seen from the inverse relation between \(MSE\) and \(PSNR\), this translates to a high value of \(PSNR\). The \(PSNR\) value is expressed in \(db\) (decibels). Good reconstructed images typically have \(PSNR\) values of 30 db or more [3].

This research work uses standard test images, seismic images, texture images and geometric images for benchmarking the performance of all algorithms. Because of its wide applications, data compression is of great importance in digital image processing.

1.6 OVERVIEW AND CONTRIBUTION OF THE THESIS

This thesis aims at developing lossy still image compression framework using bandeletization of images and neuro-statistical modeling. Structurally, this thesis is organized into seven chapters. The first three chapters provide the background information necessary to place this work in context and facilitate the understanding of the research results presented herein. The
remaining four chapters present a mix of research results and additional concepts required for the comprehension of these results.

The present chapter begins by presenting the terminology and basic concepts essential to the understanding of this thesis. Compression benefits and the need for image compression are also discussed briefly. Following this, a brief introduction to image compression is given.

Chapter 2 is a brief survey of the fundamental techniques for image compression. It discusses the various compression methods and the currently available image compression standards. Knowing the current scenario, need for this research work is also stressed.

Chapter 3 presents a review of lossy image compression algorithms in chronological order, collected from the literature.

The initial part of chapter 4 discusses the bandeletization procedure for image compression. The idea of using bandeletization is to give room for applying the redundancy removal techniques thereby increasing the compression ratio. The performance of this scheme is compared with the existing for image compression methods. The results state that compression in this sparser domain increases compression ratio. The latter part deals with the implementation of this concept in the wavelet domain resulting in higher compression ratio.

Chapter 5 examines the existing vector quantization techniques for image compression in the spatial domain. Next, the proposed work using statistical modeling through polynomial regression for quantization in the spatial domain is presented. The observations reveal that this new approach promises superior reconstructed image quality. Further, this work is
examined in the context of a wavelet transform domain. This transform based work increases the compression ratio. The performance of this work is compared with the existing frameworks in the same transform domain.

Chapter 6 integrates and extends ideas from previously proposed schemes in order to construct a single unified framework known as the Structured Reversible Neuro-Statistical Sparse Transform framework. This new framework is then used to study the relationships between the various other frameworks. The resulting framework improves the quality of the reconstructed image and increases the amount of compression.

Finally, Chapter 7 summarizes the results reported in this thesis along with the contributions that it makes. This chapter concludes by suggesting directions for future research.
2. EXPLORING IMAGE COMPRESSION

2.1 INTRODUCTION
In order to facilitate the understanding of the research results presented in this thesis, some of the elementary concepts relevant to this work are explained in this chapter. In particular, this chapter reviews the mathematical preliminaries and the fundamental techniques for gray scale still image compression. This chapter concentrates on the practical problems in using these techniques for image compression and suggests improvements, when they are adopted to be used on images. A discussion on image compression standards is also given. The discussion ends with a brief outlook of the proposed work.

Interest in image Compression dates back more than 30 years. The initial focus of research efforts in this field was the development of analog method for reducing video transmission bandwidth, a process called bandwidth compression. The advantage of digital computer and the subsequent development of advanced integrated circuits caused the shifting from analog to digital compression approaches. Over the years, the need for image compression has grown steadily and it plays a crucial role in many important and diverse applications including video conferencing, remote sensing etc.

2.2 MATHEMATICAL PRELIMINARIES
Image compression aims at reducing the amount of data used to represent an image. Information theory provides the mathematical framework to check whether there is a minimum amount of data that is sufficient to describe
completely an image without loss of information. Information is defined as knowledge, fact and news. It can be measured quantitatively. The carriers of information are symbols [4].

2.2.1 Entropy
The amount of information contained in a symbol or set of symbols is called entropy.

2.2.1.1 Symbol Entropy
The Entropy \( I \) of symbol with an occurrence probability \( p \) is the amount of information contained in the symbol. 
\[
I = \log_2 \left( \frac{1}{p} \right) = - \log_2 p
\]  
(2.1)

2.2.1.2 Source Entropy
Let, \( s_i, i = 1,2, \ldots m \) be the set of symbols generated by a discrete memoryless source, \( S = \{s_1,s_2, \ldots s_m\} \); \( p_i, i = 1,2, \ldots m \) be the corresponding occurrence probabilities; The Source Entropy \( H \) is defined as the average information content per symbol of the source.
\[
H = - \sum_{i=1}^{m} p_i \log_2 p_i
\]  
(2.2)

2.2.2 Discrete Memoryless Source
A source generates and sends out symbols or events, one at a time. By discreteness, it means that the source is a countable set of symbols. By memoryless, it means that the occurrence of a symbol in the set is independent of its preceding symbol. For example, a die generates a symbol between 1 and 6 every time it is thrown and the probability of a number appearing does not depend upon the numbers that have come up before.
2.2.3 Source Encoding Descriptions

The commonly used source encoding terms are briefly explained here.

**Source Encoding:** Assigning a code word to each symbol in the source

**Code Length:** The total number of bits in the code word

**Bit Rate:** Average length of code words usually expressed in bits per symbol

**Block Code:** Maps each source symbol from the information source into a fixed code word in the codebook

**Uniquely Decodable Code:** Unambiguously decodable code

**Instantaneous Code:** Decoding each code word in a codebook without knowing the succeeding code words

**Compact Code:** A Uniquely decodable code whose average length is the minimum among all other uniquely decodable codes based on the same source S and codebook C; Also referred to as a minimum redundancy code or an optimum code

**Variable Length Code:** Let, \( S, i = \{1,2, \ldots, m\} \) be the set of symbols generated by a discrete memoryless source, \( S \). Codebook comprises of all code words, \( C = \{c_1, c_2, \ldots, c_m\} \). A variable length code assigns a distinct code word, \( c_i, i = \{1,2, \ldots, m\} \) to each symbol, where \( c_i \) is the string of bits. The sequences of outcomes \( S_i \) from the source are represented by concatenated code words, \( c_{ij} \). The length of the code word is denoted as \( I_i \).
2.2.4 Source Encoding Theorem
This section discusses various theorems related to source encoding.

2.2.4.1 Shannon’s Noiseless Source Encoding Theorem
Shannon’s noiseless source encoding theorem states that for a discrete, memoryless, stationary information source, the average number of bits per symbol used by the code must be at least equal to the entropy of the source. This theorem provides a lower bound, i.e., entropy in source encoding. The efficiency $\eta'$ of an encoding scheme is defined with respect to the lower bound as follows:

$$\eta' = n \frac{H}{L_{\text{avg}}}$$  \hspace{1cm} (2.3)

where, $H$ is the source entropy, $n$ is the number of symbols and $L_{\text{avg}}$ denotes the average length of the code words in the code. As the entropy is the lower bound, the efficiency never exceeds the unity, i.e. $\eta' \leq 1$ [4, 5].

2.2.4.2 Shannon’s Noisy Channel Encoding Theorem
Shannon’s noisy channel encoding theorem states that it is possible to transmit symbols over a noisy channel without error if the bit rate ($R$) is below the channel capacity ($C$) (i.e.) $R < C$. To be error-free, the bit rate cannot exceed the channel capacity. That is, the channel capacity sets an upper bound on the bit rate. This theorem is concerned with the memoryless noisy channel. Shannon’s Noiseless Source Encoding Theorem and Shannon’s Noisy Channel Encoding Theorem work for lossless compression.

2.2.4.3 Shannon’s Source Encoding Theorem
Shannon’s source encoding theorem states that for a given distortion $D$, there exists a rate-distortion function $R(D)$, which is the minimum bit rate
required to transmit the source with distortion less than or equal to $R$. (i.e.)
$R \geq R(D)$; where $R$ is the bit rate. This theorem works for lossy compression.

2.2.5 Rate-Distortion Theory
There is a trade-off between minimizing the bit rate and keeping the
distortion small. If bit rate were the only criterion for lossy compression
techniques, where loss of information is permitted, the best lossy
compression scheme would be simply to throw away all the data. Therefore,
distortion measure should also be considered in addition to the bit rate
measure. Best lossy compression schemes incur minimum amount of
distortion while compressing to the lowest bit rate possible.

2.2.6 Information Transmission Theorem
The information transmission theorem combines the noisy channel encoding
theorem and the source encoding theorem. It states that if the channel
capacity $C$ of a noisy channel is larger than the rate-distortion function $R(D)$
then it is possible to transmit an information source with distortion over a
noisy channel.

2.3 IMAGE COMPRESSION TECHNIQUES
This section discusses the traditional image compression techniques for gray
scale images. Compression techniques can be categorized either based on the
quantity of information preserved in the reconstructed image or based on
the type of redundancy that is present in the original uncompressed image.
Lossless and Lossy compression are the two categories based on information
preservation.
Statistical redundancy removal, redundancy removal using predictability and image transforms are the three techniques based on redundant image content.

- **Lossless Compression**
Lossless techniques guarantee that the output generated is exactly the duplicate of the input after a compression followed by an expansion cycle. Hence, the quality of the output is retained at the cost of compression ratio. Generally these methods are used in applications where the loss of even a single bit is dangerous. Such methods are used in image archiving. Most of the graphics compression techniques have concentrated on using conventional lossless techniques.

- **Lossy Compression**
In the lossy compression, the changes (or) loss of data in the output obtained after a compression followed by an expansion cycle is allowed but without affecting perceived quality on the part of the user. Hence, the quality of the output degrades, but the compression would be better, since the intermediate data is sampled to achieve better compression. Applications of this type are in broadcast television, video conferencing etc in which certain amount of error is acceptable as a trade-off for increased compression performance.

**2.3.1 Statistical Redundancy Removal Techniques (Lossless)**
This section discusses the image compression techniques that have gained wide acceptance. Image compression is achieved when one or more of the basic redundancies are reduced or exploited. Since, statistical redundancy is
directly related to the image data probability distribution it can be treated by information theory techniques using entropy concepts. Its removal will result in lossless image compression techniques that are used in facsimile transmission.

### 2.3.1.1 Coding Redundancy Removal Techniques

Coding redundancy removal techniques assign smaller code words to the more probable symbol occurrences and larger code words to the least probable symbols. Hence, these techniques are also called variable length coding techniques. Huffman and arithmetic coding techniques are discussed here.

#### 2.3.1.1.1 Huffman Coding

Huffman code [6] is the first optimum code and is the technique most frequently used at present. Huffman codes are generated using the following rules:

1. \( l_1 \leq l_2 \leq \ldots \leq l_{m-1} \leq l_m \), where \( l_1, l_2, \ldots, l_m \) are the code word lengths;
   
   This rule implies that when the source symbol occurrence probabilities are in a non-increasing order, the length of the corresponding code words should be in a non-decreasing order.

2. The code words of the least probable source symbols should be the same except for the most significant bit.

3. Each possible sequence of length \( l_{m-1} \) bits must be used either as a code word or must have one of its prefixes used as a code word.
Huffman Algorithm

The Huffman coding algorithm consists of the following steps:

4. Sort out all source symbols in a non-increasing order of their occurrence probabilities
5. Assign a binary 0 and a binary 1 to the two least probable symbols
6. Merge the two least probable source symbols
7. Generate a new source symbol with a probability equal to the sum of the probabilities of the two least probable symbols
8. Add a binary 0 and a binary 1 to the two least probable symbols
9. Repeat steps 3-5 until the newly generated supplementary source symbol contains only one source symbol
10. Start from the source symbol in the last supplementary source symbol and trace back to each source symbol in the original source symbol to find the corresponding code words

2.3.1.1.2 Arithmetic Coding

Both encoding and decoding involve only arithmetic operations (addition and multiplication in encoding, subtraction and division in decoding), and hence the name arithmetic coding. Arithmetic coding completely bypasses the idea of replacing an input symbol with a specific code. Instead, it represents the entire input file as a small interval between the range 0 and 1.

With \( n \) bits at most \( 2^n \) different combinations can be represented; or with \( n \) bits a code interval between zero to one can be divided into \( 2^n \) parts each having the length of \( 2^{-n} \). Let, \( A = 2^{-n} \); Then an interval with the length \( A \) can be coded by using \( N = -\log_2 A \) bits.
Arithmetic coding starts by dividing the interval into subintervals according to the probability distribution of the source. The length of each subinterval equals to the probability of the corresponding symbol. Thus, the sum of the lengths of all subintervals equals to 1 filling the range 0 to 1 completely. The coding proceeds by taking the subinterval of the first symbol. Then this interval is again split into subintervals so that the length of each subinterval is relative to its probability. The process is repeated for each symbol to be coded resulting to a smaller and smaller interval. The final interval describes the source uniquely. The length of this interval is the cumulative multiplication of the probabilities of the coded symbols:

\[
A_{\text{final}} = p_1 \cdot p_2 \cdot \cdots \cdot p_n = \prod_{i=1}^{n} p_i
\]  

(2.4)

where, \( p = \{p_1, p_2, \cdots, p_n\} \) is the probability vector for the source symbol set \( S = \{s_1, s_2, \cdots, s_n\} \).

Therefore, this interval can be coded by number of bits \( n \) (assuming \( A \) is a power of 1/2),

\[
n = - \log_2 p_i = - \sum C(A) = - \log_2 \prod_{i=1}^{n} p_i = - \sum_{i=1}^{n} \log_2 p_i
\]

(2.5)

If the same model is applied for each symbol to be coded, the code length can be described in respect to the source alphabet

\[
C(A) = - \sum_{i=1}^{n} p_i \log_2 p_i
\]

(2.6)

where, \( n \) is the number of symbols in the alphabet, and \( p_i \) is the probability of that particular symbol in the alphabet. The important observation is that the \( C(A) \) equals to the entropy. This means that the source can be coded optimally if \( A \) is a power of 1/2. This is the case if the length of the source
approaches to infinite. In practice, arithmetic coding is optimal even for rather short sources.

2.3.1.2 Interpixel Redundancy Removal Techniques

Interpixel redundancy removal techniques assign fixed-length code words to variable length sequences of source symbols but require no prior knowledge of the probability of occurrence of the symbols to be encoded. Hence, these techniques are also called fixed length coding techniques. LZW coding is an example for fixed length coding.

2.3.1.2.1 LZW Coding

The Lempel-Ziv and Welch (LZW) algorithm [7] was proposed by Welch as a general purpose coding scheme. The image to be encoded is treated as a one-dimensional bit string. The coded image is also treated as a bit string. The algorithm produces the output string and updates the code table simultaneously. Thus, the codebook can adapt to the particular characteristics of the image to be compressed. The code table can have up to 4096 code words and thus each code word can be up to 12 bits long. The structure of the LZW codebook is shown in Figure 2.1. The first 256 code words are the numbers 0.....255. The code index 256 indicates a special clear code. The code index 257 indicates an End Of Information (EOI) code. The code indices from 258 to 4095 indicate bit strings that are frequently encountered in the image. The first 512 code words are represented by 9 bits each. The code words from 512 to 1023 are 10 bits long.
Similarly, the code words from 1024 to 2047 are 11 bits long and the code words from 2048 to 4095 are 12 bits long [8]. LZW coding is also called dictionary-based coding because this scheme reads an input data and looks for a matching string in a dictionary. If a string match is found, the pointer or index of the string is taken as the output. Dictionary is used like the list of references. It is static because it is built up and transmitted with the text of work. The problem here is that the dictionary needs to be transmitted along with the text, resulting in a certain amount of overhead added to the compressed text.

2.3.1.2.2 Bit Plane Coding
This technique reduces inter pixel redundancy by processing the image’s bit planes individually. It is based on the concept of decomposing a multilevel
image into a series of binary images and compressing each binary image via a binary data compression method.

The gray levels of an m-bit gray scale image can be represented in the form of the base 2 polynomial,

\[ a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \ldots + a_02^0 \]  

(2.7)

Based on this property, a simple method of decomposing the image into a collection of binary images is to separate the m coefficients of the polynomial into m 1-bit planes. The zeroth-order bit plane is generated by collecting \( a_0 \) bits of each pixel, while the \((m - 1)^{th}\) order bit plane contains the \( a_{m-1} \) bits. The higher order bits (especially the top four) contain the majority of the visually significant data. The other bit planes contribute to more subtle details in the image. The insignificant bits are assigned a binary 0 and results inter-pixel redundancy. This redundancy may be eliminated by slicing those insignificant bits.

2.3.1.2.3 Run-Length Coding

The term ‘run’ is used to indicate the repetition of a symbol, while the term ‘run-length’ is used to represent the number of repeated symbols, i.e., the number of consecutive symbols of the same value. Let \( S = \{s_1, s_2, \ldots, s_m\} \) be the set of source symbols; Assume this set is divided into k-segments having length \( l_i \) and gray level \( g_i, 1 < i < k \). The set can be represented using the following mapping:

\[ s_1, s_2, \ldots, s_m \rightarrow (g_1, l_1), (g_2, l_2) \ldots, (g_k, l_k) \]  

(2.8)

where, \( g_1 = s_1 \) and \( g_k = s_m \). Each couple \((g_i, l_i)\) is called a gray-level run.
This gray-level run representation results in considerable compression if the gray-level runs are relatively large. To determine the run-length, the image is processed row by row, from left to right. Then, the subsequent pixels having same gray-level are blocked in each scanning line. Each block, referred as run, is then coded by its gray-level information and its length resulting to a code stream like \((g_1, l_1), (g_2, l_2), \ldots, (g_k, l_k)\).

In binary images there are only two colors, thus a black run is always followed by a white run, and vice versa. Therefore it is sufficient to code only the lengths of the runs; no color information is needed. The first run in each line is assumed to be white. On the other hand, if the first pixel happens to be black, a white run of zero length is coded. The run-length coding method is purely a modeling scheme resulting to a new alphabet consisting of the lengths of the runs. These can be coded for example by using the Huffman code.

2.3.1.2.4 Predictive Run-Length Coding

The performance of the run-length coding can be improved by using prediction technique as a preprocessing stage. The idea is to form an error image from the original one by comparing the value of each original pixel to the value given by a prediction function. If these two are equal, the pixel of the error image is white; otherwise it is black. The run-length coding is then applied to the error image instead of the original one. Benefit is gained from the increased number of white pixels; thus longer white runs will be obtained.

The prediction is based on the values of certain (fixed) neighboring pixels. These pixels have already been encoded and are therefore known to
the decoder. The prediction is thus identical in the encoding and decoding phases. The image is scanned in row-major order and the value of each pixel is predicted from the particular observed combination of the neighboring pixels.

2.3.2 Redundancy Reduction Techniques using Predictability (Lossy)
This section presents the psycho-visual redundancy reduction techniques and the predictive coding technique to reduce neighborhood pixel redundancy using predictability.

2.3.2.1 Psycho-Visual Redundancy Reduction Techniques
Psycho-Visual Redundant information present in an image is reduced through the quantization process. Quantization [9] refers to the process that converts continuous input data into a set of discrete values. Figure 2.2 depicts the basic quantization process. Hence, quantization is essentially, discretization in magnitude, which is an important step in the lossy compression of digital image and video. The quantization process is irreversible and introduces distortion that is directly proportional to the square of the step size.

![Figure 2.2 The Quantization Process](image-url)

Figure 2.2 The Quantization Process
One way of measuring the distortion is to compute the MSE. The commonly used quantization techniques are the scalar quantization and the Vector Quantization (VQ).

2.3.2.1.1 Scalar Quantization

Scalar Quantization allows individual coefficients to be converted to a quantized value with the conversion being independent from coefficient to coefficient. There are two kinds of scalar quantization techniques namely the uniform and the non-uniform quantization techniques. Uniform quantization, which is the simplest yet most important case, is discussed first. Then, non-uniform quantization is covered. Finally, the well known Lloyd-Max Quantizer design is presented.

Uniform Scalar Quantization

In uniform quantization, the quantization level is an important parameter as it determines the value of the step size. All intervals are of the same size, except possibly for the two outer intervals. A large quantization level will result in a small step size while a small quantization level will result in a large step size. For a given quantization level, the step size is constant and controls the quality of an image. A large step size gives a poorer approximation of the image while a small step size will give a better one.

The equation for computing the step size is:

\[ \Delta = \frac{(m_k - m_0)}{L} \]  

(2.9)

where, \( \Delta \) – the step size; \( m_k \) – maximum value;
\( m_0 \) – minimum value; \( L \) – the number of quantization levels;
Figure 2.3 illustrates this concept. The input data is taken along the horizontal axis and the quantized output is taken along the vertical axis. The quantizer divides the range of input values into non-overlapping intervals bounded by amplitudes $d_k$, known as the decision levels. The intervals between these amplitudes $d_k$ are the step sizes, $\Delta$.

If an input coefficient falls in the interval $(d_k, (d_{k+1})$, it will be specified by the index $(k)$, which is then transmitted to the destination. At the destination, the received indices are transformed to amplitudes $r_k$ known as the reconstruction levels.

