2007

Error resilient image transmission using T-codes and edge-embedding

Premchander Reddy
West Virginia University

ERROR RESILIENT IMAGE TRANSMISSION USING T-CODES AND EDGE-EMBEDDING

By

Premchander Reddy

Thesis Submitted to the college of Engineering and Mineral resources

at West Virginia University

In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

Approved by

Dr. Donald Adjeroh, Committee Chair Person

Dr. Xin Li

Dr. James Mooney

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia

2007

Keywords: Error resilience, Edge embedding, T-codes, edge-directed interpolation, scrambling scheme.

Abstract

Error Resilient Image Transmission using T-codes and Edge-Embedding

Premchander Reddy

Current image communication applications involve image transmission over noisy channels, where the image gets damaged. The loss of synchronization at the decoder due to these errors increases the damage in the reconstructed image. Our main goal in this research is to develop an algorithm that has the capability to detect errors, achieve synchronization and conceal errors.

In this thesis we studied the performance of T-codes in comparison with Huffman codes. We develop an algorithm for the selection of best T-code set. We have shown that T-codes exhibit better synchronization properties when compared to Huffman Codes. In this work we developed an algorithm that extracts edge patterns from each 8x8 block, classifies edge patterns into different classes. In this research we also propose a novel scrambling algorithm to hide edge pattern of a block into neighboring 8X8 blocks of the image. This scrambled hidden data is used in the detection of errors and concealment of errors. We also develop an algorithm to protect the hidden data from getting damaged in the course of transmission.

This thesis is dedicated to my family, Kalpana, Ramakrishna Reddy, Vijitha

And

In the loving memory of late Pradeep and Vijayendra

Acknowledgements

I would like to take this Opportunity to express my cordial gratefulness to my advisor Dr. Donald Adjeroh who devoted immense time and effort in this research. I very much appreciate his great patience and tolerance and best of all his invaluable trust. It has been my pleasure to work with a farsighted advisor like Don.

I would also like to thank my Committee members Dr. Xin Li and Dr. James Mooney for their help.

I would like to thank my fiancé, kalpana, for her support and love through out my masters studies. I am very thankful to my father, whose invaluable trust and immense love drives me forward in my life. I am very thankful to my mother and my sister, without their love, this thesis would not have been completed.

I would like to take this opportunity and thank all my friends, whose support and cherishing company, has helped me get through hard times.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

1.1. General Introduction

Image is the best form of information for a human being to comprehend. An image was first digitized in the year of 1957. Since then digital image processing has grown in leaps and bounds in the fields of satellite imagery, medical imaging, photo enhancement etc. With growing need of transmission of digital images in various applications, the problem of network bandwidth also increased, which brought in the need of digital Image compression and decompression. Since the medium of transmission under consideration is prone to errors and delays there is also a need to design an efficient error resilient scheme. A significant research has been going on in the field of image transmission in order to cope with the problems aroused by the channel errors. The digital image communication has found increasing demand in the fields such as geographic information systems, handheld devices with multimedia compatibility, biometric applications. These applications require a highly efficient compression system, and appreciable quality of service (QOS). Both wired and wireless networks are highly prone to unexpected errors and delays which can

deteriorate the image when transmitted over them. Therefore a good error resilient scheme in addition to a good compression scheme in order to support the band limited channels is expected as a solution for this problem.

1.2. General Digital Image Communication Scheme

Figure 1.1 shows a general block diagram of a digital image communication scheme. The Image is initially transform coded to aggregate most of the image data into one corner of the result. The Different Transform coding methods used are Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT) The Quantization step takes advantage of the human visual perception which cannot exactly distinguish high frequency variation.