**Non-Uniform Scalar Quantization**

A quantizer that has non-uniform intervals is called non-uniform quantizer. Non-uniform quantization is suitable for quantizing non-uniformly distributed input. In regions with more input distribution smaller step sizes (intervals) are allowed in order to reduce the error in this region. In regions with less distribution larger step sizes are allowed resulting in larger errors.
Assuming more input distribution near the origin, the quantization levels closer to the origin are of smaller intervals. Hence the maximum value that the quantizer error can take on in these intervals is also smaller, resulting in a better approximation. But in regions with lower input distribution larger intervals are allowed resulting in larger error in these intervals. However as the probability of getting input values corresponding to the lower distribution region is smaller than those in the larger distribution region.

**The Lloyd-Max Scalar Quantizer**

The design procedure for the well known scalar quantizer introduced by Lloyd is given below:

1. Initialize the code vectors in the codebook $C = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{N-1}\}$
2. Find decision boundaries using the formula,
   \[
   t_q = \frac{\tilde{x}_{q-1} + \tilde{x}_q}{2}; \quad q = 1, 2, \ldots, N - 1
   \]
   \[\text{Eq. (2.10)}\]
3. Compute the new code vector
   \[
   \tilde{x}_q = \frac{\int_{t_q}^{t_{q+1}} \tilde{x} f(x) \, dx}{\int_{t_q}^{t_{q+1}} f(x) \, dx}
   \]
   \[\text{Eq. (2.11)}\]
4. Calculate the distortion
   \[
   d_j = \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} (x - \tilde{x})^2 f_x(x) \, dx
   \]
   \[\text{Eq. (2.12)}\]
   where $f_x(x)$ is the probability density function of $x$
5. If \( (d_{j-1} - d_j)/d_j < \tau \), then stop;
6. Otherwise set $j = j + 1$ and go to step 2; $\tau$ is the threshold for minimum distortion.
2.3.2.1.2 Vector Quantization

Vector Quantization is a generalization of scalar quantization. VQ is always better than scalar quantization because it fully exploits the correlation between components within the vector. Also, its decoding procedure is very simple since it only consists of table lookups. However, like all things in life, quality comes with a cost. For VQ that cost comes in the form of codebook storage space and design complexity requirements. The size of the codebook increases with increasing vector dimension and increasing bit rate. Therefore, careful attention should be given to the choice of dimension and rate. In practice, VQ having a dimension of 16 or less is preferred because complexity, storage space and performance tradeoffs are most attractive in this range.

**Principle of VQ**

Assume \( \{l_v\} \) be the \( k \)-dimensional input vector set. The function of an \( n \)-level vector quantizer, \( V_q \) is a mapping from \( \{l_v\} \) into a finite codebook, \( V_q: l_v \rightarrow C; \) where \( C = \{c_1, c_2, \ldots, c_n\} \) (2.13)

In other words, for an input vector \( \{l_v\} \), VQ assigns a representative code vector \( c \) from a codebook. The representative code vector is determined to be the closest in Euclidean distance from the input vector. The Euclidean distance is defined by:

\[
d(x, c_i) = \sqrt{\sum_{j=1}^{k} (x_j - c_{ij})^2}
\] (2.14)

Where, \( x_j \) is the \( j^{th} \) component of the input vector, and \( c_{ij} \) is \( j^{th} \) the component of the code vector \( c_i \).
Figure 2.4 Codebook Generation

![Codebook Generation Diagram]

Figure 2.5 An Example Codebook

<table>
<thead>
<tr>
<th>Codebook</th>
<th>code index</th>
<th>codevector</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000</td>
<td>10 12 13 14</td>
</tr>
<tr>
<td></td>
<td>0001</td>
<td>63 65 66 67</td>
</tr>
<tr>
<td></td>
<td>0010</td>
<td>35 36 38 39</td>
</tr>
<tr>
<td></td>
<td>.....</td>
<td>.....</td>
</tr>
<tr>
<td></td>
<td>1111</td>
<td>85 87 89 88</td>
</tr>
</tbody>
</table>

Figure 2.6 VQ Encoder
Encoding and decoding are viewed as the two distinct operations of the VQ process.

**VQ Encoder**

The first step of VQ encoding is codebook generation as shown in Figure 2.4. In this process the input image is decomposed into a set of vectors. Then, a representative set of the expected input vector source is chosen from the input image. This is called training set generation. These training vectors are stored in a digital memory in the form of a table and are indexed. This table is called codebook and each entry in the codebook is called code vector. An example codebook is shown in Figure 2.5. In the next step a best-matching vector decision is made from among the code vectors. This process is shown in Figure 2.6.

The encoder uses Euclidean distance which is the most common distortion measure to select a best-matching code vector. The index vector consisting of the indices of the best-matching code vector for each input vector, is the quantized output of the VQ process.

**VQ Decoder**

The operation associated with the decoder is extremely simple. No arithmetic computation is needed. The code vector corresponding to the input index is retrieved from the codebook using table lookup procedure. These code vectors are used to reconstruct the input image. This is shown in Figure 2.7.
VQ Implementation
VQ implementation involves VQ design procedures and VQ search procedures. The following section discusses these two procedures in detail.

VQ Design Procedure
The key element in designing a VQ is generating a good codebook for a given input image. Designing a codebook that best represents the set of input vectors is NP-hard. That means that it requires a full search for finding the best matching code word. Also the search increases exponentially as the number of code words increases. Linde, Buzo and Gray (LBG) design algorithm is the most commonly used VQ codebook design algorithm. LBG algorithm is also called the Generalized Lloyd Algorithm (GLA) as it is a generalization of the scalar quantization design algorithm introduced by Lloyd. The algorithm is as follows:
1. Decompose the input image into set of vectors.
2. Initialize the size of the codebook.
3. Randomly select N vectors from the input vector set, and let those be the initial code vectors of the codebook.

4. Compute the Euclidean distance between each code vector and input vector. Cluster the input vector to the code vector that yields the minimum distance.

5. Compute the average of each cluster. Let this value be the new code vector.

6. Repeat steps 4 and 5 until there are negligible changes in the code vectors obtained in the subsequent steps.

Although this algorithm is very simple to implement it requires more computational cost. Because, cluster generation in the iteration needs that each input vector be compared with all the code vectors in the codebook.

**VQ Search Procedures**

The full search method and the binary search method are the commonly used search methods to find the best matching code vector from the codebook for each input vector. The simplest search method, which is also the slowest, is full search as shown in Figure 2.8. It is an exhaustive search process over the entire codebook for finding the closest code vector. Assume \( I_v \) be the number of input vectors, \( k \) be the input vector size and \( C = \{c_1, c_2, ..., c_n\} \) be the codebook.

The quantization process requires the number of arithmetic operations:

- No. of multiplications \( kI_vC \)
- No. of additions and subtractions \( I_vC((k - 1) + k) = I_vC(2k - 1) \)
- No. of comparisons \( I_vk(C - 1) \)

This makes the full search method an expensive method.
Figure 2.8  Full Search Method

Figure 2.9  Binary Search Method
Another alternative approach is a tree search method where the search is carried out in a hierarchical pattern. But it requires more memory for the codebook. Binary trees are preferred as they are the most efficient in terms of complexity. Figure 2.9 shows the structure of a binary tree for a codebook of size $C = 8$. The leaf nodes in the binary tree represent the code vectors $\{c_1, c_2, \ldots, c_n\}$. There are 8 search paths as $C = 8$. The search process encounters $\log_2 C$ decision points, one at each level.

2.3.2.2 Predictive Coding Technique

Predictive coding technique aims at reducing neighborhood pixel redundancy. The system employs a predictor, a quantizer and a symbol encoder. The predictor uses the correlation of pixels to construct an estimate of the intensity of an image pixel from intensities in its neighborhood pixels. The prediction error is obtained from the difference between the quantizer output and estimate of that pixel. The prediction error is then coded using a variable length coding scheme.

2.3.3 Image Transforms

This section deals with different types of image transforms and their properties. The spatial representation is not always the best representation of the image for most image processing applications. In many cases, the distinguished information is hidden in the frequency content of the signal. Transform techniques migrate image representation from one domain to another domain. Transforms convert the statistically dependent image elements to independent coefficients by decomposing the input image into its components with individual characteristics. Transforms [10] are
extensively used in image filtering, image compression and feature extraction. A brief description about few of those transforms is given below.

2.3.3.1 Fourier Transform
The Fourier transform may be represented using a set of orthogonal sinusoidal waveforms; the coefficients of the Fourier representation are called frequency components and the waveforms are ordered by frequency. The discrete Fourier transform is analogous to the continuous one and may be efficiently computed using the fast Fourier algorithm. The Fourier transform is of great use in the Hadamard transform calculation of image convolutions.

2.3.3.2 Hadamard Transform
The base of the Hadamard transform consists of square waves (Walsh functions). The Walsh functions are ordered by the number of their Zero-crossings and the coefficients are called sequence components.

2.3.3.3 Discrete Cosine Transform
There are four definitions of the Discrete Cosine Transform (DCT), denoted DCT-I, DCT-II, DCT-III, and DCT-IV. The most commonly used discrete cosine transform in image processing and compression is DCT-II, which forms the basis for JPEG image compression.

2.3.3.4 Wavelet Transform
Another variation of the transform coding is the Wavelet Transform. The Wavelet is a representation of image data. The wavelet analysis procedure is implemented with dilated and translated versions of a mother wavelet. Since signals of interest can usually be expressed using wavelet decompositions,
signal processing algorithms can be performed by adjusting only the corresponding wavelet coefficients.

In theory, the dilation (scale) parameter of a wavelet can be any positive real value and the translation (shift) can be an arbitrary real number. This is referred to as the continuous wavelet transform. In practice, however, in order to improve computation efficiency, the values of the shift and scale parameters are often limited to some discrete lattices. This is then referred to as the discrete wavelet transform.

Wavelet compression consists of the same steps as DCT compression, but the DCT is replaced by a wavelet transform followed by generally identical quantization and coding.

2.3.3.5 Karhunen-Loeve Transform
The Karhunen-Loeve transform matrices consist of eigenvectors of the covariance matrix of the original image, or a class of original images. The power of this transform is in the search for non-correlated transform coefficients, which makes it very useful in data compression.

2.3.3.6 Other Discrete Image Transforms
Many other discrete image transforms exist, e.g. Haar, Hadamard – Haar, Slant, Slant-Haar, Radon, discrete sine transform, etc.,

2.3.3.7 Fractal Compression
Fractal Compression and decompression involves a clustering approach to find regions, which show the same characteristics as a sample region independent to rotation and scale. This technique compresses images as recursive equations and instructions about how to reproduce them. Fractal
image compression offers extremely high compression ratios and high quality image reconstruction. Breeding an image into pieces (fractals) and identifying self-similar ones is the main principle of this approach.

2.3.4 Other Techniques

Various other image data compression methods exist. Image representation by its low and high frequencies is another method. A significantly smaller volume of data than the original data can represent the low-frequency image. The high-frequency image has significant image edges and can be represented efficiently.

A technique that is gaining popularity is block truncation coding in which an image is divided into small square block of pixels and each pixel value in a block in truncated to one bit by threshold and moment preserving selection of binary levels. One bit value per pixel has to be transmitted, together with information describing how to recreate the moment preserving binary levels during reconstruction.

2.4 IMAGE COMPRESSION SCHEMES USING VQ

Since this thesis work contributes much to the transform based VQ compression, VQ techniques are discussed in detail in this section. This section presents few of the image compression schemes using VQ reported in the literature that include Gain-Shape VQ, Mean-Removed VQ, Classified VQ, Trellis-Coded VQ, Multistage VQ and Lattice VQ.
2.4.1 Gain-Shape Vector Quantization

A functional block diagram of Gain-Shape VQ encoder is shown in Figure 2.10. Initially the encoder decomposes the input vector into two components, the gain and the shape components. Eqn. 2.15 and Eqn. 2.16 are used for this decomposition.

The gain component is a scalar and is obtained using the Euclidean norm:

\[ g = \| I \| = \sqrt{\sum_{j=0}^{N-1} x^2(j)} \]  \hspace{1cm} (2.14)

where, \( I \) – input vector; \( N \) – number of elements in \( I \);

The shape factor is a gain normalized vector component given by,

\[ S = \frac{I}{g} \]  \hspace{1cm} (2.16)

Then, the gain component is quantized with a conventional scalar quantizer while the shape components are quantized with a vector quantizer.

![Figure 2.10 Gain-Shape VQ Encoder](image-url)
Mean-Removed VQ is similar to the Gain-Shape VQ but involves extracting a mean term as the scalar component instead of a gain term. First the mean of the input vector is computed and quantized with a scalar quantizer. The mean value is subtracted from the respective input vector resulting in a mean-removed input vector. This resultant vector is subject to vector quantization. At the decoder, the mean-removed vectors are obtained from a codebook lookup procedure. The mean-removed vector is added to a set of values obtained by the scalar index re-assignment process in order to restore the mean to it. The functional block diagram of the VQ encoder is shown in Figure 2.11.

2.4.3 Classified Vector Quantization

The classified VQ process is shown in Figure 2.12. This approach divides the input image vector into separate classes with different properties. For
example, edge vectors and non-edge vectors form two different classes. The edge vectors can be further divided into many types according to their location and angular orientations. Separate vector quantizers are designed for the different classes. During the encoding process, this method assigns the corresponding codebook to the classified vector. The encoder transmits both the label for the codebook used and the label for the vector in the codebook.

2.4.4 Trellis-Coded Vector Quantization

Trellis-Coded Vector Quantization is a special case of Trellis-Coding. A Trellis is a state transition diagram for a finite state machine. It is used to study the sequences of state transitions. This scheme originates from communication theory and it employs ideas from Trellis-Coded Modulation. The algorithm initializes a codebook with $2^{R+1}$ code vectors; where $R$ is the number of bits per sample used by the quantizer. Then it quantizes each element of the input vector using $2^R$ code vectors selected from the codebook. This selection is based on the previous code vectors used to quantize the previous quantizer input.

![Figure 2.11 A Simple Classified VQ Encoder](image-url)

Figure 2.11 A Simple Classified VQ Encoder
Figure 2.12 Multi-Stage Vector Quantization
2.4.5 Multi-Stage Vector Quantization

Multi-Stage Vector Quantization (MSVQ) quantizes the input vector in several stages. The principle of operation is illustrated in Figure 2.133. The first stage quantizes the original input vector coarsely. The second stage accepts the error between the original input and the VQ output from the first stage and quantizes this error. This method of quantizing the error is repeated for several stages. The difference between the input and the respective VQ output vector is often called the residual error and hence MSVQ is also called the residual vector quantization. The final reconstructed vector is the sum of the VQ output vectors of each of the stages.

2.4.6 Lattice Vector Quantization

Lattice Vector Quantization aims at reducing the quantization error due to the choice of the size and shape of the quantization level. Lattice represents a systematic arrangement of points in a two-dimensional space. If the space coordinates are integers then the lattice is called integer lattice. The quantizer is built based on a fixed size lattice. Then the lattice point closest to the input vector is found by computing the sum of the coefficients of the input vector.

2.5 IMAGE COMPRESSION STANDARDS

There is an increasing effort to achieve standardization in image compression. Many of the lossy and lossless compression methods have played important roles in the development and adoption of current principal image compression standards. These standards in most cases have been developed and sanctioned under the joint agreement of the International
Standards Organization (ISO) and the Consultative Committee of the International Telephone and Telegraph (CCITT). The two organizations CCITT and ISO have developed the standards namely MH, MR, MMR, JBIG, and JPEG. These standardized algorithms for image compression can be defined, along with the parameter settings used to maximize their compression, as follows:

- **MH**: Modified Huffman Coding
  An algorithm defined in CCITT Recommendation T.4 using an 1-D Minimum length-coding model followed by static Huffman codes.

- **MR**: Modified READ (Relative Element Address Designate) coding
  An algorithm defined in CCITT recommendation T.4, which uses a 2-D, reference and run-length coding model followed by static Huffman codes.

- **JBIG**: Joint Bi-level Image Experts Group Coding
  An algorithm defined in CCITT Recommendation T.82, which for bi-level compression uses an adaptive 2-D coding model, followed by an adaptive arithmetic coder.

- **JPEG**: Joint Photographic (Image) Experts Group Coding
  An algorithm defined in CCITT Recommendation T.81, for lossless coding uses a static 3-D coding model, followed by a Huffman coder.

### JPEG Still Image Compression Standard

The Joint Photographic Experts Group has developed an international standard for general purpose, color, still image compression. Four different compression modes are defined by this standard. They are the Sequential DCT based compression, Progressive DCT based compression, Sequential lossless predictive Compression and Hierarchical lossy or lossless...
compression. The lossy compression modes were designed to achieve compression ratios around 15 with very good or excellent image quality, the quality deteriorates for higher compression ratios. A compression ratio between 2 and 3 is typically achieved in the lossless mode. The JPEG algorithm is based on the forward DCT as applied to a block breakdown of the original image into 8 x 8 blocks, quantized down to a finite set of possible values, and then entropy encoded using Huffman or arithmetic coding.

**MPEG - Full Motion Video Compression Standard**

As a logical extension of JPEG still image compression, the Motion Picture Experts Group standard was developed for full motion Video Image Sequences with applications to digital video distribution and High-Definition Television in mind. This Compression standard facilitates the following features of the compressed Video: random access, fast forward / reverse searches, reverse playback, audio - visual synchronization, robustness to error, edit-ability, format flexibility and cost trade-off.

**2.6 ADVANTAGES AND DISADVANTAGES OF THE EXISTING COMPRESSION METHODS**

Based on the survey, the advantages and disadvantages of the various image compression methods are analyzed and presented here.

Among the coding schemes, Huffman coding (which resembles almost Shannon-Fano coding in performance) provides optimal compression and error-free decompression, but it requires integral number of bits of each code. As a systematic procedure to encode a finite, discrete memoryless source, the Huffman code has found wide application in image and video
compression. Huffman coding always encodes a source symbol with an integer number of bits.

Arithmetic coding converts a source symbol string to a code symbol string. Although arithmetic coding is more powerful than Huffman coding in compression ratio, arithmetic coding requires more computational power and memory. Huffman coding is more attractive than arithmetic coding when simplicity is the major concern. Block truncation coding is fast and easy to implement, LZW coding improves the compression efficiency, but it requires the dictionary to be transmitted along with the compressed text.

Cosine and Sine transforms are fast transforms; both find the applications in image processing problems. Cosine transform has excellent energy compaction for images. Sinusoidal Transforms have fast implementation, useful in finding practical substitutes for the KL Transform and mathematical modeling of signals. Hadamard is faster than sinusoidal transforms, since no multiplication is required. It is useful in digital hardware implementation of image processing algorithms. It is easy to simulate but difficult to analyze. It has good energy compaction for images. It is useful in feature extraction image coding and image analysis problems. Energy compaction is fair. KL transform is optimal. It is useful in performance evaluation and for finding performance bounds.

Vector Quantization is an efficient approach to low bit-rate image compression. But it requires a complex coder. The quantization parameters are very sensitive to image data and they blur edge details present in images. Predictive Methods can achieve larger compression ratios in a much less expensive way.
Fractal image compression offers extremely high compression ratios and high quality image reconstruction. Fractal Compression is non symmetrical. Decompression takes significantly less time than compression. Decompression can also be significantly faster than JPEG. It can be used as a component in live Video Compression. But the Compression time is very long, unless hardware assisted. It is not suited for repeated compression and decompression cycles due to its lossy nature. JPEG is ISO supported. The compression rate is user selectable, but is very low (4 to 10). It results in fast degradation in image quality.

2.7 PROPOSED WORK
Abundant algorithms for achieving various image processing tasks are available in literature. Nevertheless, there does not appear to be any unifying principle guiding many of them. Some are one dimensional signal processing techniques, which have been extended to two dimensions. Others apply methods from different disciplines to image data in a somewhat improper manner. Many are basic algorithms with constraint values modified to suit the problem at hand. Alternatively, the constraints are optimized with respect to a suitable training set, without thought on how to vary them for images with different characteristics. There exist well considered methods, but unfortunately a large amount of new ideas have been ad-hoc, without any fundamental guiding principle.

This thesis proposes a unified approach to image compression. The image data in the spatial domain is converted to a different domain in order to make the subsequent quantization and joint entropy encoding more
efficient. Bandeletization concept is introduced because the geometry of the image is summarized with local clustering of similar geometric vectors and the homogeneous areas being taken from the quad tree structure using this process. This process helps to identify the areas in the image that are geometrically similar to each other. Artificial Neural Networks, including Competitive learning networks have also been studied widely and a number of systematic approaches have been proposed in the contributory chapters.

2.8 METHODOLOGY OF THE PROPOSED FRAMEWORK FOR IMAGE COMPRESSION

The methodology followed in the proposed research work is as follows:

a) Transform the black and white still images using Wavelet transformation technique
b) Apply Bandeletization process to the transformed coefficients
c) Design a codebook for the Vector Quantizer using a supervised learning neural network and model the code vectors using polynomial regression procedure
d) Quantize the bandeletized coefficients using the vector quantization technique that uses the codebook designed in step (c)
e) Encode the quantized output using Huffman encoder
f) Inspect the compression efficiency of the proposed algorithm using various test images
g) Compare its performance with the existing techniques to prove the effectiveness of the proposed algorithm.
3. REVIEW OF IMAGE COMPRESSION LITERATURE

3.1 INTRODUCTION
The literature abounds with algorithms for achieving various image processing tasks. This chapter presents a survey of gray scale still image compression works found from the literature. As this thesis presents Vector Quantization based Image compression, review related to vector quantization is elaborately discussed. The following sections discuss various compression algorithms and the research carried out by various researchers.

3.2 LBG VECTOR QUANTIZATION
Linde, Buzo and Gray algorithm [11] is the first practical codebook design technique for Vector Quantization. This scheme is a generalization of Lloyd's Pulse Code Modulation design technique [12] which was developed by Lloyd in 1957. Hence it is also called Generalized Lloyd's Algorithm. This algorithm requires an initial codebook. This initial codebook is obtained by the splitting method. In this method, an initial code vector is set as the average of the entire training sequence. This code vector is then split into two. The iterative algorithm is run with these two vectors as the initial codebook. The final two code vectors are split into four and the process is repeated until the desired number of code vectors is obtained. The drawback of this scheme is that the algorithm incurs computational overhead. In addition, a poorly selected initial codebook may result in an undesirable final codebook.

Maximum-Separation codebook initialization algorithm for use with k-means algorithm was presented by Easwaran and Gowdy [13]. This algorithm gives better performance when compared with codebook
initialization using randomly selected feature vectors and codebook initialization using the first few feature vectors of the input data. An image coding scheme that combines wavelet transform with LBG VQ was presented by Amir et. al. [14]. Wavelet transform decomposes an image into sub-images with different resolutions. These sub-images correspond to different frequency bands in the image spectrum. Each sub-image is quantized using an appropriate codebook. The relation of each sub-band to its codebook is saved in a map file. The process of searching a large codebook is eliminated in this scheme.

The conventional K-means algorithm is iterative in nature and requires a large amount of computation time for convergence. The computation time mainly depends on the amount of training data, codebook size, vector dimension, and distortion measure. A Modified K-means algorithm was presented by Lee et. al. [15]. This algorithm uses a scaled-updating scheme where a code vector is updated along the direction of the local gradient by a step-size larger than that used by the centroid update of the conventional K-means algorithm (or) the LBG algorithm. Therefore this algorithm provides faster convergence rate.

An improved version of the GLA presented by Veprek and Bradley [16] considers code vectors that contribute much in minimizing the reconstructed signal distortion for the successive iterations. This algorithm achieves better performance because of the consideration of significant code vectors. Linearly Constrained Generalized Lloyd Algorithm [17] combines the concept of Linearly Constrained Vector Quantization and GLA. The codebooks designed using this algorithm possess desirable properties such
as good sphere packing, low-complexity implementation, fine resolution, and guaranteed convergence.

Stochastic K-means algorithm for VQ [18] uses the concept of probability distribution in the existing K-means algorithm in associating a vector with a cluster which depends on the distance between the vector and the cluster gravity centre. It is less dependent than the K-means algorithm on the initial centre choice. It has been applied to vector quantization of speech signals. One of the main drawbacks of clustering algorithms is that the possibility of achieving poor clusters in the case of a bad choice of the initial codebook.

The concept of code word utility is used in Enhanced LBG algorithm to overcome this drawback. The Enhanced LBG algorithm [19] is able to generate better codebooks than previous clustering techniques and the computational complexity is virtually the same as the simpler LBG. Enhanced LBG concentrates on two concepts namely partitioning and centroid. In order to improve the performance of the Enhanced LBG algorithm some modification is made in the partitioning process [20]. This scheme assigns a weight to each code vector in the initial codebook. Further a weighted Euclidean distance which is a product of the Euclidean distance and weight of an involved code vector is used in the partitioning process. Then a centroid operation is performed based on this weighted partition for a given set of training vectors by iteratively adapting weights of different code vectors.

Generalized Lloyd Algorithm is an iterative gradient descent algorithm. This algorithm suffers from empty cell problem (i.e.) generation
of the codebook with some unused code vectors. Due to this reason the performance is sensitive to the initialization of the codebook. To overcome this problem a Modified Generalized Lloyd algorithm was presented by Jayanthi et.al. [21]. This algorithm identifies the unused code vector and replaces it with a training vector with the highest distortion.

An Enhanced Generalized Lloyd Algorithm using Pattern Reduction was presented by Chun-Wei et.al. [22]. The main focus of this algorithm is to reduce the computational complexity by using the pattern reduction approach. The quality of the reconstructed image is not affected because the size of the codebook is not reduced. Kekre and Tanuja’s [23] fast codebook generation algorithm is used to reduce the convergence time in order to optimize the vector quantized codebook. This algorithm generates unique minima for each code vector. As a result it takes less number of iterations as compared to LBG codebook.

3.3 TREE STRUCTURED VECTOR QUANTIZATION
Tree Structured Vector Quantization (TSVQ) algorithm for the design of codebook was presented by Buzo et. al. [24]. The codebook design process begins with the initialization of a root level codebook. Then each vector in the training set is mapped into one of the code vectors of the root level codebook. The second level codebook is obtained by applying LBG algorithm to each cluster. This process is repeated until the desired level tree structured codebook is generated. The storage complexities of a TSVQ are greater than those of a full-search vector quantizer, because TSVQ employs
a unique set of code vectors to encode the source vectors assigned to each node of the tree.

Constrained Storage Vector Quantization [25] uses the concept of codebook sharing by multiple vector sources in order to solve the problem of codebook storage complexity for the design of balanced tree-structured vector quantizers. The Variable-Length Tree-Structured Vector Quantizer [26] outperforms the balanced tree-structured vector quantizer. Variable-length trees grown by splitting nodes suffers from distortion errors. This scheme allocates more nodes (code vectors) to the parts of the tree thereby reducing the expected distortion. But the storage complexity is higher than that of a balanced tree-structured vector quantizer of the same rate.

Tree Structured Vector Quantization algorithms suffer from three shortcomings that result in suboptimal performance. They are greedy growing, sparse set of pruned trees and the use of suboptimal encoding rule. The Leaf Optimal Tree Design technique [27] eliminates these shortcomings. The application of this technique for image coding produces Entropy-Constrained Tree-Structured Vector Quantizer at arbitrary rates. The standard nearest-neighbor rule is used at all layers except the leaf layer and an optimal entropy-constrained nearest-neighbor rule is used at the leaves for the quantization process.

Fast Tree-Structured Nearest Neighbor Encoding for VQ was presented by Ioannis et. al. [28]. The initial codebook is stored at the root of a binary tree. GLA is applied to find two split values. The hyper plane that separates the two voronoi cells is determined. The signed differences between each code vector and the hyper plane are determined. The indices
of the code vectors with negative difference measure are stored at the left node and others at the right node. This process is repeated until each leaf contains only one code word. A greedy binary search is carried out from root to leaf. Search all code words associated with the node with opposite signed distance measure in order to select the best candidate code word. Then move to its parent node and repeat the process until the root is reached.

Ulug Bayazit and William A. Pearlman [29] introduced a non-adaptive greedy growth Variable-Length Constrained Storage Tree Structured Vector Quantizer codebook design algorithm. Instead of designing a unique sibling pair to quantize each set of residual vectors as in Variable-Length Tree-Structured Vector Quantizer, a sibling pair shared by two or more sets is designed to grow incrementally a different tree structure. Through the sharing of sub-codebooks this algorithm yields low storage complexity.

In Variable-Branch Tree-Structured Vector Quantization [30] genetic algorithm is employed for the searching process. The design of the growing method for the variable-branch tree encoder is a “top-down” approach, where the tree is built starting from the root node towards the leaf nodes until the desired rate coder is obtained. When a node is split owing to growing of the tree, the genetic algorithm can determine the number of child nodes of this node, required to maximize the ratio of the increase in distortion to the decrease in rate. The direct use of VQ suffers from a serious complexity barrier. To design for image compression with compact codebook approach.
3.4 RESIDUAL VECTOR QUANTIZER

Residual Vector Quantizer (RVQ) is a sequential search product code VQ, which uses direct sum product codebook structure and a sequential search procedure for the quantization process. The purpose of a direct sum constraint is to reduce the memory requirements and the purpose of the sequential search constraint is to reduce the computational complexity. First RVQ structure was presented by Juang, B.H and Gray A.H [32]. This algorithm generates codebooks in stages. The stages of RVQ are designed using GLA. It is observed that the sequential use of GLA gives optimum performance at high bit rates for a two-stage RVQ. For RVQ stages of more than two, this GLA based design procedure gives poor performance. It is because of this reason that the codebooks in each stage are generated considering only the previous stage errors and ignoring the subsequent stage errors. This quantizer has a sequence of quantization stages in which each stage quantizes residual vector of the prior stage. Therefore, RVQ is also called MSVQ. Various joint optimization techniques have been introduced to overcome this problem by taking into account both the previous stage and the subsequent stage errors in order to reduce the overall error.

Mean Residual Vector Quantization (MRVQ) was presented by Baker [33]. Initially, the mean of each vector is subtracted from each vector component. Then the codebook is designed for the residual vectors. This algorithm gives high compression ratio. However, edge or texture information cannot be adequately preserved with this scheme. Variable rate RVQ depends on entropy coding for their variable rate structure. The distortion is minimized by imposing a constraint on the output entropy of
the quantizer. Entropy coded RVQ [34] exploits the lower entropy of direct sum codebooks using an entropy encoder at the RVQ output. The distortion due to the direct sum codebook constraints are partially reduced by the use of this entropy encoder. However, the improvement in performance is achieved at the cost of storage area needed for the encoder lookup table.

In Fixed Rate RVQ [35], a new stage codebook is added whenever the relative change in distortion between outer loop iterations fell below a threshold value. Small training sets are sufficient to construct decoder codebooks. For example, this paper uses 64 training vectors to design a 32-stage RVQ. Since residual vector is iteratively fed back into a single/multiple stage codebook, this quantizer is referred to as Recursive Quantizer [36]. It is also called Summation Product Code VQ [37] because of the use of direct sum product codebook structure.

Predictive Residual Vector Quantization [38] generates a prediction residual vector. The generalized RVQ is used to quantize this prediction vector. The prediction residual vector is generated by predicting the current block from the previously quantized blocks. In Image adaptive RVQ [39] the codebook is designed using each source image that is to be coded rather than designing a generic codebook for the composite source. The training set representation of the composite-source probability density function often results in codebooks with mediocre performance on each of the different image types. This problem is solved with the use of image-adaptive RVQ.

Finite State Residual Vector Quantization method was presented by Syed A Rizvi, and N. M. Nasrabadi [40]. A Finite State Vector Quantization has a finite number of states in which a relatively small sized codebook is
associated with each state. The quantizer state is determined by the previous state and the previously quantized vectors. The input vector is then quantized using the current state codebook. Entropy-Constrained Residual Vector Quantization [41] uses an iterative descent algorithm based on a Lagrangian formulation for the quantizer design. The memory required for storing the entropy coder’s lookup table is very large since a variable length code word is stored for each direct sum index. Conditional Entropy-Constrained RVQ [42] takes the advantage of using a conditional chain rule formulation of direct sum index probability in order to reduce the entropy coder memory requirements.

Entropy-Constrained Residual Vector Quantization [43] is a joint optimization technique which combines a VQ and an entropy coder. The training sequence is used to generate the codebook. Then the algorithm searches for the direct-sum code vector whose Lagrangian function is minimized thereby reducing the average distortion subject to a constraint on the output entropy of the RVQ. Predictive Residual Vector Quantization [44] is an image coding technique which combines the concepts of Predictive VQ and RVQ. Neural network learning algorithms are used in the design process. The vector predictor is implemented by a multilayer Perceptron. The multiple stage codebooks of the RVQ are designed by the multilayer competitive neural network. It is shown that this combined approach outperforms the Predictive VQ and RVQ schemes.

Multi-Stage Vector Quantization is a structured VQ approach for low complexity implementation of high-dimensional quantizers, which has found applications within speech, audio, and image coding. Zhu Minhui et. al. [45]
designed an enhanced MSVQ coder for Satellite Aperture Radar images. In order to optimize the overall performance, the enhanced MSVQ coder uses a different design procedure for the decoder codebook. This design procedure is derived from the encoder codebook characteristics. The performance improvement is achieved only by modifying the decoder codebook design. Wavelet transforms have become more and more popular in the last decade in the field of image compression. Due to implementation constraints, scalar wavelets do not possess the properties like orthogonality and linear phase symmetry, which are very much essential for signal processing. To overcome these drawbacks, a new image coding algorithm based on non-linear approximation of multi-wavelet transform along with MSVQ was presented by Esakkirajan et. al. [46]. The performance of multi-wavelets is compared against scalar wavelets on different images along with the application of MSVQ on both the schemes.

3.5 FUZZY VECTOR QUANTIZATION
The application of fuzzy concepts for codebook generation eliminates the effect of initial codebook selection on the quality of clustering and avoids apriority assumptions regarding the level of fuzziness necessary for a given clustering task. The first Fuzzy K-means algorithm which is also known as fuzzy ISODATA algorithm was presented by Dunn [47]. Based on the idea of fuzzy concepts another fuzzy clustering algorithm was presented by Ruspini [48]. Following Dunn's formulation a Fuzzy K-means algorithm was presented by Bezdek et. al. [49]. Fuzzy Vector Quantization Algorithms [50] exploits the advantages offered by fuzzy clustering algorithms while
satisfying the requirements imposed by the VQ problem. The formulation of these algorithms is based on the strategies for the transition from soft decisions to hard or crisp decisions during clustering. Membership functions are employed for the quantitative measurement of the uncertainty associated with each vector assignment.

Fuzzy Algorithms for Learning Vector Quantization [51] is used to design multi-resolution codebooks. An image compression system using wavelet-based sub-band decomposition and Fuzzy VQ was presented by Nicolaos et. al. [52]. This approach transforms the input image into a set of sub-bands with different resolutions. Hierarchical Vector Quantization algorithm was presented by N. M. Nasrabadi [53]. This scheme partitions the image into blocks of different sizes. LBG algorithm is used to generate independent codebook for each block. The larger block is processed first. If the texture or information in the image is not adequately recovered then the next smaller block is processed and so on. The computational complexity and the memory requirement for this technique are greater than those compared with the MSVQ, MRVQ and TSVQ techniques. Hierarchical Multi-Rate VQ [54] exploits the concept of optimization and sub-sampling to compress images.

### 3.6 Hierarchical Vector Quantization and Adaptive Vector Quantization

Hierarchical structure sub-samples the original image into several layers according to their gray scale contrast. Codebook is generated using block segmentation that divides each layer into blocks of equal size. Variant bitrates are used for block coding of different layers within the codebook.
Hence this technique provides high encoded image quality with very low bitrates. Hierarchical Adaptive Search Vector Quantization technique for image compression was presented by Mohammad A. Ghafourian and Chien-Min Huang [55]. In this scheme the search starts with the largest block. The distortion is compared against the threshold. If the distortion is still greater after going down to the bottom of the tree structured codebook then the entire adaptive encoding process restarts for the next block size and so on. The best matching index along with the block size and the tree depth are transmitted. The performance of Hierarchical Adaptive Search Vector Quantization is evaluated over a bit rate of 0.625 bpp to 1.25 bpp. Comparison is made against the existing Adaptive Tree Search VQ, Multi-stage Adaptive Search VQ and the JPEG standard. Results indicate that Hierarchical Adaptive Search Vector Quantization outperforms other algorithms over the entire range of bits examined.

In Adaptive Vector Quantization (AVQ) [56] progressive code vector replacement and stochastic gradient based codebook adaptation are used. This concept is derived from the theory of adaptive signal processing. The parameters of the quantizer are updated during real-time operation based on observed information regarding the statistics of the signal being quantized. The problem of adaptively optimizing the entire quantizer in real-time using both the single parameter backward-adaptive and forward-adaptive techniques is described in this paper. AVQ algorithm [57] is used to design operational codebook that is adaptive to the changing source statistics during the quantization process. The output of an AVQ system includes indices and some side-information. Side-information consists of flags
indicating the status of the codebook updating process. This information is used to update decoder codebook.

Locality-based codebook updating and history aid codebook updating techniques that are applicable to AVQ are presented by Guobin Shen et. al. [58]. Locality-based codebook updating technique updates the operational codebook using the current input vector but also the code words at all positions within the locality. History aid technique updates the operational codebook using the information of previously coded vectors to quantize the current input vector if it is used to update the operational codebook. Further it is reported that the unified history aid and locality-based updating approach gives better performance gain. An improved AVQ algorithm was presented by Hsiu-Niang Chen and Kuo-Liang Chung [59]. This method generates a hybrid codebook consisting of three sub-codebooks and leads to a considerable computation-saving effect.

3.7 PREDICTIVE VECTOR QUANTIZATION AND CLASSIFIED VECTOR QUANTIZATION
Predictive Vector Quantization of images presented by Hsueh-Ming Hang and John W. Woods [60] consists of two parts namely the predictive part and the VQ part. The predictive part of the encoder use a predictive filter to remove the predictable redundancy in the data and then use a VQ to encode the prediction error. An image compression algorithm using Classified Vector Quantization of images was presented by Bhaskar Ramamurthi, and Allen Gersho [61], where the source image is viewed as a bank of sub-image vectors. At each instance, one of these sub-images is randomly selected with
a specified probability distribution. Each sub-image generates blocks with a perceptual feature, e.g., blocks with an edge at a particular orientation and location. The classifier is used to classify this image blocks. The classification algorithm is implemented in two steps: an edge enhancement step, followed by a decision tree which extracts the edge description from the enhanced version. To achieve near optimum performance in image compression, greater tree depth is required for the tree based coding and large vector length is required for the VQ process. However these factors are limited by the exponentially growing complexity of the source encoder. Hence to preserve the useful characteristics from these two coding processes, Predictive tree encoding of still images with VQ was presented by Jens-Rainer Ohm and Peter Noll [62]. This scheme is applied to the prediction error signals of images produced using non-adaptive and adaptive linear predictors by integrating the tree structure coder with VQ.

Predictive Classified Vector Quantization [63] predicts the class number of the input vector. Prediction is carried out in two parts: shade prediction and edge prediction. If the prediction is correct a codebook corresponding to that class is chosen to code this input vector. Only the shade/edge bit, the error code and the code index are transmitted. If prediction error occurs then the acceptable codebook will be selected in an iterative manner. If the error is irrecoverable then the class number will be transmitted along with the shade/edge bit, the error code and the code index. Sub-sampling Based Predictive Vector Quantization was introduced by Ce Zhu [64]. This scheme exploits the neighboring-pixel smoothness in the boundary areas of inter-blocks. The image is partitioned into non-overlapped
image blocks and the blocks are down-sampled. The sub-sampled blocks are quantized using VQ. Compared with the Predictive VQ method, this new method achieves significant improvement in terms of rate-distortion performance while maintaining comparable computation complexity.

3.8 PYRAMID VECTOR QUANTIZATION AND LATTICE VECTOR QUANTIZATION

Pyramid Vector Quantization (PVQ) takes its name from the geometric shape of the points in its codebook. PVQ uses the lattice points of a pyramidal shape in multidimensional space as the quantizer codebook. A lattice is a discrete set of points in the n-dimensional Euclidean Space $\mathbb{S}^n$, which can be generated by the integral linear combination of a given set of basis vectors. Enumeration assigns a unique index to all possible vectors in the PVQ codebook, imparting a sorting order to the PVQ codebook vectors. The first PVQ enumeration technique was introduced by Fischer [65]. It uses magnitude enumeration to sort each vector based on the magnitude of each of its elements. Lattice Vector Quantizers are optimal for uniformly distributed sources. Lattice truncation and scaling are the two steps involved in the design of a Uniform Lattice Vector Quantizer [66] for a specific lattice. Truncating a lattice is accomplished using normalization functions. But scaling of the lattice is accomplished experimentally. Image compression using nonlinear PVQ was presented by Xudong Song and Yrjö Neuvo [67]. Image pyramids are built using Multistage Median Filters in order to preserve fine details. Also, Multistage Median Filters de-correlates the difference pyramids, resulting in smaller first order entropy.
Vector Quantization of transform coefficients are usually referred to as transform coding techniques. An image coding algorithm that combines wavelet transform with Lattice VQ was presented by Michel et. al. [68]. The image is transformed using the bi-orthogonal wavelet transform. The wavelet coefficients are quantized using lattice codebooks with pyramidal boundaries. This scheme provides good coding results at low bit rates than the spatial Lattice VQ. The analytical methods for lattice truncation and scaling were presented by Dae Gwon Jeong and Jerry D. Gibson [69]. Further the Uniform and Piecewise-Uniform Lattice Vector Quantizers are used to quantize the DCT coefficients of the image. A Successive Approximation Vector Quantizer for Wavelet Transform Image Coding method was proposed by Eduardo et. al. [70]. This method employs successive approximation of the input vectors using only a finite set of code vectors called Successive Approximation Lattice Vector Quantization to quantize the transform co-efficient of the image. Further an adaptive arithmetic coder is used to encode the quantized output. This method eliminates the need for large lattice codebook.