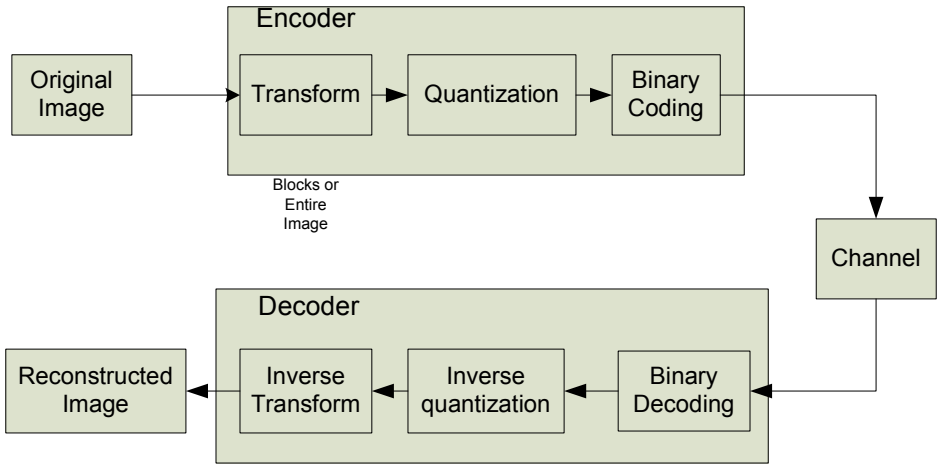

Figure 1.1 General Digital Image Communication Scheme

This is the main lossy step in the process where all the high frequency coefficients are rounded to zero. The low frequency coefficients become smaller in this process which can be binary coded using less number of bits. The choice of quantization parameters decides the compression rate and distortion. The Quantized coefficients which are in a 2-D format are reshaped to form a 1-D stream and then sent to a Binary Coder which makes use of the statistics of the quantized coefficients. In the Binary Coding stage the Quantized coefficients are assigned binary codes. Some binary coding methods are Huffman coding, Arithmetic coding, and T-coding. This is a lossless stage and is completely reversible. The Coded stream of bits is then sent through a network channel where the bit stream could get damaged due to the noise present in the channel. The received bit stream is then sent through a binary decoder which performs the opposite action as that of the binary decoder. Then the stream is again reshaped into 2-D format and it is inverse quantized. The inverse quantized coefficients are then inverse transform coded to get the reconstructed image. A stage for error correction or error concealment could be added before or after the inverse transform coding to enhance the quality of the reconstructed image.

1.3. Problem Addressed

Multimedia Communication has become very popular recent years, and is being used in a large variety of applications. Any transmission error can result

in high degradation of the image. Many standards such as JPEG [1] and JPEG2000 [2] are used to achieve required compression rates demanded by the band-limited networks. But most of these compression standards use variable length coding as entropy coding to achieve high compression. An error in the transmission may result in loss of synchronization at the decoder end. A common solution to this problem is to include at intervals some relatively long unique synchronization code words (sync words) followed by some block address information. These sync words involve the addition of redundant information [3].

Many variable length codes exhibit a tendency for resynchronization to occur automatically following any error. As stated in [4], the range and richness of the structures implicit in variable length codes renders the identification of general mechanisms for self-synchronization difficult, perhaps impossible. Huffman's construction [5] ensures optimal coding, i.e., minimal redundancy, but provides no assurance as to how the decoder will perform in the face of channel disturbances. In practice, errors may propagate for a considerable period [4]. In [6], Titchener and Hunter illustrated the expected synchronization delay (ESD) to determine the synchronization status. In [9] with ESD as an objective function, Titchener explored a method for estimating the synchronization performance based on T-codes. T-codes are a class of self-synchronizing codes that are derives from a recursive T-augmentation

construction, which typically synchronize within 2-3 code words. The Usage of self synchronizing T-codes against Huffman has not been studied in detail.