An Error-Resilient Pyramid Vector Quantization scheme for Image Compression was presented by Andy et. al. [71]. Three new enumeration techniques namely linear enumeration and two product enumeration techniques (conditional product code enumeration and conditional product-product code enumeration techniques) that automatically yield robust indices have been introduced. Linear enumeration relies on constructing a better tree structure while product enumeration techniques form sub-ranges based on the number of significant elements in the vector. Further error
resilience is achieved through permutation of the PVQ codebook indices to minimize the effect of bit-errors on the indices. It is reported that product enumeration techniques reduce the susceptibility to channel noise by up to 3 db over existing methods for no additional coding overhead. The product enumeration techniques have efficient hardware variants and can be applied to other fixed-rate symmetric quantizers also. These new error-resilient PVQ techniques, when combined with sub-band coding, achieves better compression performance than the variable rate JPEG image compression standard.

N. M. Nasrabadi and Feng Y [72] gave the definition and basic theory of VQ algorithm for image coding. In addition, the application of VQ in the spatial, predictive, transform and hybrid domains are reviewed. Successive Refinement Lattice Vector Quantization [73] employs Multiple Lattice VQs for the codebook generation process. With successively decreasing scales this algorithm yields successive refinement VQ systems. Use of relatively small lattice codebooks rather than a single large codebook eliminates the problem of searching large indices. This scheme is more efficient than the gain shape type VQ algorithm. Multistage Lattice Vector Quantization with Adaptive Sub-band Threshold [74] reduces coding complexity and computation due to its regular structure. This technique concentrates on reducing the quantization error of the quantized vectors by blowing out the residual quantization errors with a scale factor. An optimized adaptive threshold scheme is used to identify significant coefficients of each sub-band.
3.9 FRAC TAL VECTOR QUANTIZATION

Barnesly [75] introduces the idea of fractal image coding. Fractal image coding utilizes self-similarity existing in an input image. The first automatic algorithm to find the contraction mappings for real images was presented by Jacquin [76]. In this fractal coder, the source image is approximated using a unique attractor. To reconstruct the attractor at the decoder, a mapping function is applied to an arbitrary image iteratively. An image coding algorithm for still images using VQ and Fractal Approximation was presented by In Kwon Kim and Rae-Hong Park [77]. This algorithm generates a static codebook for the low frequency components of the input image and an adaptive codebook for the residual signals. The static codebook is generated using a generalized VQ technique and the adaptive codebook is generated using Fractal coding. To obtain a domain pool, the image is approximated using a reversible transformation followed by decimation. This scheme is a non-iterative algorithm resulting in fast decoding.

Fractal coders generate the codebook by finding an optimal code word for each range block of the transformed source image. This process results in the generation of a codebook with redundant code vectors. In Fractal Vector Quantization [78], the source image is approximated coarsely by fixed basis blocks, and the codebooks are self-trained from the coarsely approximated image, rather than from an outside training set or the source image itself. Thus the redundancy in the codebook is eliminated without side information, since the decoder cannot be informed of the exact codebook generated in the encoder.
3.10 TRELLIS CODED QUANTIZATION

The origin of Trellis Coded Quantization [79] is from Ungerboeck’s formulation of Trellis Coded Modulation. Assuming an encoding bit rate of $B$ bits/sample, this algorithm first partitions a scalar codebook having $2^{B+1}$ elements into four subsets each containing $2^{B-1}$ code words. These subsets are used to label the branches of a suitably chosen Trellis. A Trellis is a sequence of subsets. Then the sequence of code words that minimizes the MSE between the image data and the selected code word sequence is found.

Fischer T.R, and Wang M [80] introduced Entropy-Constrained Trellis Coded Quantization. This algorithm allocates one bit/sample to specify a path through the Trellis. The remaining bits are used to specify a code word from the subset. This algorithm gives near optimal performance for image coding. It is reported that lower bit rates are achieved for Vector Quantizers rather than the Scalar Quantizers. Takanori Senoo and Bernd Girod [81] presented an analysis about various VQ techniques including Full search fixed length VQ, Entropy Constrained VQ and Lattice VQ examined for a base band image and sub-band images. The source image is the base band image and the sub-band images are obtained by dividing a source image into sub-images. It is shown that these VQ techniques yield almost optimum performance for sub-band images than the base band image.

Gain-shape VQ, Mean-Residual VQ, and Mean-gain shape VQ [82] are examples of this class. To find the best matching code-vector for each source vector the components of the product code are sequentially searched in predetermined order. Direct sum codebook structure is employed to search the components of the product codes instead of sequential search product.
code. Parthasarathy Sriram and Michael W. Marcellin [83] investigated the use of Entropy-Constrained Trellis-Coded Quantization for quantizing wavelet coefficients of images. The performance of this wavelet coder is compared with Adaptive Entropy coded sub-band coder [84], Entropy-Constrained Scalar Quantization based sub-band coder [85], and Entropy-Constrained Trellis-Coded Quantization based transform coder [86].

3.11 SIDE MATCH VECTOR QUANTIZATION

Side Match Vector Quantization [87] designs a state codebook with a subset of the code words with the smallest side-match distortions in the super codebook. However, the performance of Side Match Vector Quantization degrades if the gray levels of the pixels across the boundaries between neighboring blocks is increasing or decreasing. This problem is overcome in the smooth side-match method by considering the smoothness of the gray levels between neighboring blocks.

Chen T. S and Chang C. C [88] introduced a new image coding algorithm using variable-rate side-match finite-state vector quantization. Side-match technique is used to select the optimum codebook state for each input image block. Smooth Side-Match Classified Vector Quantizer [89] employs Clustering algorithm to find the number of edge classifiers for high-detail blocks and to design the codebooks. A smooth side-match method is used to select a state codebook based on the smoothness of the gray levels between neighboring blocks. It is given that Smooth Side-Match Classified Vector Quantizer outperforms both the Classified Vector Quantizer and the Side Match Vector Quantization algorithms in terms of image coding quality.
3.12 NEAREST NEIGHBOR SEARCH ALGORITHM

Many image compression algorithms have been developed to lessen the computation load. The Equal-average Nearest Neighbor Search (ENNS) algorithm [90] uses the mean value as a feature to reject unlikely code words. However, two vectors with the same mean value may have a large distance. This algorithm reduces computational time further with $2N$ additional memory. Where, $N$ is the VQ level. The Equal-average Equal-variance Nearest Neighbor Search (EENNS) algorithm [91] uses the variance as well as the mean value of the input vector separately to reject more code words. This algorithm reduces computational time further with 2 additional memories.

The Improved EENNS algorithm [92] uses the mean and the variance of an input vector simultaneously, but develops a new inequality between these features and the distance. The Improved ENNS algorithm [93] is based on the Improved Absolute Error Inequality criterion. The algorithm divides the input vector into two sub-vectors. Two inequalities based on the sums of its two sub-vector components are used to reject those code words that cannot be rejected by ENNS.

A new theorem that defines three inequalities based on the sum and variance of a vector and its two sub-vectors components was introduced by Jeng et. al. [94]. These inequalities are used to reject those code words that are impossible to be the nearest code word, thereby reducing the computational overhead. The algorithm uses this theorem to eliminate many of the impossible matching code words which cannot be eliminated by the ENNS, Improved ENNS, EENNS and the Improved EENNS algorithms.
3.13 JOINT PHOTOGRAPHIC EXPERTS GROUP

The Joint Photographic Experts Group has released two international standards for image compression namely the JPEG and the JPEG 2000. Joint Photographic Experts Group still image coding standard [95] has been widely employed in applications involving storage and transmission of still images and graphics over the network. The evolution of image coding technology and an increasing field of applications have inspired the JPEG committee to start a new project in 1997 and to develop the next generation still image coding standard. The joint effort of the ISO and the ITU (International Telecommunication Union) resulted in the JPEG 2000 International Standard [96]. Hong Man et al. [97] presented an article that serves as a guideline for the evaluation of the effectiveness of JPEG 2000 for various applications, and for the selection of effective options and parameter settings according to particular application requirements.

3.14 QUAD TREES

Most commonly Quad trees are constructed either by the Top-down approach or the Bottom-up approach. Top-down approach [98] decides whether the entire block can be represented as a single leaf or should it be divided into four blocks. If the block is divided, then a binary decision is made for each block to find out whether it needs further division and so on. Bottom-up approach [99] makes out decisions to merge the smallest possible blocks. If all possible blocks have been combined into larger blocks then a decision is made whether to combine larger blocks into yet another larger block and so on.
Quad Tree decomposition [100] is a powerful technique which divides an image into 2-D homogeneous regions. The decomposition builds a tree. Each node has four children and it is associated with unique regions of the image. To compress the image, the tree structure and the leaf information of the resulting tree are coded. Gray introduced an optimal Quad tree construction using Lagrange multiplier [101] to find optimal rate allocations with no monotonic restrictions. This method uses a stepwise search to determine the overall structure of the Quad tree. The search is driven with the average predicted performance.

3.15 FINITE-STATE VECTOR QUANTIZATION

Dynamic Finite-State Vector Quantization (DFSVQ) [102] assumes that the quantizer is in a state with a dedicated sub-codebook for each input vector. This sub-codebook is generated dynamically from the super codebook. The state is determined by the next-state function. The next-state function should effectively predict the behavior of the next input vector and select the closest code vectors from the super codebook to create a sub-codebook in order to use only the sub-codebook for the quantization process. Consequently, the bit rate and computational complexity can be reduced significantly.

In practice, the best code vector representing the input vector may not be found in the sub-codebook. This can be solved by the Adaptive DFSVQ [103], in which the sub-codebook is first searched and if the distortion between the input vector and the best match in the sub-codebook is greater than a predefined threshold, then the super codebook is searched for a better match. The choice of the next-state function is the key element of DFSVQ.
which determines the performance of the system. Several next-state functions have been presented by N. M. Nasrabadi, and Rizvi S. A [104]: the conditional histogram, index prediction, vector prediction, nearest neighbor design and frequency usage of code vectors. The authors concluded that the nearest neighbor design seems to be a good choice, although it needs more computational cost.

Jyi-Chang et. al. [105] presented a New Dynamic Finite-State Vector Quantization (New DFSVQ) algorithm for image compression. The new DFSVQ exploits the global inter-block correlation of image blocks instead of local correlation in conventional DFSVQs. Side-match technique is used to search the closest vector from the previously quantized vector for each input vector. The closest vector is used to generate the dynamic codebook. It is reported that this algorithm provides better picture quality than that of the conventional memory VQ techniques.

3.16 NEURAL NETWORKS BASED VECTOR QUANTIZATION

The role of Neural Networks in the field of image compression is analyzed and an unsupervised learning based codebook generation method was presented by Stanley et. al. [106]. The training process in a CPL Network is unsupervised. This network is used to generate the codebook needed for the quantization process. The performance of this work and traditional non-neural algorithms for VQ is also compared. VQ techniques suffer from stability-plasticity dilemma, dead-node problem and deficiency of local minimum. To overcome these drawbacks Jung-Hua Wang and Chung-Yun Peng [107] used a self-creating neural network scheme. This scheme
achieves the biologically plausible learning property and also harmonizes equi-error and equi-probable criteria. The role of SOM for clustering was demonstrated by Arthur-Flexer [108]. It is enlightened that the number of output units used in this neural network influences its applicability for either clustering or visualization. Further it is understood that SOM is a flexible tool which can be used for various forms of explorative data analysis but it is also made obvious that this flexibility comes with a price in terms of impaired performance.

Marcos M. Campos and Gail A. Carpenter [109] introduced two unsupervised learning algorithms in which one features a new tree-building algorithm that can be implemented with various cost functions and another a new double-path search procedure using Self-Organizing Trees. These algorithms are used to construct hierarchical representations of data and online tree-structured vector quantizers. The use of Competitive learning algorithm (CPL) [110] for the codebook generation reduces the overall distortion. This algorithm normalizes connection strength between the weight vectors and the input patterns in order to reach a local minimum and also regulates the repulsion between the weight vectors and the input's gravity center to favor convergence to the global minimum. Hence it leads to near optimal solution and allows the network to escape from local minima during training. This approach outperforms the simple competitive learning algorithm by generating better codebooks.

Arijit et. al [111] used Self-Organizing Feature Map to construct a generic codebook. Further cubic surface fitting technique is applied on the code vectors of this codebook in order to enhance the codebook. The aim of
generic codebook design method is to reduce the compression bit rate. It is reported that the algorithm performs better in the spatial domain than in the transform domain. Kwang-Baek et. al. [112] presents an Enhanced Self-Organizing Map algorithm for medical image compression in the wavelet domain. With this approach the weight updation is done based on the frequency of the winner node or on the ratio of the present change in weight to the previous change in weight. This enhanced SOM algorithm is used in the codebook generation phase of VQ in-order to reduce the time needed for searching the codebook to find the best matching code vector for an input vector.

To reduce this computational overhead a fast codebook searching method for a Self-Organizing Map (SOM) based vector quantizer was proposed by Arijit et. al. [113]. A non-exhaustive search method is used in this quantizer to find a matching code vector from the codebook for an input vector instead of the exhaustive search in a large codebook with high dimensional vectors. Learning Vector Quantization (LVQ) has attracted great attention in recent years as a technique for image coding. Mihajlo and Slobodan [114] introduced a modification of the LVQ algorithm that addresses problems of determining appropriate number of prototypes, sensitivity to initialization, and noise in data. This algorithm allows adaptive addition of prototypes at beneficial locations and removal of harmful or less useful prototypes. Various methods to improve the performance of neural networks for image compression have been proposed recently [115], [116], [117], and [118].
3.17 OTHER VECTOR QUANTIZERS

Product Code vector Quantizer [119] is a structured vector quantizer class that simultaneously reduces both computation and memory requirements. In this class of VQ, different attributes or features of the source outputs are quantized by different components of the quantizer. Two-Stage Vector Quantizer [120] uses a structured and an unstructured codebook. First stage uses an unstructured codebook and a spherical lattice codebook is used in the second stage. The reason for this joint optimum codebook is based on the assumption that the distribution of the first stage quantization error is modeled as Gaussian. The coding performance is superior to that reported for the Uniform/ Piecewise-Uniform Lattice Vector Quantizers.

Discrete Cosine Transform for image compression was presented by Keith A. Birney and Thomas R. Fischer [121]. As a result of the Kolmogorov Smimov goodness-of-fit tests conducted, it is concluded that the transform coefficients can be better modeled as Laplacian than as Gaussian, Raleigh or Gamma distribution. To achieve low bit rate and high quality image coding a Combined Techniques of Singular Value Decomposition (SVD) and VQ for Image Coding was presented by Jar-Ferr Yang and Chiou-Liang Lu [122]. The SVD transformation has the characteristic of optimal energy compaction in the least square sense. It makes SVD useful for image coding. The image is first transformed using SVD. The SVD coefficients are linearly quantized using K-mean VQ. The results are compared to the DCT coder. It is shown that SVD coder performs better than the DCT coder in terms of energy compaction, data rate, image quality and decoding complexity. In image compression algorithms using VQ distortion is introduced by an arbitrary
exchange of training vectors among clusters. Carlo Braccini et. al. [123] analyzed the effects on distortion using Apriori explicit analysis. The MSE distortion is reduced based on the observation made from the analysis.

The two stage Quantization [124] uses scalar quantization to quantize pixel blocks and VQ to compress prediction errors resulting from the first stage. This algorithm characterizes each pattern into a key value. The key values are used in the reflection operation to get more patterns. As a result the visual quality of the reconstructed image is improved. Pasi Fränti et. al. [125] analyzed the existing variants of the splitting method and its application to codebook generation with and without the GLA. In addition, an algorithm which integrates hyper plane partitioning of the clusters along the principal axis with a local repartitioning phase at each step is also presented.

In Reduced Storage Vector Quantization via Secondary Quantization [126], a secondary quantizer is used to quantize the code vectors of the primary quantizer. Any existing (primary) quantization schemes may be considered for secondary quantization. It is reported that significant storage reduction is obtained with little loss of reconstructed quality. The combined Nonlinear Principal Component Analysis method with VQ for image coding was presented by Dimitrios Tzovaras and Michael G. Strintzis [127]. Back propagation neural network is used to design Nonlinear Principal Component Analysis and LVQ is used for the VQ design. Further, codebook vector optimization procedure is used to reduce the effects of quantization in the quality of the reconstructed images.

A sequence of merge and split operations [128], [129] is used to improve the initial codebook designed for VQ. The merging operation
combines small neighboring clusters while releasing additional code vectors. The code vectors released by the merging process can be reallocated by splitting large clusters. The process is repeated until distortion of the codebook is minimized. Genetic Algorithm is a powerful tool for the codebook generation of VQ in image compression. In the conventional Genetic algorithm based codebook design for VQ the generation of best individuals (children) that satisfies a fitness criterion is a time consuming problem.

To overcome this computational overhead a partition-based genetic algorithm that uses simulated annealing algorithm was presented by Hsiang-cheh Huang et. al. [130]. The algorithm uses linear scaling technique to calculate the objective functions and uses special crossover and mutation operations in order to obtain better codebooks in much shorter CPU time. Projection and Triangular Inequalities are defined by Jim Z. C. Lai and Yi-Ching Liaw [131] to reduce the computation time needed for VQ. These two inequalities are based on the mean value, edge strength, and texture strength of the input vector. The triangular inequality is used for terminating the searching process and the projection inequality is used to delete impossible code words thereby reducing the distortion computations.

Multi-resolution Vector Quantization algorithm [132], [133] introduces optimal vector quantizers for multi-resolution source coding scheme. This algorithm is designed using an iterative descent technique on a Lagrangian performance measure. The choice of the Lagrangian parameters is equivalent to selecting a point on a hyper plane tangent to the convex hull of the space of achievable rate-distortion vectors. The algorithm achieves
performance improvements over prior codes at the expense of increased computational complexity. A novel multiple transform domain split VQ technique for image compression was presented by Wasfy B. Mikhael, and Pradeep Ragothaman [134]. The image is partitioned into sub-images first. The histogram measure is obtained for each sub-image in order to select suitable transform. After this process the transformed coefficients are subject to VQ. This scheme results in consistent improvement in the reconstructed images compared with existing single transform approaches, for the same compression ratio.

Hongwei et. al. [135] presented a scheme that combines the virtues of the Principal Component Analysis and Genetic algorithm for codebook generation with optimal coding performance. Principal Component Analysis is used to sort the training vectors thereby reducing the computational complexity while the near global optimal searching ability of Genetic algorithm is used to generate a codebook with minimum distortion. This algorithm is suitable for image compression where a high compression ratio is needed. An enhanced version of VQ using reflections of triangular sub-code vectors was presented by Makkapatti et. al. [136]. The code vector is divided into triangular sub-code vectors along the diagonal axes. The reflections of these sub-code vectors about the diagonal axes are utilized to generate orientations of a code vector. This method reduces the memory required for a codebook significantly by generating orientations of code vectors.

Jim et. al. [137] presented Codebook Generation Using Code word Displacement. The code word displacement measure is computed for the successive partition processes. This method reduces the codebook
generation time considerably. The superiority of this method is more remarkable when a larger codebook is generated. Jim Z.C. Lai and Yi-Ching [138] presented a novel encoding algorithm for VQ. A set of transformed code words and partial distortion rejection are used to determine the reproduction vector of an input vector. This method reduces the computing time and number of distance calculations significantly. The performance of this method is better when a larger codebook is used.

3.18 OTHER IMAGE CODERS

The process of wavelet decomposition has localization properties in both time and frequency domains. In wavelet based image coders [139] and [140], the image is decomposed into several sub-images prior to the encoding stage. As a result it is usually more efficient to encode a transformed image than to directly encode the pixels. A transform based image compression using VQ was presented by Wang Kai and Song Guowen [141]. The application of a rectangular transform reduces the correlations that exist among the image pixels. The transformed output is subject to VQ. This combination cuts down the dimensions of vector coding. The size of the codebook can reasonably be reduced. Hence computational complexity of the VQ process is reduced.

Embedded wavelet image coder was presented by Sharpio [142]. In many coding algorithms the occurrence of a single error irreparably damages the image. The embedded nature of the bit stream produced by this coder provides a certain degree of error protection. In particular, only the information which arrives before the occurrence of the first bit error is used to reconstruct the image; everything that arrives after is lost. The error is
detected when the decoder terminates by decoding a stop symbol before reaching its target rate or distortion.

An Embedded Zero tree Wavelet image compression algorithm was presented by Sharpio [143]. This algorithm progressively quantizes the coefficients using a form of bit plane coding to create an embedded representation of the wavelet coefficients of the input image. This process results in a representation in which a high resolution image containing all coarser resolutions. This is accomplished by comparing the magnitudes of the wavelet coefficients to a threshold S to determine the significant coefficients. Then the sign and position of all significant coefficients are encoded. For all significant coefficients only the resolution enhancement bits are transmitted after they are arithmetically encoded.

Ping Wah Wong and Steven Noyes [144] presented a wavelet coder based on space frequency partitioning where the product of the spatial coverage and the frequency coverage is a constant for all blocks. Each block is assigned an individual bit rate and is encoded accordingly using scalar quantization. Further it is suggested that the performance of this coder can be improved by employing a vector quantizer. The Robust Embedded Zero tree Wavelet [145] image compression algorithm divides the wavelet coefficients of the input image into K blocks. These blocks are quantized and coded independently resulting in K different embedded bit streams. These bit streams are then grouped into appropriate bits, bytes or packets prior to transmission in order to maintain the embedded nature of the composite bit stream. The occurrence of a single bit error truncates only the respective stream while the other bit streams are still completely received. This is
achieved by coding the wavelet coefficients with multiple, independent bit streams. But in the basic Embedded Zero tree Wavelet transform bits arrived after the occurrence of the single bit error are truncated.

Accordingly, the wavelet coefficients represented by the truncated stream are reconstructed at reduced resolution while those represented by the other streams are reconstructed at the full encoder resolution. If the set of coefficients in each stream spans the entire image, then the inverse wavelet transform in the decoder evenly blends the different resolutions so that the resulting image has a spatially consistent quality. Furthermore, the search complexity of this coder actually decreases slightly with the image blocks. An embedded image coder generates an encoded image bit stream that contains embedded subsets affording efficient compression of the original image at reduced resolution (spatial scalability) or increased distortion (quality scalability) [146].

An embedded image coder that exploits the intra band correlation in wavelet domain and pixel sorting was presented by Kewu Peng and John C. Kieffer [147]. In addition to better compression performance, this method provides either spatial or quality scalability with flexible complexity. The Set Partitioning in Hierarchical Trees algorithm has been used in the field of image compression in recent years. This algorithm is used for image compression through ordering of wavelet coefficients into subsets and bit plane quantization of significant coefficients. In order to exploit the correlation among the significant coefficients in each pass a trained scalar or quantization is performed depending on a boundary threshold [148]. In each
pass, the decoder reconstructs coefficients with scalar or vector quantized values rather than with bit plane quantized values.

Wentao Wang et. al. [149] investigated the problem of quantizing the wavelet coefficients in the lowest frequency sub-band with multi-scalar Set Partitioning in Hierarchical Trees method. In the higher bit plane, this algorithm quantizes the wavelet coefficients in the lowest frequency sub-band and quantizes other ones by uniform scalar. It gives better coding gain in the low bit rate image coding.
4. IMAGE COMPRESSION USING BANDELETS

4.1 INTRODUCTION
This chapter explores the application of a multiscale directional procedure known as Bandeletization that provides an optimally compact basis for images by exploring their directional characteristics. Having a compact basis is useful both for compression and for designing efficient numerical algorithms. The motivation behind the usage of the bandelet transform is that the geometry of the image is summarized with local clustering of similar geometric vectors and the homogeneous areas being taken from the quad tree structure. This process helps to identify the areas in the image that are geometrically similar to each other. Then the spatial and geometric interpixel redundancies present in the bandeletized coefficients are removed. The psycho-visual redundancies are removed using simple Vector Quantization process. Finally, the consequential coefficients are encoded using Huffman encoder. Experiments based on the proposed scheme achieve near-optimal rate-distortion performance for natural images.

Sparse representation and multi-resolution properties [150] have been utilized in the computing field to speed up various numerical operations [151]. Sparse representation of the images results in faster computation of their linear combinations since only the non-zero coefficients are considered. Multi-resolution makes it easy to perform the warping operation at multiple resolutions, as well as in a coarse-to-fine fashion. The wavelet transform [152] has emerged as an excellent tool for image modeling. The success of wavelets is due to the fact that they provide a
sparse representation for smooth signals interrupted by isolated discontinuities [153]. The success of wavelets does not extend to two-dimensional images. Although wavelet-based image processing algorithms define the state-of-the-art, they have significant shortcomings in their treatment of edge structure. Large wavelet coefficients cluster around the contours irrespective of the smoothness of the contours. The wavelet transform is not sparse for images that are made up of smooth regions separated by smooth contours [154]. It simply takes too many wavelet basis functions accurately to build up an edge. In short, the wavelet transform makes it easy to model gray scale regularity, but not geometric regularity.

Several recently proposed directional approaches use the lifting scheme in image compression algorithms. This scheme has been exploited by Gerek and Cetin [155] where transform directions are adapted pixel-wise throughout images. To enhance wavelets representations, Ding et al. [156] have proposed to approximate the wavelet coefficients using adaptive vector quantization techniques. However, even though these methods are computationally efficient and provide good compression results, they show a weaker performance when combined with zero-tree based compression algorithms. Following the work on adaptive coding scheme, new lifting algorithms [157], [158] have also been proposed to predict wavelet coefficients from their neighbors. These works are mostly algorithmic and do not provide mathematical bounds. They use the fact that wavelet coefficients inherit some regularity from the image geometric regularity.

Filter Bank Techniques uses windowing of the sub-band coefficients. It may lead to blocking effects. To overcome this problem, Yuichi et al. [159]
proposed the multiresolution image representations using a combination of two dimensional filter bank and directional wavelet transform. This approach overcomes the block based approach of the curvelet by using a directional filter bank [160]. Many adaptive geometric representations have also been proposed recently with good results in image processing [161]. Instead of decomposing an image in a fixed priori basis, an adaptive algorithm shall be used to modify the image representation [162]. Adaptive techniques are techniques where the directional component of an image is adaptively estimated and the transform is steered based on the estimate. For example, the Bandelet transform of Pennec and Mallat [163] links the significant wavelet coefficients along a discontinuity and represents it as a smooth one dimensional curve geometry computed from the image as there is a strong coherency among the coefficients which is imposed by the structure of the geometry.

In the proposed work, the problem of Image Compression is addressed from a new angle. Instead of using the multi-resolution theory, it is projected as a problem of a geometrical similarity optimization. This work introduces a new class of transform bases called bandelet bases, which decomposes the image along multi-scale vectors that are elongated in the direction of a geometric flow. This geometric flow indicates the direction in which the image gray levels have regular variations. The image decomposition in a bandelet basis is implemented with a fast sub-band filtering algorithm. Bandelet bases lead to optimal approximation rates for geometrically regular images. Comparisons are made for image compression with wavelet bases.
4.2 THE BANDELET THEORY

Bandelet transform is a redundant transform with twice number of coefficients as critically sampled 2-D discrete wavelet transform. The challenge of using a redundant transform in coding applications is to select basis functions to provide a sparse representation in which only a small portion of coefficients have large magnitude. This problem is actually to optimize a measurement of coefficient sparseness subject to a set of suitable linear equations.

Bandelet transform exploits anisotropic regularity by constructing orthogonal vectors that are elongated in the direction where the function has a maximum of regularity. Bandelet bases decompose the image along multiscale vectors that are elongated in the direction of a geometric flow. This geometric flow indicates directions in which the image gray levels have regular variations. Bandelets in a region $\Omega$ are computed by applying a bandeletization to warped wavelets, which are separable along a fixed direction (horizontal or vertical) and along the flow lines. The geometric flow in a region $\Omega$ is a vector field $\vec{V}_i[m,n]$ defined over the image sampling grid $G$. If the flow is vertically parallel then $\vec{V}_i[m,n] = \vec{V}_i[m] = (1, c'_i[m])$, where $c'_i[m]$ measures an average relative displacement of the gray levels $\Omega_i$ in along the line $m$ with respect to the line $n - 1$. If the geometric flow is horizontally parallel in $\Omega_i$, then $\vec{V}_i[m,n] = \vec{V}_i[n] = (c'_i[n], 1)$. Given the original image sample values $f(m,n)$, at each grid point $G_i[k_1,k_2]$ the resampling computes an interpolated image value that is written $V_i[k_1,k_2]$. For a flow vertically parallel, the grid points $(k_1,k_2 + c_i[k_1]) \in \Omega_i$ are obtained with one-dimensional translations along the $y$ direction of the integer
sampling grid \((m, n) \in \Omega_t\). If the flow is horizontally parallel then the one-dimensional translation is along the \(x\) direction. For image compression and noise removal applications, bandelet basis produces a minimum distortion with \(O(n^2 (\log_2 n^2))\) operations for an image of \(n^2\) pixels. A full detailed description of the Bandelet transform is given by Mallet et.al. [164].

The input image is decomposed using the warped Haar Transform. The basis function for this transform is formed by the translation and dilation of three mother wavelets for the horizontal, vertical and diagonal directions. After transformation the quad-tree is constructed by dividing the image into dyadic squares. For each square in the quad-tree the optimal geometrical direction is computed by the minimization of a Lagrangian. This function requires a threshold to determine significant geometry coefficients. Even in squares with no geometric features (on which the function is constant), the algorithm chooses some arbitrary orientation because the Lagrangian function does not have zero mean, so a bandelet transform (with any direction) is better than leaving the data untransformed. This situation does not appear in the wavelet algorithm, since in flat areas, a wavelet transform has zero mean. Then a projection of the transform coefficients along the optimal direction is performed. Finally a one dimensional Haar transform is carried on the projected coefficients. Particularly, the size and the optimal geometrical direction of each square will be used as criteria to study the similarity. The inverse discrete bandelet transform computes the image values on the original integer sampling grid \((m, n)\) from the sample values \(V_i[k_1, k_2]\) along the flow lines in each \(\Omega_t\) where \((k_1, k_2) \in \mathbb{Z}^2\).
4.3 THE PROPOSED SCHEME FOR IMAGE COMPRESSION

This chapter proposes a relatively simple transform coder with a quantization and entropy coding of all coefficients. The transformation stage decomposes the original image into bandelet basis associated to the optimized partition and its geometric flow. The choice of the size for the dyadic squares, the threshold for Lagrangian computation and a scale factor for this decomposition stage influences the performance of the proposed work. Bandeletization results in transformed coefficients with sparse representation. The next stage analyses the correlation among the row vectors, column vectors and block vectors of the decomposed image. Statistically, the correlation coefficient \( r \) is defined as,

\[
r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}}
\] (4.1)

Where \( A, B \) are two vectors (row/column/block) in the transformed input matrix; \( \bar{A}, \bar{B} \) are the respective mean values of the vectors \( A, B \). The concept of point estimation is used to find out the sequential pattern. Point estimation refers to the process of estimating a population parameter (e.g. correlation), by actually calculating the parameter value for a population sample [165]. Initially, \( A \) is assigned with the first row/column vector of the input to this stage and \( B \) with the second vector. The correlation between \( A \) and \( B \) is computed using Eqn. (4.1). A threshold value is used to identify the correlation between \( A \) and \( B \). If \( 0.8 < r < 1 \), then \( A \) and \( B \) are said to be highly correlated. Then, the third vector is assigned to \( B \) and the process of computing the correlation is repeated. This process of comparison is repeated for few input vectors and then the sequential patterns present in
the input matrix are determined. These patterns are indexed. Vectors with high correlation are considered for further processing.

A simple quantization process is used to reduce psycho-visual redundancy. This process clusters the near similar vectors and generates a codebook consisting of a representative code vector for each cluster. Subsequently, the application of Zero Vector Pruning (ZVP) process reduces the inter-pixel redundancy. ZVP identifies all non-zero vectors along with their indices thereby eliminating redundant zero vectors. Finally, Huffman encoder reduces the coding redundancy. The amount of compression is measured at this stage. The algorithm decompresses the compressed data by using Huffman decoder, reverse ZVP and vector reassignment procedures followed by reverse bandelet transformation in order to reconstruct the image. The reconstructed image quality is tested at this stage. The flow of the proposed work is depicted in Figure 4.1.

4.4 EXPERIMENTAL RESULTS
To evaluate the performance of this bandelet based compression algorithm, a comparison is made with the same coder applied to a wavelet and wavelet packet based compression system. Figure 4.2a shows 256 x 256 Barbara input image. Figure 4.2b shows the respective reconstructed image using the proposed Bandelet based Vector Quantizer [BVQ] while Figures. 2c, 2d and 2e depict the same using Wavelet based k-means VQ Coder [Wk-means VQC], Wavelet Packet based Linde-Buzo-Gray Coder [WP-LBGC] and the Wavelet-Packet based k-means VQ Coder [WP-k-means VQC] respectively.
Figure 4.1 The Proposed Bandelet based Vector Quantization Scheme

Figure 4.2 Original Image and Respective Reconstructed Images obtained using the Proposed and Existing methods (PSNR: 24db) : (a) 256 x 256 Original Image, (b) Reconstructed Image using the Proposed BVQ, (c) Reconstructed Image using Wk-means VQC, (d) Reconstructed Image using WPLBGC, and (e) Reconstructed Image using WPk-means VQC
It is observed from Figure 4.2 that Wk-means VQC (Figure 4.2c) suffers from more pronounced blocking artifacts. Though the effect of blocking artifacts is reduced using WpLBGC (Figure 4.2d), it is observed that this method suffers from smoothening effect and hence ignoring the detail information. And Wpk-means VQC (Figure 4.2e) eliminates blocking artifacts, but detail information is not preserved substantially. The proposed BVQ (Figure 4.2b) preserves details because of its nature of preserving geometry. Further this coder reduces blocking artifacts in the reconstructed image thereby improving the psycho-visual quality remarkably.

Figure 4.3  Original Images and their Respective Reconstructed Images using BVQ Algorithm: (a) Wood and Stock Textures Image, (b) Seabed Seismic Image, (c) Geometric Shapes Image and (d) Geometric Pattern Image; (e), (f), (g), and (h) the Respective Reconstructed Images of (a), (b), (c), and (d).
Barbara image is purposely chosen as the test image, since it contains more detail information which helps in the measure of the subjective evaluation of the quality of the reconstructed images using the proposed and other existing methods. The proposed coder is tested with various images which include seismic, texture and geometric images. Few of these images and their respective reconstructed images obtained using the proposed coder is shown in Figure 4.3.

**Table 4.1 Performance Of The Proposed Work At Various Square Sizes And Scale Factors For The Cameraman And Barbara Images**

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>SQUARE SIZE (Ω)</th>
<th>SCALE FACTOR</th>
<th>PSNR (db)</th>
<th>COMPRESSION RATIO</th>
<th>COMPUTATION TIME (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman</td>
<td>4</td>
<td>0</td>
<td>24.11</td>
<td>11.16</td>
<td>8.79</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>27.42</td>
<td>6.03</td>
<td>25.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>27.68</td>
<td>4.38</td>
<td>51.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>27.80</td>
<td>3.46</td>
<td>76.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>27.87</td>
<td>2.97</td>
<td>117.43</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>25.49</td>
<td>11.14</td>
<td>28.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>28.18</td>
<td>6.17</td>
<td>83.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>28.54</td>
<td>4.35</td>
<td>176.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>28.72</td>
<td>3.30</td>
<td>307.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>28.77</td>
<td>2.59</td>
<td>465.17</td>
</tr>
<tr>
<td>Barbara</td>
<td>4</td>
<td>0</td>
<td>23.40</td>
<td>9.67</td>
<td>8.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>26.53</td>
<td>4.59</td>
<td>23.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>26.74</td>
<td>3.29</td>
<td>48.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>26.78</td>
<td>2.64</td>
<td>92.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>26.84</td>
<td>2.31</td>
<td>136.31</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>0</td>
<td>24.21</td>
<td>9.67</td>
<td>27.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>27.20</td>
<td>4.68</td>
<td>83.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>27.50</td>
<td>3.33</td>
<td>176.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>27.59</td>
<td>2.53</td>
<td>351.36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>27.62</td>
<td>1.98</td>
<td>538.26</td>
</tr>
</tbody>
</table>
The implementation stage of the proposed work necessitates fixing the size for the dyadic squares ($\Omega$), the threshold ($\tau$) for Lagrangian computation and a scale factor for the bandeletization process. Selecting optimal values for the dyadic square size and the scale factor are influenced by parameters like PSNR, Compression Ratio and the computation time.

The proposed work is tested with the two possible square sizes, $\Omega = 4$ and $\Omega = 8$ and with the scale factors 0 to 4. The results are recorded in Table 4.1. It is observed from this table that for a constant scale factor though the computation time increases with an increase in window size, the reconstructed image quality is improved significantly with a negligible change in the compression ratio (rate-distortion trade off). For example, for the cameraman image, a scale factor ‘0’ promises 24.11db PSNR and a compression ratio of 11.16 for $\omega = 4$; and for $\omega = 8$, the PSNR value is 25.49db and the compression ratio is 11.14. Since the change in compression ratio is negligible, $\omega$ resulting in a higher PSNR value is chosen for further observations. Hence the size $\omega = 8$ is used in this work. Further it is noted that the quality of the reconstructed image is not affected radically with varying scale factors and a constant dyadic square size. But the compression ratio varies from 11.14 to 2.59 as the scale factor is varied for the Cameraman image with $\omega = 8$. In addition, it is clear from that table that the computation time increases as the Scale Factor increases from 0 to 4. Hence a scale factor of ‘0’ is chosen. The graph shown in Figure 4.4 depicts the impact of the Threshold ($\tau$) value for Lagrangian computation. It obvious from the graph that as $\tau$ increases upto a certain value, PSNR also increases and for further increase in $\tau$, PSNR starts decreasing. It is seen in the figure that the
gain in PSNR is maximum with $\tau = 0.4$. After bandeletization, the coefficients are divided into subgroups (considering each row / column/ 4 x 4 blocks / 8 x8 blocks / 16 x16 block as a subgroup). The correlations among these groups are determined by computing $|r|$. The observations are recorded in Table 4.2. From this Table it is observed that the higher the correlation the lower the distortion is. Further, it shall be observed that the bandeletized coefficients possess high correlation along the row vector as compared to that along the column and other block vectors.

![Graph showing PSNR vs Threshold](image)

**Figure 4.4** Threshold Vs Image Quality

**TABLE 4.2** Significance of Correlation Analysis

| IMAGE   | IMAGE BLOCKS      | $|r|$ | MSE    |
|---------|-------------------|------|--------|
| Cameraman | Row vector       | 0.83 | 183.56 |
|          | Column vector    | 0.59 | 1326.5 |
|          | 4x4 block vector | 0.77 | 389.43 |
|          | 8x8 block vector | 0.66 | 705.77 |
|          | 16 x16 block vector | 0.62 | 1056.3 |
**Table 4.3. Correlation Analysis along the Row, Column and Block Vectors of the Bandeletized Coefficients for Various Test Images**