Another problem encountered in multimedia communication is error concealment and error correction. There has been a lot of research in the field of error resilient entropy coding. Error resilient techniques can be classified into 3 major groups with the following properties [7]: an interaction between the encoder and decoder, as a re-send signal, or post-processing operations at the decoder to recover lost information, or leaving some extra redundancy at the encoder to minimize the reconstruction error. Some of these methods by hiding some information at the encoder which can be useful at the time of error concealment. This data hiding technique was used in steganographic applications like copyright protection and other security-based applications [8, 9]. Error resilient image coding has become a new application of data hiding. Many methods were proposed utilizing this property but none of the previous approaches has proposed a complete image/video codec with detection, synchronization and recovery (reconstruction) capabilities together based on data hiding and tested under noisy channel conditions [10]. This work concentrates on trying to develop a scheme that takes into consideration all of the above capabilities together based on data hiding. Many methods have been introduced on how to hide the data, the most common one being the odd-even method. Here the DCT coefficients are forced to be even or odd based on their

frequency or their position. Next comes the question about the information that has to be embedded. Block intensity information is approximated by using either edge-direction information [11] or low quality coded version [12]. According to [5] interpolation along the edge direction gives superior results when compared to the popular bilinear interpolation. Instead of trying to estimate the edge direction of the lost block from the neighbors at the decoder, hiding the edge information at encoder is preferred [11]. Some novel concealment methods have been proposed that uses data hiding as a tool for error concealment but many of them neglected the use of embedded data in the detection of error, which could be found by measuring the inconsistency between the block and hidden data. Most of the above mentioned approaches neglected the fact that detection of the edge information from the hidden data becomes hard when the hidden data gets damaged in the course of transmission. Protection for hidden data is also a must when considering data hiding as a tool for error concealment.

1.4. Research Objectives

Most of the standard image codecs are block based and a real-time channel introduces unexpected errors and a possible error may damage not only a single block but also all the blocks that follow. As most of the standard compression schemes use variable length coding, any error would propagate to the next symbols or even blocks. So there is a need for the utilization of self

synchronizing codes such as T-codes at the encoder. Considerable error is still left over even with the help of lower synchronization delay that would still damage the image which should be efficiently concealed at the decoder. The main objectives of the research is to develop a scheme which uses self synchronizing T-codes as the entropy coding, a detailed study of T-codes against Huffman coding, and to demonstrate the use of data hiding into neighbor blocks, which is used at the decoder for better error concealment. This work also shows a novel method on how to protect the hidden data by hiding them in more than one neighbor block.

The main motivation for this work comes from the fact that T-codes demonstrate the lowest synchronization delay of about 2-3 symbols and the data hiding scheme can be utilized for both error detection and error correction without adding extra overhead.

1.5. Research Contributions

Most of the current standard compression schemes use variable length codes for compression. Although many variable length codes exhibit a tendency of resynchronization following an error, According to [4] it is very difficult for variable length codes to self-synchronize. Huffman coding gives ideal compression with minimum redundancy, but it does not assure synchronization at the decoder. T-codes are a set of self synchronizing variable

length codes which exhibit very low synchronization delay. This property of T-codes is not used much in any recent compression schemes. This work provides a detailed comparison between T-codes and Huffman coding and shows that T-codes show good synchronization properties when compared to Huffman coding with a compromise made in CPU time taken.

In a natural image the intensities from one pixel to its neighborhood varies in a smooth fashion, unless there is an edge. Most of techniques which conceal the error in a block make use of this smoothness property by making use of spatially closer data. The most popular method is the bilinear interpolation, where a pixel from the erroneous block is replaced by the weighted average of the four neighboring pixels. Since some parts of an image may have high variations simple bilinear interpolation does not give satisfactory results. So the edge information needs to be taken while interpolating making sure that the interpolation is not done across the edge. This work makes use of data hiding as a tool to hide the edge information at the encoder end so that the interpolation at the decoder is done along the edge direction where ever an edge is detected. This work also proposes a novel method to detect the errors at the decoder and also proposes a new method for the protection of hidden data.

1.6. Organization of research

Chapter 2 gives a basic background of Image compression and talks about different techniques used to tackle with the problem of error resilience and the problems with the different approaches. The concept of data hiding for error concealment is discussed briefly. This chapter also mentions the role of variable length codes in error resilience. At the end a brief introduction to the approach used in the work is presented.