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>IMAGE BLOCKS</th>
<th>MSE</th>
<th>PSNR (db)</th>
<th>COMPRESSION RATIO</th>
<th>SPACE SAVING (%)</th>
<th>BIT RATE (bpp)</th>
<th>COMPUTATION TIME (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman</td>
<td>Row vector</td>
<td>183.56</td>
<td>25.49</td>
<td>11.14</td>
<td>91.02</td>
<td>1.44</td>
<td>28.50</td>
</tr>
<tr>
<td></td>
<td>Column vector</td>
<td>1326.50</td>
<td>16.90</td>
<td>12.42</td>
<td>91.95</td>
<td>1.28</td>
<td>195.04</td>
</tr>
<tr>
<td></td>
<td>4x4</td>
<td>389.43</td>
<td>22.22</td>
<td>12.27</td>
<td>91.85</td>
<td>1.30</td>
<td>833.62</td>
</tr>
<tr>
<td></td>
<td>8x8</td>
<td>705.77</td>
<td>19.64</td>
<td>11.54</td>
<td>91.34</td>
<td>1.38</td>
<td>194.62</td>
</tr>
<tr>
<td></td>
<td>16 x 16</td>
<td>1056.3</td>
<td>17.89</td>
<td>11.21</td>
<td>91.08</td>
<td>1.42</td>
<td>54.43</td>
</tr>
<tr>
<td>Seismic Survey</td>
<td>Row vector</td>
<td>58.67</td>
<td>30.44</td>
<td>14.60</td>
<td>93.15</td>
<td>1.09</td>
<td>19.95</td>
</tr>
<tr>
<td></td>
<td>Column vector</td>
<td>314.18</td>
<td>23.15</td>
<td>14.66</td>
<td>93.18</td>
<td>1.09</td>
<td>182.40</td>
</tr>
<tr>
<td></td>
<td>4x4</td>
<td>129.99</td>
<td>26.99</td>
<td>14.66</td>
<td>93.18</td>
<td>1.09</td>
<td>847.20</td>
</tr>
<tr>
<td></td>
<td>8x8</td>
<td>276.04</td>
<td>23.72</td>
<td>14.56</td>
<td>93.13</td>
<td>1.09</td>
<td>187.81</td>
</tr>
<tr>
<td></td>
<td>16 x 16</td>
<td>367.18</td>
<td>22.48</td>
<td>14.65</td>
<td>93.17</td>
<td>1.09</td>
<td>52.46</td>
</tr>
<tr>
<td>Seabed</td>
<td>Row vector</td>
<td>35.09</td>
<td>32.67</td>
<td>11.05</td>
<td>90.95</td>
<td>1.44</td>
<td>19.73</td>
</tr>
<tr>
<td>Seismic</td>
<td>Column vector</td>
<td>542.86</td>
<td>20.78</td>
<td>11.20</td>
<td>91.07</td>
<td>1.42</td>
<td>182.42</td>
</tr>
<tr>
<td></td>
<td>4x4</td>
<td>52.86</td>
<td>30.89</td>
<td>11.08</td>
<td>90.97</td>
<td>1.44</td>
<td>1235.96</td>
</tr>
<tr>
<td></td>
<td>8x8</td>
<td>105.97</td>
<td>27.87</td>
<td>11.09</td>
<td>90.98</td>
<td>1.44</td>
<td>194.22</td>
</tr>
<tr>
<td></td>
<td>16 x 16</td>
<td>206.40</td>
<td>24.98</td>
<td>11.09</td>
<td>90.98</td>
<td>1.44</td>
<td>55.51</td>
</tr>
<tr>
<td>Wood and Stock</td>
<td>Row vector</td>
<td>29.13</td>
<td>33.48</td>
<td>11.77</td>
<td>91.50</td>
<td>1.35</td>
<td>20.09</td>
</tr>
<tr>
<td>Textures</td>
<td>Column vector</td>
<td>222.56</td>
<td>24.65</td>
<td>12.06</td>
<td>91.70</td>
<td>1.32</td>
<td>179.00</td>
</tr>
<tr>
<td></td>
<td>4x4</td>
<td>61.93</td>
<td>30.21</td>
<td>11.78</td>
<td>91.51</td>
<td>1.35</td>
<td>758.39</td>
</tr>
<tr>
<td></td>
<td>8x8</td>
<td>130.52</td>
<td>26.97</td>
<td>11.65</td>
<td>91.42</td>
<td>1.37</td>
<td>183.71</td>
</tr>
<tr>
<td></td>
<td>16 x 16</td>
<td>173.52</td>
<td>25.73</td>
<td>11.68</td>
<td>91.44</td>
<td>1.36</td>
<td>52.98</td>
</tr>
<tr>
<td>Fish Texture</td>
<td>Row vector</td>
<td>33.38</td>
<td>32.89</td>
<td>12.62</td>
<td>92.07</td>
<td>1.26</td>
<td>20.36</td>
</tr>
<tr>
<td></td>
<td>Column vector</td>
<td>169.61</td>
<td>25.83</td>
<td>12.48</td>
<td>91.99</td>
<td>1.28</td>
<td>182.80</td>
</tr>
<tr>
<td></td>
<td>4x4</td>
<td>88.48</td>
<td>28.66</td>
<td>12.55</td>
<td>92.03</td>
<td>1.27</td>
<td>896.81</td>
</tr>
<tr>
<td></td>
<td>8x8</td>
<td>223.64</td>
<td>24.63</td>
<td>12.64</td>
<td>92.08</td>
<td>1.26</td>
<td>185.96</td>
</tr>
<tr>
<td></td>
<td>16 x 16</td>
<td>261.89</td>
<td>23.94</td>
<td>12.62</td>
<td>92.08</td>
<td>1.26</td>
<td>53.02</td>
</tr>
<tr>
<td>Geometric</td>
<td>Row vector</td>
<td>23.24</td>
<td>34.46</td>
<td>16.42</td>
<td>93.91</td>
<td>0.97</td>
<td>20.28</td>
</tr>
<tr>
<td>Shapes</td>
<td>Column vector</td>
<td>879.51</td>
<td>18.68</td>
<td>16.24</td>
<td>93.84</td>
<td>0.98</td>
<td>161.01</td>
</tr>
<tr>
<td></td>
<td>4x4</td>
<td>72.34</td>
<td>29.53</td>
<td>16.44</td>
<td>93.92</td>
<td>0.97</td>
<td>689.27</td>
</tr>
<tr>
<td></td>
<td>8x8</td>
<td>186.17</td>
<td>25.43</td>
<td>16.46</td>
<td>93.92</td>
<td>0.97</td>
<td>168.75</td>
</tr>
<tr>
<td></td>
<td>16x16</td>
<td>373.40</td>
<td>22.40</td>
<td>16.44</td>
<td>93.92</td>
<td>0.97</td>
<td>48.48</td>
</tr>
<tr>
<td>Geometric</td>
<td>Row vector</td>
<td>44.49</td>
<td>31.64</td>
<td>15.38</td>
<td>93.50</td>
<td>1.03</td>
<td>20.47</td>
</tr>
<tr>
<td>Pattern</td>
<td>Column vector</td>
<td>954.66</td>
<td>18.33</td>
<td>15.06</td>
<td>93.36</td>
<td>1.06</td>
<td>186.38</td>
</tr>
<tr>
<td></td>
<td>4x4</td>
<td>139.26</td>
<td>26.69</td>
<td>15.54</td>
<td>93.56</td>
<td>1.02</td>
<td>729.03</td>
</tr>
<tr>
<td></td>
<td>8x8</td>
<td>331.99</td>
<td>22.91</td>
<td>15.75</td>
<td>93.65</td>
<td>1.01</td>
<td>182.13</td>
</tr>
<tr>
<td></td>
<td>16x16</td>
<td>1088.9</td>
<td>17.76</td>
<td>16.29</td>
<td>93.86</td>
<td>0.98</td>
<td>51.37</td>
</tr>
</tbody>
</table>
Table 4.3 reveals that processing the row vector takes less computation time than that of the other modes of processing. For cameraman image the processing along the row vector takes about 21.7 sec for the compression and decompression process, whereas column vector processing takes 195.04 sec, block vectors of size 4 x 4 takes 833.62 sec, block vectors of size 8 x 8 takes 194.62 sec and block vectors of size 16 x 16 takes 54.43 sec. Hence row vector processing is done for the subsequent quantization stage which comes after the bandeletization process. The performance of the proposed work is analyzed with $\Omega = 8$; $\tau = 0.4$ and a Scale factor 0. The results are tabulated for various images in Table 4.4. It is perceived from this Table that on an average the proposed work gives a Compression Ratio of 10 leading to 1.5 bits per pixel representation for the compressed file with 90% space saving. Also it maintains the PSNR value to about 28 db.

**Table 4.4 Performance Analysis of the Proposed BVQ for Various Images**

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>IMAGE</th>
<th>MSE</th>
<th>PSNR (db)</th>
<th>COMPRESSION RATIO</th>
<th>SPACE SAVING (%)</th>
<th>BIT RATE (bits/pixel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pepper</td>
<td>96.27</td>
<td>28.30</td>
<td>11.55</td>
<td>91.34</td>
<td>1.39</td>
</tr>
<tr>
<td>2</td>
<td>Baboon</td>
<td>78.50</td>
<td>29.18</td>
<td>10.25</td>
<td>90.24</td>
<td>1.56</td>
</tr>
<tr>
<td>3</td>
<td>Cameraman</td>
<td>183.56</td>
<td>25.49</td>
<td>11.14</td>
<td>91.02</td>
<td>1.44</td>
</tr>
<tr>
<td>4</td>
<td>Rice</td>
<td>70.27</td>
<td>29.66</td>
<td>10.76</td>
<td>90.70</td>
<td>1.49</td>
</tr>
<tr>
<td>5</td>
<td>Lena</td>
<td>118.79</td>
<td>27.38</td>
<td>9.89</td>
<td>89.89</td>
<td>1.62</td>
</tr>
<tr>
<td>6</td>
<td>Bird</td>
<td>56.07</td>
<td>30.64</td>
<td>10.14</td>
<td>90.14</td>
<td>1.58</td>
</tr>
<tr>
<td>7</td>
<td>Barbara</td>
<td>246.48</td>
<td>24.21</td>
<td>9.67</td>
<td>89.66</td>
<td>1.65</td>
</tr>
<tr>
<td>8</td>
<td>Zelta</td>
<td>82.59</td>
<td>28.96</td>
<td>9.78</td>
<td>89.78</td>
<td>1.64</td>
</tr>
</tbody>
</table>
Table 4.5 illustrates the result of comparison of the proposed work with the existing methods for the Barbara image. It is inferred that as compared with the other existing methods the proposed work results in a high Compression Ratio of 9.67 while maintaining the same PSNR value.

4.5 OBSERVATIONS USING WAVELET DECOMPOSITION

This section is meant to show the performance of this scheme in the wavelet domain. Table 4.6 shows the results of this work for various wavelet functions for the 256 x 256 Lena image. It is observed that the proposed BVQ in the Wavelet domain (WBVQ) results in a very high compression ratio with a small reduction in the reconstructed image quality. Table 4.4 reports that 27.38db quality factor with a compression ratio of 9.89 using the BVQ coder for the Lena image; whereas for the same Lena image in the wavelet domain the quality factor is reduced to 25.58db with an increased compression ratio of 35.46 (Table 4.6).
**Table 4.6 Performance Analysis of the Proposed BVQ Using Various Wavelet Functions for the 256 x 256 Lena Image**

<table>
<thead>
<tr>
<th>Wavelet Family</th>
<th>Wavelet Function</th>
<th>MSE</th>
<th>PSNR (db)</th>
<th>Compression Ratio</th>
<th>Space Saving (%)</th>
<th>Bit Rate (bpp)</th>
<th>Computation Time (Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daubechies</td>
<td>db1/Haar</td>
<td>179.73</td>
<td>25.58</td>
<td>35.46</td>
<td>97.18</td>
<td>0.45</td>
<td>7.23</td>
</tr>
<tr>
<td></td>
<td>db2</td>
<td>183.19</td>
<td>25.50</td>
<td>31.44</td>
<td>96.82</td>
<td>0.50</td>
<td>9.47</td>
</tr>
<tr>
<td></td>
<td>db3</td>
<td>195.81</td>
<td>25.21</td>
<td>31.40</td>
<td>96.81</td>
<td>0.50</td>
<td>6.77</td>
</tr>
<tr>
<td></td>
<td>db4</td>
<td>194.90</td>
<td>25.23</td>
<td>31.87</td>
<td>96.86</td>
<td>0.50</td>
<td>8.28</td>
</tr>
<tr>
<td></td>
<td>db5</td>
<td>208.55</td>
<td>24.93</td>
<td>32.42</td>
<td>96.91</td>
<td>0.49</td>
<td>8.23</td>
</tr>
<tr>
<td></td>
<td>db10</td>
<td>212.56</td>
<td>24.85</td>
<td>29.01</td>
<td>96.55</td>
<td>0.55</td>
<td>10.29</td>
</tr>
<tr>
<td></td>
<td>db15</td>
<td>212.85</td>
<td>24.84</td>
<td>30.21</td>
<td>96.69</td>
<td>0.52</td>
<td>11.36</td>
</tr>
<tr>
<td></td>
<td>db20</td>
<td>215.21</td>
<td>24.80</td>
<td>27.92</td>
<td>96.41</td>
<td>0.57</td>
<td>10.69</td>
</tr>
<tr>
<td></td>
<td>db25</td>
<td>222.47</td>
<td>24.65</td>
<td>28.44</td>
<td>96.48</td>
<td>0.56</td>
<td>11.19</td>
</tr>
<tr>
<td></td>
<td>db30</td>
<td>221.33</td>
<td>24.68</td>
<td>25.96</td>
<td>96.14</td>
<td>0.61</td>
<td>12.26</td>
</tr>
<tr>
<td></td>
<td>db35</td>
<td>226.29</td>
<td>24.58</td>
<td>23.97</td>
<td>95.82</td>
<td>0.66</td>
<td>15.90</td>
</tr>
<tr>
<td></td>
<td>db40</td>
<td>223.48</td>
<td>24.63</td>
<td>23.96</td>
<td>95.82</td>
<td>0.66</td>
<td>11.51</td>
</tr>
<tr>
<td>Coiflets</td>
<td>coif1</td>
<td>208.69</td>
<td>24.93</td>
<td>32.03</td>
<td>96.87</td>
<td>0.49</td>
<td>8.30</td>
</tr>
<tr>
<td></td>
<td>coif2</td>
<td>192.61</td>
<td>25.28</td>
<td>32.17</td>
<td>96.89</td>
<td>0.49</td>
<td>9.66</td>
</tr>
<tr>
<td></td>
<td>coif3</td>
<td>195.21</td>
<td>25.22</td>
<td>32.22</td>
<td>96.89</td>
<td>0.49</td>
<td>7.99</td>
</tr>
<tr>
<td></td>
<td>coif4</td>
<td>195.83</td>
<td>25.21</td>
<td>29.07</td>
<td>96.56</td>
<td>0.55</td>
<td>11.75</td>
</tr>
<tr>
<td></td>
<td>coif5</td>
<td>205.61</td>
<td>25.00</td>
<td>29.16</td>
<td>96.57</td>
<td>0.55</td>
<td>9.02</td>
</tr>
<tr>
<td>Symlets</td>
<td>sym2</td>
<td>183.19</td>
<td>25.50</td>
<td>31.44</td>
<td>96.82</td>
<td>0.50</td>
<td>8.72</td>
</tr>
<tr>
<td></td>
<td>sym5</td>
<td>191.17</td>
<td>25.31</td>
<td>31.43</td>
<td>96.81</td>
<td>0.50</td>
<td>8.72</td>
</tr>
<tr>
<td></td>
<td>sym10</td>
<td>203.50</td>
<td>25.04</td>
<td>29.32</td>
<td>96.58</td>
<td>0.54</td>
<td>9.77</td>
</tr>
<tr>
<td></td>
<td>sym15</td>
<td>209.55</td>
<td>24.91</td>
<td>29.11</td>
<td>96.56</td>
<td>0.54</td>
<td>11.14</td>
</tr>
<tr>
<td></td>
<td>sym20</td>
<td>197.33</td>
<td>25.17</td>
<td>25.95</td>
<td>96.14</td>
<td>0.61</td>
<td>11.31</td>
</tr>
<tr>
<td></td>
<td>sym25</td>
<td>200.45</td>
<td>25.11</td>
<td>25.87</td>
<td>96.13</td>
<td>0.61</td>
<td>15.23</td>
</tr>
<tr>
<td></td>
<td>sym30</td>
<td>197.78</td>
<td>25.16</td>
<td>23.58</td>
<td>95.75</td>
<td>0.67</td>
<td>43.92</td>
</tr>
<tr>
<td></td>
<td>sym35</td>
<td>211.63</td>
<td>24.87</td>
<td>21.94</td>
<td>95.44</td>
<td>0.72</td>
<td>159.74</td>
</tr>
<tr>
<td>Discrete Meyer</td>
<td>dmey</td>
<td>203.95</td>
<td>25.03</td>
<td>18.59</td>
<td>94.62</td>
<td>0.86</td>
<td>17.15</td>
</tr>
<tr>
<td>Biorthogonal</td>
<td>bior1.1, rbio1.1</td>
<td>179.73</td>
<td>25.58</td>
<td>35.46</td>
<td>97.18</td>
<td>0.45</td>
<td>7.50</td>
</tr>
<tr>
<td></td>
<td>bior2.2, rbio2.2</td>
<td>401.72</td>
<td>22.09</td>
<td>32.78</td>
<td>96.94</td>
<td>0.48</td>
<td>8.50</td>
</tr>
<tr>
<td></td>
<td>bior3.3, rbio3.3</td>
<td>610.14</td>
<td>20.27</td>
<td>32.09</td>
<td>96.88</td>
<td>0.49</td>
<td>7.25</td>
</tr>
<tr>
<td></td>
<td>bior4.4, rbio4.4</td>
<td>203.32</td>
<td>25.04</td>
<td>31.95</td>
<td>96.87</td>
<td>0.50</td>
<td>8.82</td>
</tr>
<tr>
<td></td>
<td>bior5.5, rbio5.5</td>
<td>168.70</td>
<td>25.85</td>
<td>31.04</td>
<td>96.77</td>
<td>0.51</td>
<td>10.50</td>
</tr>
</tbody>
</table>
The proposed work is tested with various wavelet functions and Table 4.6 presents the observations. It is observed that the Db1 wavelet function from the Daubechies family and the Biorthogonal wavelet function version 1.1 gives better results than the other wavelet functions. For further analysis, the Db1 version is used as it takes less computation time.

4.6 CONCLUSION

A novel anisotropic transform for images that use separable filtering in many directions is proposed in this work. The bandeletization process is reversible and introduces redundancies. At the same time, the compression algorithm obtained as a combination of Bandelets, Quantization and Coding outperforms the state-of-the-art methods in terms of both the numerical criterion and the visual quality.
5. NEURO-STATISTICAL QUANTIZER DESIGN FOR IMAGE COMPRESSION

5.1 INTRODUCTION
This chapter converses about the compression of the gray scale images using a statistical approach for modeling the code vectors designed using a competitive learning neural network. Traditional methods are compared to a new method, based on neuro-statistical modeling taking into account the psycho visual features in the images in the spatial domain. This method uses competitive learning algorithms and Savitzky–Golay polynomial in the compression process. This method is adapted to design the codebook for the transform coefficients of each pixel. This approach exploits the psycho-visual as well as statistical redundancies in the image data, enabling improved quality and bit rate reduction. Two crucial issues in compression methods are the coding efficiency and the psycho visual quality achieved while modeling different image regions. This chapter presents a high performance wavelet coder which provides a new framework for handling these issues in a simple and effective manner.

In order to facilitate the understanding of the research results presented in this chapter, this section explains some of the fundamental concepts relevant to this work. Image compression is radically different from data compression. Logically, general purpose data compression algorithms can be used to compress images, but the result is less than optimal. This is because some of the finer details in the image can be forfeited for the sake of saving a little more bandwidth or storage space. In addition, images possess
certain statistical properties which can be exploited by the exclusively designed encoders. This chapter deals with the design of one such encoder called Vector Quantizer both in the spatial and transform domains.

Vector Quantization provides a means of converting the decomposed signal into bits in a manner that takes advantage of remaining inter and intra-band correlation. It is a relatively new coding technique that has aroused wide interest [166] when applied to image coding. VQ provides many attractive features in applications where high compression ratios are desired. Linde, Buzo and Gray known as the LBG algorithm is the well known iterative technique for the codebook design phase in the VQ process. This algorithm is conceptually simple but involves computational complexity and requires an initial codebook. The performance of this algorithm is based on the selection of this initial codebook. A Modified Generalized Lloyd algorithm is used to overcome this problem identifies the unused code vector and replaces it with a training vector with the highest distortion. The Enhanced Generalized Lloyd Algorithm using Pattern Reduction reduces the computational complexity by using the pattern reduction approach. The quality of the reconstructed image is not affected because the size of the codebook is not reduced. Fast Codebook Generation algorithm reduces the convergence time in order to optimize the vector quantized codebook. The algorithm generates unique minima for each code vector. As a result it takes less number of iterations as compared to LBG codebook.

Neural networks seem to serve the purpose of initializing a codebook for the VQ process. Self-Organizing Map based VQ algorithm performs better than the early VQ techniques. A unified approach that combines SOM
and stochastic gradient techniques promises good quality reconstructed images. Self-Organizing Map is also used to construct a generic codebook. Further cubic surface fitting technique is applied on the code vectors of this codebook in order to enhance the codebook. The aim of generic codebook design method is to reduce the compression bit rate. It is reported that the algorithm performs better in the spatial domain than in the transform domain. Amerijckx et. al. [167] applied SOM based VQ technique in a discrete cosine transform domain. This method improves the compression ratio significantly. The Enhanced Self-Organizing Map algorithm [168] for medical image compression in the wavelet domain performs the weight updation in two ways: based on the frequency of the winner node and based on the ratio of the present change in weight to the previous change in weight. This enhanced SOM algorithm is used in the codebook generation phase of VQ. In VQ, searching the codebook to find the best matching code vector for an input vector is time consuming. To reduce this computational overhead a fast codebook searching method for a Self-Organizing Map based vector quantizer is proposed in this work. A non-exhaustive search method is used to find a matching code vector from the codebook for an input vector instead of the exhaustive search in a large codebook with high dimensional vectors.

This chapter presents a Neuro-Statistical Quantization (NSQ) scheme to design vector quantizer codebook for image compression using Kohonen’s competitive learning rule based neural network and Savitzky-Golay polynomial modeling. An initial codebook is generated by training a neural network with the training vectors using the competitive learning rule. Then, to achieve better psycho-visual fidelity, each code vector is replaced with
minimum distortion polynomial coefficient generated by statistical modeling process. The set of code indices produced by the quantizer is further compressed using Huffman encoder. This technique exploits the psycho-visual as well as statistical redundancies in the image data, enabling bit rate reduction.

5.2 VECTOR QUANTIZATION

Vector Quantization is an effective technique for performing image compression. VQ is always better than scalar quantization because it fully exploits the correlation between components within the vector. Also, its decoding procedure is very simple since it only consists of table lookups. The density matching property of vector quantization is powerful, especially for identifying the density of large and high dimensional data. Since data points are represented by the index of their closest centroid frequently occurring data will have lower error. For this reason VQ is suitable for lossy data compression. This research work uses VQ to encode the wavelet coefficients.

The principle of VQ is defined as follows:

Let $C = \{c_i, \ i = 1,2,...,n\}$ be a codebook of size $n$, where $c_i = \{c_{i1}, c_{i2}, ..., c_{ik}\}$ is a $k$-dimensional code vector. For a given input vector $x = \{x_{i1}, x_{i2}, ..., x_{ik}\}$, find the code vector $y(x)$ which is almost similar or closest (in some sense) to $x$. The distance between $x$ and a code vector $c_i$ is denoted by $d(x, c_i)$.