Chapter 3 presents a detailed description of T-codes and Huffman coding. All the details of the study on T-codes VS Huffman coding are also presented in this section. Finally the comparative results obtained from the simulations are presented.

Chapter 4 briefly describes the concept of error resilience and the concept of utilization of data hiding for error concealment. This section presents also presents the proposed method of data hiding based error detection and edge-based interpolation. Finally the results from the simulations are presented.

Chapter 5 concludes the thesis by summarizing the work, giving comparative results with other proposed methods and providing suggestions for any future work.

Chapter 2

Background & Literature Review

2.1. Image Compression

Image compression is a process of compressing an image to reduce its redundancy for the purpose of storage or transmission. There are two types of compression, namely, lossy compression and lossless compression. In lossy compression, the compressed image can be reproduced with a possible loss of information. In lossless compression, the compressed image is reproduced exactly without any loss of information. Some of the methods used for lossless compression are run-length coding, entropy coding, dictionary based coding. An important measure in image compression is the compression rate or bit-rate. Compression rate is defined as the quotient of the size of the output in bits to the size of the input in bytes. Compression rate or bit rate is measured in bits per symbol (bps). For effective utilization of compression technology, different applications designed by different vendors should communicate with each other. Some sort of standardization is required in order to achieve this. Some of the standards in image compression are Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF), and Portable Network Graphics (PNG).

Compressing an image typically comprises of two steps

1. Reduction of redundancy making use of the structure of the data.
2. Assigning binary code words to the non-redundant data.

Step 1 usually consists of two steps namely transformation and quantization. The output of the transformation gives a representation of the image that is more suitable for efficient compression. The quantization process is irreversible, and produces an output with limited number of symbols that are ready to be binary coded. Step 2 is an important step in image compression technique, where the quantized coefficients are assigned some binary codes. The binary codes can be Fixed Length Codes or Variable Length Codes (VLC). VLC maps code words depending on the frequency of the symbols, higher frequency symbols is mapped onto a smaller length code word and vice versa, so as to minimize the average length of the binary code word. Some of the approaches used for VLC are Huffman coding [57], arithmetic coding, and T-codes [28]. Some standards like JPEG use DCT-based compression, where the image is first broken into 8x8 blocks. These blocks are then transformed using DCT. The DCT coefficients are then quantized using a standard table and coded using either Huffman Coding or Arithmetic Coding. The binary coded bit stream is then transmitted over a channel to a receiver where it is decoded. Ideally the decoded image should vary only a little from the original image due to the lossy quantization, However in practical channels, there is a lot of unexpected noise. Due to this noise the image gets further deteriorated before

it gets decoded. The solution for this problem is designing a good error resilient system that can handle real life channel error.

2.2. Error Resilience in Image Transmission

Error control in image communication poses great challenges for many reasons. Compressed image streams are highly vulnerable to channel errors due to the usage of VLC and predictive coding at the source coding stage. With VLC, even a single bit error can cause the decoder to lose synchronization. This makes the correctly received bits useless. The techniques used to overcome transmission errors can be classified into three categories [14].

(I) Error Resilient Encoding: Those introduced at the source and channel encoder that provide more resilience to potential errors.

(II) Decoder Error Concealment: Those invoked at the decoder after detection of errors, to conceal the effect of errors.

(III) Encoder and Decoder Interactive Error Control: Those which require communication between encoder and decoder, so that the encoder adapts its operation based on the conditions detected at the decoder.

2.2.1. Error Resilient Encoding

In this approach some redundancy bits are added at the encoder side to counter the effect of error during the transmission. This addition of redundant bits makes the encoder less efficient than the encoders that aim for coding efficiency. The design of Error Resilient (ER) encoders aim at providing maximum gain in error resilience with smallest possible redundancy [14]. Many approaches have been proposed to implement ER encoding.