The basic VQ process is shown in Figure 5.1. First a codebook is generated using the input vector $x$. Then a minimum distortion code vector’s index is assigned to each input vector using nearest neighbor rule.
These indices are transmitted over the channel. At the receiver, these indices are replaced by the corresponding code vector using the codebook as the look up table.

**Competitive Learning Neural Networks for Codebook Design**

Design of an initial codebook is a crucial step in the VQ process because image quality is the result of the comparison between training vectors with codebook. The proposed work employs competitive learning neural networks [169] for the initial codebook design. The mechanism by which only one unit is chosen to respond is called competition. The mostly used competition among group of neurons is Winner-Takes-All. Hence, only one neuron in the competing group will have a non-zero output signal when the competition is completed. This section discusses three of those Neural Networks relevant to this research. They are the Competitive Learning Neural Networks, Self-Organizing Map and Learning Vector Quantization.
The architecture of these nets is similar. It is given in Figure 5.2. The network consists of two layers namely the input layer and the competitive layer. The competitive layer is assumed to be one dimensional for the CPL and LVQ networks and two dimensional for the SOM network. The network is fully connected (i.e.) all nodes in input layer are connected to all nodes in competitive layer.

The input vector is presented at the input layer and propagated to the competitive layer. They differ only in the learning process. These nets use Kohonen’s Learning Approach. In this learning, the units update their weights by forming a new weight vector that is a linear combination of the old weight vector and the current input vector. The unit whose weight vector is closest to the input vector is allowed to learn. Euclidean distance measure is used to determine the neuron in the competitive layer to which the input vector is presented.
vector is the closest. This neuron is called the winning neuron or the firing neuron.

Algorithm for the neural network learning using competitive learning rule:

1. Initialization:
   
   Step 1: Initialize weights and Initialize learning rate.
   
   Step 2: Initialize target vector \( t = \{t_1, t_2, \ldots, t_n\} \) (for LVQ only)

2. Winning neuron detection:
   
   Step 3: For each input vector \( X = \{x_1, x_2, \ldots, x_m\} \), do steps 4-6
   
   Step 4: For each competitive unit \( y_j \), compute squared Euclidean distance
   
   \[ D(j) = \sum (w_{ij} - x_i)^2; \quad i = 1, \ldots, m \text{ and } j = 1, 2, \ldots, n \]  
   
   (5.1)

   Step 5: Find the winning neuron \( j \) with minimum distance \( D(j) \)

3. Weight updation:

   Step 6:
   
   i) For CPL net:
   
   Update the weights of the winning neuron.
   
   \[ w_{ij(new)} = w_{ij(old)} + \alpha (x_i - w_{ij(old)}) \]  
   
   (5.2)

   ii) For SOM net:

   Determine the neighbors of \( r \) using 8-neighborhood rule
   
   (Competitive units that lie within a radius \( j \) of the winning neuron)

   Update the weights of the winning neuron and its neighbors.
   
   \[ w_{ij(new)} = w_{ij(old)} + \alpha n(j,k)(x_i - w_{ij(old)}) \]  
   
   (5.3)
iii) For LVQ net:

If \( t_j = y_j \), then
\[
\begin{align*}
    w_{ij}^{(new)} &= w_{ij}^{(old)} + \alpha \left( x_i - w_{ij}^{(old)} \right) \\
    (5.4)
\end{align*}
\]

If \( t_j \neq y_j \), then
\[
\begin{align*}
    w_{ij}^{(new)} &= w_{ij}^{(old)} - \alpha \left( x_i - w_{ij}^{(old)} \right) \\
    (5.5)
\end{align*}
\]

Where \( \alpha \) is the learning parameter and \( n(j,k) \) is called the neighborhood function that has value 1 when \( j = k \) and falls off with the radial distance \( r \) between units in the competitive layer. The competitive layer can be used to find cluster centers in data. As it can find clusters in data, it can effectively compress data and extract relevant features from it, removing redundant information. The weight vectors are moved in the direction of the input vectors, thus finding clusters in the data.

Once the network is trained, the codebook can readily be designed using the trained weight vectors as the code vectors. Images can be quantized by finding out the closest code vector for each input image vector. However, vector quantizers produce some checker-board pattern in the reconstructed image. Even though the reconstructed image shows quite good PSNR, this effect often had some adverse psycho-visual impact. The proposed method adopts a scheme of polynomial rendering to modify the code vectors generated by the competitive learning neural network that reduces the checker-board pattern in the reconstructed image and improves its psycho-visual quality. The computational overhead occurs only at the codebook design stage and not during encoding or decoding of each image.
The Savitzky-Golay method [170] essentially performs a local polynomial regression of degree $k$ on a series of equally spaced data values $f(x, y)$ of at least $k + 1$ points which are treated as being equally spaced in the series to determine the smoothed value for each point.

For an image of size $M \times N$, polynomial regression replaces each data value $f(x, y)$ by a linear combination $g(x, y)$ of itself and some number of nearby neighbors,

$$g(x, y) = \sum_{a=-a}^{a} \sum_{t=-a}^{a} c(s, t) f(x + s, y + t) \quad (5.6)$$

Here, $a = (m - 1)/2$, $x = 0, 1, 2, ..., M - 1$, $y = 0, 1, 2, ..., N - 1$, $m = 7$ and $c(s, t)$ is the mask co-efficient. Since $m$ is 7, the value of $a$ is 3. The idea of Savitzky-Golay method is to find mask coefficients $c(s, t)$ that preserve higher moments. These coefficients are calculated through the process of polynomial least-squares fitting inside a moving window.

The calculation part is as follows:

Let, $k$ be the degree of the polynomial; $m$ be the number of points to be fitted by the polynomial; $F$ be the array of actual data points $f(x + s, y + t)$;

The polynomial expression given below is used to fit the data.

$$F(i) = f(x_i, y_i) = c(0,0) + c(1,0)x_i + c(0,1)y_i + c(2,0)x_i^2 + c(1,1)x_iy_i + c(0,2)y_i^2 + \cdots + c(0, k)y_i^k \quad (5.7)$$

where, $k = 5$; the coefficient of $(x_i, y_i)$ is $c(i, i)$ and $(x_i, y_i)$ is the pixel coordinate of pixel $F(i)$. Next step is to fit a polynomial of type in Eqn. (5.7), to the data. By solving the least squares, the polynomial coefficients can be found. Eqn. (5.8) is used to solve the least squares.

$$F = Xc \quad (5.8)$$
where, \( X \) is defined by the following matrix:

\[
\begin{bmatrix}
1 & x_0 & y_0 & x_0^2 & y_0^2 & x_0^3 & y_0^3 & \cdots & y_0^k \\
1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1^3 & y_1^3 & \cdots & y_1^k \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_0 & y_0 & x_0y_0 & y_0^2 & x_0y_0^2 & y_0^3 & \cdots & y_0^k
\end{bmatrix}
\tag{5.9}
\]

\( c \) is the vector of polynomial coefficients given by,

\[
c = \left( c(0,0) c(1,0) c(0,1) c(2,0) c(1,1) c(0,2) \cdots c(0,k) \right)^T
\tag{5.10}
\]

and \( F \) represents \( m \times m \) block image data, i.e.

\[
F = \left( F(0) F(1) F(2) \cdots F(m^2) \right)^T
\tag{5.11}
\]

Rewriting Eqn. (5.8) gives us,

\[
c = PF
\tag{5.12}
\]

where, \( P = (X^TX)^{-1}X^T \) is the pseudo inverse of \( X \), and it is independent of the image data.

Eqn. (5.12) simply reproduces the polynomial for each pixel in the image patch. Each polynomial coefficient is computed as the inner product of one row of \( P \) and the column of pixel values \( F \). This is the surprising part about Savitzky-Golay model: the polynomial coefficients are computed using a linear filter on the data. Just as one can reassemble \( F \) back into a rectangular patch of pixels, one can also assemble each row of \( P \) into the same size rectangle to get a traditional looking image filter.

The advantage of the Savitzky-Golay model is its ability to preserve higher moments in the data and thus reduce smoothing on peak heights. The coefficient vector \( a \) is calculated for all code vectors obtained from
competitive Learning network. Once the coefficient vector is available, the codebook can be designed by reconstructing the code vectors.

The effectiveness of this method is tested using several images. Although the quality improvement in terms of PSNR appears marginal, significant improvement in terms of psycho-visual quality is observed consistently.

5.3 THE PROPOSED SCHEME FOR IMAGE COMPRESSION

This section discusses the design of the proposed quantizer and the role of this quantizer in a wavelet based image compression system.

5.3.1 Proposed Neuro-Statistical Quantizer Design Procedure

The diagram shown in Figure 5.3 depicts the proposed Neuro-Statistical Quantizer design process. The code vectors are generated using a neural net and are refined through polynomial regression. This process has the following stages:

5.3.1.1 Vector Formation

The input image is divided into sub-blocks of equal sizes. This is called vector formation process.

5.3.1.2 Code vector Generation

In this process the codebook is initialized with few of the input image vectors. The codebook whose size is equal to one third of the size of the input image is initialized randomly from this image vector.
5.3.1.3 Code vector Training using Neural Network

Neural network algorithm is used for training the code vectors in the codebook. The initial code vectors are used as the initial weight vectors for the neural network. The input image vectors are used as the input to this network. The training is performed in an iterative fashion to modify the weights so that the weight vectors approximate the distribution of the training vector and also preserve topology of input data on the viewing plane. The training process is explained in section 5.2. Finally, the trained weight vectors constitute the code vectors of the codebook necessary for the VQ.

5.3.1.4 Code vector Modeling using Polynomial Regression

Since the quality of the reconstructed image is based on the comparison between original images with Codebook, conserving the traits of the code vectors becomes indispensable. It is proposed to use polynomial regression for modeling the codebook in order to preserve the traits of the code vectors. The code vectors of the codebook are divided into subgroups of size 7. Then each point in the subgroup is replaced by the value that is obtained by the process of polynomial least-squares fitting inside that subgroup.

5.3.1.5 Minimum Distortion Index Generation

Euclidean distance measure is used as a nearest neighbor rule to find the distortion error between each input vector to this stage and the code vectors. An index vector consisting of the indices of the minimum distortion code vector corresponding to each input vector is generated.
5.3.2 Proposed Architecture

This section explains the detailed design procedure of the proposed architecture shown in Figure 5.4. The architecture consists of two parts: the Compression stage (Coder) and the Decompression stage (Decoder).

5.3.2.1 Compression Stage

The compression stage includes a transformation stage for multi-resolution compression, error-resilient streaming through quantization and a lossless encoder.

5.3.2.1.1 Wavelet Transform

The wavelet transform (db4) decomposes the original image \(i\) into different resolution sub-bands. At each decomposition level, four sub-bands are produced. They are named approximation sub-band (low-pass version)
and detail sub-bands (vertical, horizontal, and diagonal). The transformed coefficients \( t = T(i) \) are subject to the vector quantization process.

**5.3.2.1.2 Vector Quantization**

The coefficients are grouped into sub-blocks of equal sizes. Then an initial codebook is generated and the code vectors are trained and modeled using neural nets and polynomial regression respectively. Finally, for each sub-block the index of the matching code vector is assigned. This constitutes the index vector \( q = Q(t) \). The detailed process is explained above in Section 5.3.1.

**5.3.2.2 Decompression Stage**

The decompression stage comprises the Huffman decoder \( q' = C^{-1}(c') \), index to code vector re-assignment \( t' = Q^{-1}(q') \) followed by the inverse transformation \( t' = T^{-1}(t') \).

**Computational Complexity**

Computational complexity can be investigated on the basis of time, memory or other resources used to solve the problem. Time and space are two of the most important and popular considerations when problems of complexity are analyzed. Since wavelets are localized in both time and frequency they give a better signal representation using multiresolution analysis, with balanced resolution at any time and frequency. Their computational complexity is \( O(MN) \). This computational advantage is not inherent to the transform, but reflects the choice of a logarithmic division of frequency.
Figure 5.4 Proposed Image Compression Scheme
To design a codebook of size $m$, the computational complexity of LVQ is $O(TMNm)$, where $T$ is the number of iterations and $MN$ is the total number of training vectors. The computational complexity for the Savitzky Polynomial method $O(n \log n)$, where $n = 64 \times 128$ is the size of the codebook. The computational complexity of the Huffman algorithm is $O(n \log n)$. Using a heap to store the weight of each tree, the iteration requires $O(\log n)$ time to determine the cheapest weight and insert the new weight. There are $O(n)$ iterations, one for each item. Therefore the computational complexity of the proposed work is the sum of $O(N)$, $O(TNm)$, $O(n \log n)$ and $O(n \log n)$. It shall be noted from this analysis that the proposed work incurs a computational overhead of $O(n \log n)$. This little overhead may be ignored because the quality factor and the compression rate are the important criteria for compression.

5.4 IMPLEMENTATION RESULTS

This section presents experiments in which previously derived results are applied to a number of images. It will be shown that the application of this method improves the visual quality of the reconstructed image. The performance of the proposed system is measured using two mathematical metrics. One of them is the MSE, which measures the cumulative squared error between the original and the reconstructed image. The other is the peak signal-to noise ratio measure, known as PSNR. Section 5.4.1 reports the experimental results among the existing VQ techniques and the observations of the proposed work in the spatial domain. Section 5.4.2 presents detailed analysis of this work and the experimental results in the wavelet domain.
5.4.1 Spatial Domain Approach

The first experiment considers the Baboon image (Figure 5.5a) and the lighthouse image of size 256 x 256. Experiments are carried out to compress these images using the generalized Neural based Vector Quantization (NVQ), Multi Stage Vector Quantization (MSVQ), Pyramid Vector Quantization (PVQ), Tree Structured Vector Quantization (TSVQ) and the Linde-Buzo and Gray (LBG) algorithm vector quantization techniques. The next experiment is carried out to illustrate the effective performance of the proposed method in the spatial domain. Table 5.1 and Table 5.2 show the respective results.

**Table 5.1 Performance Comparison of the Existing VQ Techniques for a Compression Ratio of 2.9**

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>IMAGE</th>
<th>ALGORITHM</th>
<th>MSE</th>
<th>PSNR (db)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baboon</td>
<td>NVQ</td>
<td>143.64</td>
<td>26.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSVQ</td>
<td>252.03</td>
<td>25.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PVQ</td>
<td>221.64</td>
<td>24.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSVQ</td>
<td>395.62</td>
<td>22.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LBG</td>
<td>438.22</td>
<td>21.79</td>
</tr>
<tr>
<td>2</td>
<td>Lighthouse</td>
<td>NVQ</td>
<td>238.84</td>
<td>24.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MSVQ</td>
<td>309.00</td>
<td>23.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PVQ</td>
<td>487.89</td>
<td>21.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSVQ</td>
<td>630.50</td>
<td>20.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LBG</td>
<td>570.52</td>
<td>20.56</td>
</tr>
</tbody>
</table>
Table 5.1 shows the comparison among the existing generalized VQ techniques for image compression and hence reveals the reason for choosing the neural network platform for this research. The table compares the reconstructed image quality in terms of MSE and PSNR at a compression ratio of 2.9 for the NVQ, MSVQ, PVQ, TSVQ and the LBG vector quantization techniques. It shows that the NVQ technique performs better than the other reported techniques. Therefore, the proposed work uses the NVQ technique as its base. Table 5.2 shows the result of comparison of the proposed work with the NVQ technique in the spatial domain. It is practically proved that the

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>IMAGE</th>
<th>ALGORITHM</th>
<th>MSE</th>
<th>PSNR (db)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baboon</td>
<td>NSQ</td>
<td>103.21</td>
<td>27.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVQ</td>
<td>143.64</td>
<td>26.55</td>
</tr>
<tr>
<td>2</td>
<td>Lighthouse</td>
<td>NSQ</td>
<td>235.62</td>
<td>24.41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVQ</td>
<td>238.84</td>
<td>24.34</td>
</tr>
<tr>
<td>3</td>
<td>Zelta</td>
<td>NSQ</td>
<td>127.03</td>
<td>27.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVQ</td>
<td>212.04</td>
<td>24.86</td>
</tr>
<tr>
<td>4</td>
<td>Bird</td>
<td>NSQ</td>
<td>81.61</td>
<td>29.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVQ</td>
<td>176.05</td>
<td>25.67</td>
</tr>
<tr>
<td>5</td>
<td>Boat</td>
<td>NSQ</td>
<td>300.00</td>
<td>23.36</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVQ</td>
<td>487.89</td>
<td>21.24</td>
</tr>
<tr>
<td>6</td>
<td>Boy</td>
<td>NSQ</td>
<td>117.86</td>
<td>27.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVQ</td>
<td>337.42</td>
<td>22.84</td>
</tr>
<tr>
<td>7</td>
<td>Lion</td>
<td>NSQ</td>
<td>97.92</td>
<td>28.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVQ</td>
<td>132.72</td>
<td>26.90</td>
</tr>
</tbody>
</table>
The proposed work yields superior reconstructed psycho-visual image quality than the NVQ technique in terms of better MSE and PSNR while maintaining same compression ratio. The proposed technique gives compression ratio to about 3, above 66% space saving and compression rate in the order of 2.6 bits/pixel in the spatial domain.

Figure 5.5  Original Image and Respective Reconstructed Images obtained using the Proposed and Existing Methods: (a)  256 x 256 Original Image, (b) Reconstructed Image using the Proposed NSQ, (c) Reconstructed Image using NVQ
Figure 5.5a shows 256 x 256 Baboon input image. Figure 5.5b shows respective reconstructed image using the proposed NSQ codebook design and Figure 5.5c depicts the same using the existing NVQ codebook design respectively at a compression ratio of 2.9. According to human visual system it is evident that the psycho-visual quality is rated ‘good’ for the proposed work, ‘bad’ for the NVQ method. It is obvious that the proposed work reduces blocking effect thereby improving the psycho-visual quality.

5.4.2 Wavelet Domain Approach
This section presents the detailed analysis of the performance of the proposed work in the wavelet domain. In Section 5.4.1, it is reported that the NVQ outperforms other VQ techniques. Hence the experimental analysis in the wavelet domain uses NVQ for the quantization process. Further, it is mentioned in Section 5.2 that there are three neural nets namely CPL, SOM and LVQ available using the competitive learning rule. Hence the proposed polynomial regression analysis is carried out using these three nets resulting in three new schemes namely the WNSQ I (Wavelet based Neuro Statistical Quantizer using LVQ), the WNSQ II (Wavelet based Neuro Statistical Quantizer using SOM) and the WNSQ III (Wavelet based Neuro Statistical Quantizer using CPL). This section presents the observations for the proposed work using LVQ (i.e.) the WNSQ I. A comparative analysis between these three schemes is also provided at the end of this chapter. First experiments are carried out to illustrate the effects of the choice of the order of the polynomial for modeling and the size of the image for compression. Next, both the subjective and objective performance measures are evaluated.
The choice of a suitable polynomial order for the proposed work is based on the graphs shown in Figure 5.6. The graph exhibits the impact of polynomial order on the bit rate and the quality of the reconstructed image. It is observed from Figure 5.6a. that the bit rate is not influenced by the application of the modeling process; for some images the change in the polynomial order does not affect the bit rate. For certain other images there is a negligible gain in the bit rate for the second / third order polynomial. For the Rose and Bird images this gain is 0.04bpp and 0.05bpp respectively.

The graph showed in Figure 5.6b. reveals that the gain in PSNR for the Rose and Bird images are 6.07db and 6.58db respectively at the fourth order polynomial as compared to the third order polynomial. In addition, from Figure 5.6b., it is clear that the PSNR increases with the increase in the polynomial order until fourth order and the value decreases for polynomials of order greater than 5. Therefore, fourth order polynomial is used in the proposed work.

The graphs showed in Figure 5.7. depict the impact of the proposed work on various performance measures by varying the size of the original image. It is observed from Figure 5.7a. that the Bit Rate decreases with the increase in image size. And the PSNR increases with the increase in image size (Figure 5.7b) as desired. But the computation time is increased gradually as the image size is varied from 64 x 64 to 256 x 256 (Table 5.3). And for the 512 x 512 size images, the computation time is increased drastically which is an undesirable characteristic. Therefore images of size 256 x 256 are used for the analysis of the proposed work.
Figure 5.6  The Impact of Polynomial Order on Bit Rate and Image Quality:  
(a) Polynomial Order Vs Bit Rate and  
(b) Polynomial Order Vs PSNR.

Figure 5.7  The Impact of Image Size on Various Performance Measures  
for Various Images:  (a) Image size Vs Bit Rate and  
(b) Image Size Vs PSNR.
Table 5.3 The Impact of Image Size on Computation Time (in Sec.)

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>SIZE</th>
<th>64 X 64</th>
<th>128 X 128</th>
<th>256 X 256</th>
<th>512 X 512</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zelta</td>
<td>0.29</td>
<td>0.32</td>
<td>0.78</td>
<td>3.95</td>
<td></td>
</tr>
<tr>
<td>Rose</td>
<td>0.30</td>
<td>0.37</td>
<td>0.81</td>
<td>3.81</td>
<td></td>
</tr>
<tr>
<td>Bird</td>
<td>0.26</td>
<td>0.33</td>
<td>0.77</td>
<td>3.77</td>
<td></td>
</tr>
<tr>
<td>Cereal</td>
<td>0.15</td>
<td>0.24</td>
<td>0.69</td>
<td>3.69</td>
<td></td>
</tr>
</tbody>
</table>

The reconstructed image quality can be measured both subjectively and objectively. Figure 5.8 is used to give the subjective psycho-visual quality. Figure 5.8a shows the original Lena image. Figure 5.8d and Figure 5.8e show the respective reconstructed image using the LVQ based proposed work with (WNSQ I) and without statistical modeling (Wavelet LVQ). To demonstrate the effect more clearly Figure 5.8b and Figure 5.8c shows the enlarged right eye portion of the images shown in Figure 5.8d and Figure 5.8e respectively. Statistical modeling of the LVQ code vectors reduces the blurring effect thereby improving the visual quality. Figure 5.8f and Figure 5.8g show the respective reconstructed images using the SOM based proposed work with (WNSQ II) and without statistical modeling (Wavelet SOM). Figure 5.8h and Figure 5.8i show the respective reconstructed image using the CPL based proposed work with (WNSQ III) and without statistical modeling (Wavelet CPL). These reconstructed images have a quality factor ≈ 24.5db.
Figure 5.8  Original and Reconstructed Images using Various Neural Networks Illustrating the Effect of Statistical Modeling: (a) 256 x 256 Original Lena Image, (b) Enlarged Eye Portion of the Original Image, (c) Enlarged Eye Portion of the Proposed WNSQ 1 Scheme; Reconstructed Images: (d) WNSQ I, (e) Wavelet LVQ, (f) WNSQ II, (g) Wavelet SOM, (h) WNSQ III, (i) Wavelet CPL
Though the reconstructed images have almost similar objective quality factor, their subjective quality is quite different. The images used are sampled 256 x 256 black and white images. The intensity of each pixel is coded on 256 gray levels (8bpp). It is evident from the images shown in Figure 5.8 that, the image obtained by the proposed algorithm is quite superior to the resultant image of the existing algorithm in terms of psycho visual quality.

The objective performance measures like PSNR, Bit Rate etc., for various images including standard images and other natural images are computed. In order to highlight the contribution made by the proposed work a comparison between the performance of proposed Polynomial Regression based LVQ codebook design method (WNSQ I) and the generalized LVQ codebook design method (Wavelet-LVQ) is given in Table 5.4.

**Table 5.4 Performance Comparison between the Proposed WNSQ I (with modeling) and the Wavelet-LVQ (without modeling)**

<table>
<thead>
<tr>
<th>S.NO</th>
<th>IMAGE</th>
<th>ALGORITHM</th>
<th>MSE</th>
<th>PSNR (db)</th>
<th>COMPUTATION TIME (sec)</th>
<th>COMPRESSION RATIO</th>
<th>SPACE SAVING (%)</th>
<th>BIT RATE (bpp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baboon</td>
<td>WNSQ I</td>
<td>161.23</td>
<td>26.05</td>
<td>0.791</td>
<td>13.31</td>
<td>92.48</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wavelet-LVQ</td>
<td>186.26</td>
<td>25.42</td>
<td>0.772</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Cereal</td>
<td>WNSQ I</td>
<td>25.52</td>
<td>34.06</td>
<td>0.670</td>
<td>16.39</td>
<td>93.90</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wavelet-LVQ</td>
<td>27.94</td>
<td>33.67</td>
<td>0.679</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Pepper</td>
<td>WNSQ I</td>
<td>204.92</td>
<td>25.01</td>
<td>0.844</td>
<td>13.27</td>
<td>92.47</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wavelet-LVQ</td>
<td>209.56</td>
<td>24.92</td>
<td>0.833</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Rose</td>
<td>WNSQ I</td>
<td>356.00</td>
<td>22.62</td>
<td>0.844</td>
<td>12.40</td>
<td>91.93</td>
<td>1.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wavelet-LVQ</td>
<td>421.57</td>
<td>21.88</td>
<td>0.827</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Blood Cells</td>
<td>WNSQ I</td>
<td>125.78</td>
<td>27.13</td>
<td>0.844</td>
<td>12.21</td>
<td>91.81</td>
<td>1.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wavelet-LVQ</td>
<td>133.91</td>
<td>26.86</td>
<td>0.823</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
It is clear that the proposed work gives better PSNR than that of the LVQ method for a similar bit rate. The proposed work gives reconstructed images with a quality factor varying from 22.62 db to 34.06 db and a bit rate of ~1bpp. In addition, it is inferred that the proposed work results in a 2 db gain in quality in comparison with the generalized LVQ technique. The
performance of the proposed work is also compared with the three competitive learning neural networks both with and without statistical modeling in the same wavelet domain. The comparison is done based on the quality factor corresponding to the bit rate $\approx 1.2$ bpp for each image. The observations are tabulated in Table 5.5. The proposed WNSQ I outperforms the other image coders in terms of good reconstructed image quality.

5.5 CONCLUSION

In this chapter a new codebook design method through Neuro-Savitziky-Galoy polynomial modeling is proposed. It is a practical method for improving the performance of the quantizer that is obtained from a known method of competitive learning. Simulation results show that the Savitzky-Golay codebooks produce images with better PSNR with respect to the blocking effect. The proposed technique exploits superior coding performance at low bit rate. The proposed work not only provides improvement in the objective rate distortion performance measure but also provides good subjective visual quality. Experiments have shown significant and consistent improvement in the performance of the proposed work over other existing methods. This work employs single level decomposition. In future, this work may be extended with more levels of decomposition along with a joint optimization using an entropy constrained coder. This joint optimization may result in further improvement in the rate distortion performance.
6. STRUCTURED REVERSIBLE NEURO-STATISTICAL SPARSE TRANSFORM FRAMEWORK FOR COMPRESSION OF GRAY SCALE IMAGES

6.1 INTRODUCTION
This chapter presents a single unified framework, the Generalized Reversible Directional Transform framework for image compression. This new framework is an outcome based on the study of the previously proposed frameworks and their interrelationships. The applicability of this framework for image compression is also demonstrated. The superiority of this method is illustrated by experiments in image compression. It is shown that very low bit rates of approximately 0.4bpp can be achieved when using bandeletization process for the transform coefficient.

A more structured approach to the design of the reversible transforms for image compression is advantageous than ad-hoc methods. By using a highly structured approach to transform design, more sophisticated transform can be produced than is possible with ad-hoc methods. Thus, general frameworks for the construction of reversible transforms are of great interest.

6.2 STRUCTURED REVERSIBLE NEURO-STATISTICAL SPARSE TRANSFORM FRAMEWORK
The reversible directional Wavelet-Bandelet transformation is considered for image compression. The preceding chapters consider two previously proposed schemes for image compression.
These schemes utilize a number of key ideas, but only a subset of these ideas is employed by each individual scheme. Table 6.1 shows the result of comparison between the performances of these schemes. The proposed WNSQ I gives reconstructed Lena image with a PSNR value 25.20 db and a bit rate of 1.24 bpp. The BVQ in the spatial domain gives less distorted reconstructed Lena image with PSNR value 27.38 db at a higher bit rate of 1.62 bpp. Obviously, the BVQ coder in the wavelet domain (WBVQ) gives a better rate-distortion measure than the others by reducing the bit rate of 0.45 bpp with a reduction in PSNR value by 2 db. Clearly, it would be beneficial to have a single unified framework that exploited all of these ideas at once.
Figure 6.1 Structured Reversible Neuro-Statistical Sparse Transform Framework for Image Compression: (a) Forward Transform (b) Reverse Transform
The key ideas from the proposed schemes are integrated and extended somewhat in order to create a single unified framework for reversible directional transform known as the Structured Reversible Neuro-Statistical Sparse Transform (SRNSST) framework for image compression. First, the wavelet transformed image coefficients are subject to the bandeletization process. Then the spatial LVQ quantizer discussed in Chapter 5 quantizes the bandeletized coefficients. Finally, the ZVP followed by Huffman encoder compresses the quantizer output. Given the operations employed in the forward transform, the inverse transform can be constructed through the stepwise inversion of each operation in the forward transform. That is, the inverse transform is formed by applying, in reverse order, the inverse of each operation in the forward transform.

6.3 EXPERIMENTAL RESULTS
This section presents the experimental results that illustrate the performance of the SRNSST framework. The graphs shown in Figure 6.2 depict the impact of the proposed work on various performance measures by varying the size of the original image. These graphs indicate that increasing the size of the image improves the performance of the system at the cost of computation time.

Figure 6.2a shows that the distortion is minimized as the image size is increased and thereby improving quality of the reconstructed image (Figure 6.2b). Figure 6.2c reveals that the compression ratio increases with the increase in image size. Hence the Bit Rate decreases with the increase in image size (Figure 6.2d).
Figure 6.2  The Impact of the Proposed Work on Various Performance Measures by Varying the Size of the Original Image: (a) Image Size Vs Distortion, (b) Image Size Vs PSNR, (c) Image Size Vs Compression Ratio, (d) Image Size Vs Bit Rate and (e) Image Size Vs Computation Time
This results in an improved rate-distortion measure which is the desirable factor (i.e.) both the bit rate and the MSE are reduced. But the computation time is increased gradually as the image size is varied from 64 x 64 to 256 x 256 (Figure 6.2e). And for the 512 x 512 size, images consume relatively much CPU time. This is practical in the sense that the information content present in an image is more in larger images. The objective performance measures like PSNR, Bit Rate etc., for various images including standard images and other natural images are tabulated in Table 6.2 and graphically described in Figure 6.3. In order to highlight the contribution made by the proposed work a comparison between the performance of the proposed SRNSST frame work and the WBVQ scheme is given in Table 6.2. It is clear from Table 6.2 that the proposed work reduces

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>IMAGE</th>
<th>PROPOSED ALGORITHMS</th>
<th>MSE</th>
<th>PSNR (db)</th>
<th>COMPRESSION RATIO</th>
<th>SPACE SAVING (%)</th>
<th>BIT RATE (bpp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rice</td>
<td>SRNSST</td>
<td>202.62</td>
<td>25.06</td>
<td>38.01</td>
<td>97.36</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WBVQ</td>
<td>238.74</td>
<td>24.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Baboon</td>
<td>SRNSST</td>
<td>185.89</td>
<td>25.43</td>
<td>36.49</td>
<td>97.25</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WBVQ</td>
<td>207.95</td>
<td>24.95</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Bird</td>
<td>SRNSST</td>
<td>96.40</td>
<td>28.28</td>
<td>36.00</td>
<td>97.22</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WBVQ</td>
<td>110.87</td>
<td>27.68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Lena</td>
<td>SRNSST</td>
<td>145.04</td>
<td>26.51</td>
<td>35.46</td>
<td>97.18</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WBVQ</td>
<td>179.73</td>
<td>25.58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Seismic Survey</td>
<td>SRNSST</td>
<td>192.17</td>
<td>25.29</td>
<td>51.80</td>
<td>98.01</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WBVQ</td>
<td>218.94</td>
<td>24.72</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Lifting body</td>
<td>SRNSST</td>
<td>159.85</td>
<td>26.09</td>
<td>45.10</td>
<td>97.78</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WBVQ</td>
<td>169.79</td>
<td>25.83</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
distortion consequently giving better PSNR than that of the WBVQ scheme for a similar Bit Rate. In addition, it is inferred that the proposed work results in a 1db gain in quality in comparison with the generalized WBVQ scheme. For the Seismic image with an acceptable amount of distortion, the compression ratio is reasonably high (51.8:1) saving 98.01% of the required memory space resulting in a very low bit rate (0.3 bpp).

The bar chart shown in Figure 6.3 describes the rate-distortion relationship for the Lena, Bird, Seismic and the geometric pattern images. This new methodology guarantees very low bit rate (≈ 0.3bpp) at tolerable amount of distortion (≈ 192 bytes per image) for the seismic image and a bit rate around 0.4bpp with distortion (≈ 145) for the Lena image. For the Bird image the amount of distortion is much smaller. Further, it reveals that the distortion reduces as the bit rate increases.

**Figure 6.3** Rate-Distortion Chart for Various Images using the SRNSST Framework
As the bit rate increases from 0.3 bpp to 2 bpp, the distortion reduces from 91.03 to 77.17 for the Geometric image, from 96.4 to 62.81 for the Bird image, from 145.04 to 80.23 for the Lena image and from 192.17 to 147.29 for the Seismic image respectively.

The performance of the proposed work is also compared with the well known JPEG 2000 image compression standard that is implemented in the same wavelet domain. Each of the test images given in Table 6.3 is compressed in a lossy manner using both the proposed work under evaluation and the JPEG 2000 standard, the resulting compression performance measures are shown in that table. For a lossy compression system, it is desirable to have a lower distortion with quality factor 20-30 db at possibly lower bit rate. Table 6.3 clarifies that the proposed framework results in lower distortion and lower bitrates in comparison with those results that are obtained for the JPEG 2000 standard. Further, the relative distortion and bit rate measures are provided to show the amount of increase in percentage for the JPEG 2000 with respect to the proposed framework.

Subjective testing is also undertaken to compare the quality of the lossy image reconstructions obtained with this frame work and the JPEG 2000. Figure 6.4 depicts the subjective quality measure for the Lena image. The subjective image quality obtained with the new frame work is better than that of the JPEG 2000, as demonstrated by the example depicted in Figure 6.4. The JPEG 2000 does not distribute the error effectively and the certain areas of the picture are not encoded with enough detail.
### Table 6.3 Performance Comparison of the Proposed Work with the JPEG2000 Standard

<table>
<thead>
<tr>
<th>S. No</th>
<th>Image</th>
<th>Algorithm</th>
<th>MSE</th>
<th>PSNR (db)</th>
<th>Compression Ratio</th>
<th>Space Saving (%)</th>
<th>Bit Rate (bpp)</th>
<th>Relative Increase in MSE (%)</th>
<th>Relative Increase in Bit-Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baboon</td>
<td>SRNSST</td>
<td>185.89</td>
<td>25.43</td>
<td>36.49</td>
<td>97.25</td>
<td>0.43</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JPEG2000</td>
<td>206.79</td>
<td>24.98</td>
<td>30.14</td>
<td>96.68</td>
<td>0.53</td>
<td>11.24</td>
<td>23.25</td>
</tr>
<tr>
<td>2</td>
<td>Lena</td>
<td>SRNSST</td>
<td>145.04</td>
<td>26.51</td>
<td>35.46</td>
<td>97.18</td>
<td>0.45</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JPEG2000</td>
<td>153.39</td>
<td>26.27</td>
<td>27.98</td>
<td>96.42</td>
<td>0.57</td>
<td>5.75</td>
<td>26.66</td>
</tr>
<tr>
<td>3</td>
<td>Dolphin</td>
<td>SRNSST</td>
<td>53.87</td>
<td>30.81</td>
<td>42.86</td>
<td>97.66</td>
<td>0.37</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JPEG2000</td>
<td>54.75</td>
<td>30.74</td>
<td>40.00</td>
<td>97.50</td>
<td>0.39</td>
<td>1.63</td>
<td>5.40</td>
</tr>
<tr>
<td>4</td>
<td>Seabed</td>
<td>SRNSST</td>
<td>82.74</td>
<td>28.95</td>
<td>39.17</td>
<td>97.44</td>
<td>0.40</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Seismic</td>
<td>JPEG2000</td>
<td>83.10</td>
<td>28.93</td>
<td>33.92</td>
<td>97.05</td>
<td>0.47</td>
<td>0.44</td>
<td>17.50</td>
</tr>
</tbody>
</table>

#### Figure 6.4: Original and Reconstructed Images

- (a) Original Image
- (b) Reconstructed Image using the Proposed SRNSST framework
- (c) Reconstructed Image using the JPEG 2000 Standard
The performance of the proposed framework is analyzed using natural images because image content is an important factor influencing the compression system effectiveness. These images vary considerably in size and cover a wide variety of content. Some primarily useful observations have
been made and are tabulated in Table 6.4. For images with a relatively greater amount of high-frequency content (e.g., Stars, Universe and Finger Print), the distortion is considerably more even though the bit rate is much lower. For the other natural images this framework performs reasonably well.

6.4 CONCLUSION

In this chapter, a new image compression framework has been constructed that is based on combining the bandeletization and code vector modeling features. It has been shown that especially the lowest bitrates with acceptable level of distortion results in a considerably better performance than the schemes that are based on the individual features. Further, the results are compared the JPEG 2000 standard results. The observations made for natural images are also recorded.
7. CONCLUSIONS AND RECOMMENDATIONS

7.1 INTRODUCTION
This thesis presents image compression using Bandelets and Neuro-Statistical methods. To accomplish bit rate reduction and improved visual quality, this thesis focuses on three principal challenges in image compression: Bandeletization of images to enhance redundancy removal, code vector generation using neural network and code vector modeling using polynomial regression. Research into these challenges has yielded improvements for most of them. Few directions for further research into image compression are also presented in this section.

7.2 CONCLUSIONS
The role of Bandelets in the field of image compression is analyzed in this thesis. The compression algorithm obtained as a combination of Bandelets, Quantization and Coding, outperforms the state-of-the-art methods in terms of the numerical criterion and the visual quality.

This thesis has looked at the commonly used vectors quantizers for still image compression. Experiments are carried out to compress few of the test images using the generalized Neural based Vector Quantization (NVQ), Multi Stage Vector Quantization (MSVQ), Pyramid Vector Quantization (PVQ), Tree Structured Vector Quantization (TSVQ) and the Linde-Buzo and Gray (LBG) algorithm for vector quantization. And the results are tabulated in chapter 4. It is shown that the NVQ technique performs better than the other reported techniques. Therefore, the proposed work uses the NVQ technique as its base. There are three neural networks which use the
competitive learning rule. They are the ComPetitive Learning Neural Net (CPL), the Self-Organizing Map (SOM) and the Learning Vector Quantization (LVQ). Then proposed polynomial regression analysis is carried out using these three neural networks.

A summary of the conclusions of the individual methods are also discussed in this section. The role of bandeletization of images in the field of image compression is studied in Chapter 4. It is found that this process introduces redundancy which is a desirable feature for compression. Hence a new approach has been devised to compress images efficiently. This approach is also tailored to the wavelet domain.

The proposed work in the chapter 5 employs LVQ for the codebook generation phase. Further, a new polynomial regression concept is introduced for modeling the code vectors of the codebook which helps reducing the distortion thereby improving the reconstructed image quality. It is practically proved that the proposed NSQ scheme yields superior reconstructed psycho-visual image quality than the NVQ technique in terms of better PSNR ratio while maintaining same compression ratio. Further, this method is extended to the wavelet domain which promises high compression ratio and hence more memory space saving and low bit rate.

Finally, a single unified framework called the Structured Reversible Neuro-Statistical Sparse Transform framework for image compression is developed. This new framework is an outcome based on the study of the previously proposed schemes and their interrelationships. The applicability of this framework for block transforms is also demonstrated. The superiority of this method is illustrated by experiments in image compression. It is
shown that very low bit rates of approximately 0.4bpp can be achieved with the process of bandeletization of the wavelet co-efficient.

The state-of-the-art in image compression research can be summarized as follows: Continuous research is performed on natural images, aiming at better reconstructed image quality and lower bit rate. The reduction in bit rate leads to higher compression ratio and more memory space saving. The present work has outlined a prototype technology, a multiresolution image analysis and bandeletization, which lends itself to meet all these objectives and in fact has proved to be feasible. This new Image compression framework performs at a satisfactory level for all standard test images, seismic and texture images, while they often fail when applied to images with high frequency content like Finger print images.

7.3 **RECOMMENDATIONS**

Despite the achievements of this work, a number of problems remain to be solved in image processing. Further research should focus on three specific issues: computational complexity, design of a generic codebook for the vector quantizer and issues related to images of different shapes. This section gives an exploration of the possible approaches to these issues.

7.3.1 **Computational Complexity**

In general, high complexity methods are the only way to obtain competitive rate distortion performances and hence a balance has to be reached between the speed of a system and its performance which is particularly relevant in selecting codec for use in video compression.
7.3.2 Design of a Generic Codebook

Codebook design plays a crucial role in the vector quantization phase of an image compression system. It is expected to transmit a copy of the static codebook along with the coded image. Therefore, if it is possible to construct a representative codebook then the space and time consumed by this codebook may be reduced. Neuro-Fuzzy Soft computing approaches may be applied in this context as they do imitate human thinking to some extent.

7.3.3 Shape of the Image

The proposed framework accepts only square images because the bandeletization process expects that the input must be square in size. As natural images are not always square images, necessary changes are to be incorporated in this stage.

Furthermore, the proposed work may be extended to compress color images, audio and video data. The application of the bandeletization process and polynomial regression may be studied in various contexts of image processing which includes image segmentation, water marking and enhancement.
REFERENCES


151


[157] Baocai Tin, Xin Li, Yunhui Shi, Feizhou and Nan Zang. Directional lifting-based wavelet transform for multiple description image


159