Insertion of Synchronization markers: The main reason for a multimedia stream to be sensitive to bit errors is loss of synchronization at the decoder. The easiest way of ER encoding is by adding synchronization markers at regular intervals so that the decoder can resume decoding correctly after it recognizes a synchronization marker. But the drawback with this approach is the increase in overhead, due to the increase in number of bits used to represent a stream. Another drawback is that, once an error is discovered, the decoder discards all the bits until it finds a sync marker resulting in loss of data. Insertion of Synchronization Markers has been considered for the MPEG-4 and JPEG2000 standards [35], [36].

Reversible Variable Length Coding: In this approach the decoder can also decode from the backward direction [54]. The decoder decodes properly after synchronization marker is observed and before the next synchronization marker is observed. But the problem with this approach is increased complexity due to the addition of backward decoder.

Multiple Description Coding: Multiple Description Coding (MDC) [15], [18], was invented at Bell Laboratories for speech communication over telephone networks. It can be considered as a joint source channel coding scheme that can provide good resilience against errors. One of the first MDC coder was designed by Vaishampayam in which multiple description scalar quantizers were used in extension to the JPEG coder [16]. The basic concept in this approach is assuming multiple independent channels between encoder and decoder. Each channel is assumed to be detachable temporarily. The source bit stream is divided into several streams called Descriptions and are transmitted over different channels. All the descriptions are correlated and have equal importance. This correlation is what enables the decoder to estimate the missing description from a received one. Again the drawback in this approach is added redundancy due to the correlation.

Forward Error Correction Coding (FEC): One of the standard methods used to achieve resilience against channel errors was Automatic Repeat Request (ARQ). Although ARQ protocols transmit the images in a lossless manner, there is an excessive transmission delay due to the retransmission of the packets [17]. Therefore an alternate approach is to apply Forward Error Correction to packets. The idea behind FEC is add redundancies to the data before transmitting it. When there is an error the redundant data is used to recover the lost data. FEC is usually used in conjunction with other source coders.

Self –Synchronizing Entropy Coding: Many compression schemes including JPEG [1] still image standards use VLC as they are very efficient in reducing the bit-rate but they have a disadvantage of losing synchronization at the decoder when an error occurs. Therefore designing a VLC that can reduce the loss of synchronization still remains as a major problem. One way to counter this is by designing a VLC that has some synchronizing code words [20], [24] in them. Another is to insert some synchronizing code words that make the VLC synchronizing. There have been many attempts to design codes that could overcome the problems of error propagation and synchronization. Most of the early works have been summarized in [25] and the references contained therein. Some of the techniques that were proposed earlier are as follows, periodic insertion of universal synchronizing sequence that can resynchronize the decoder regardless of the slippage. Although this is an efficient technique it introduces unwanted overhead. In [20], [21], [23], [26] a code that guarantees synchronization is used. In all the cases the use of optimal source code without additional overhead is precluded in order to guarantee synchronization. So there is a need to design self-synchronizing entropy coding which takes less bits to resynchronize in order to reduce the over head problem. Titchener invented T-codes in 1984 [28], which have these properties; it has been proved that T-codes synchronize within 2-3 symbols. This will be elaborated in the later chapters. This self-synchronizing entropy coding combined with good decoder error concealment scheme could yield better results.

2.2.2. Decoder Error Concealment

Decoder error concealment is the process where by the effect of the error is concealed after it has been detected. It is well-known that images of natural scenes have predominantly low frequency components, *i.e.* the color values of spatial and temporally adjacent pixels vary smoothly, except in regions with edges. Techniques that have been developed for recovering texture information make use of the above smoothness property of image/video signals, and essentially they all perform some kind of spatial/temporal interpolation [14]. Some of the techniques proposed for decoder error concealment are discussed briefly in this section.

Spatial Interpolation: This is a simple and popular approach of error concealment where the pixels from the erroneous blocks are interpolated using the correctly received neighboring blocks [29]. The interpolation weights are calculated according to the distances between the damaged pixel and correctly received pixel. The smaller the distance is the greater the weight. Since some parts of an image have high variations ordinary spatial interpolation does not yield satisfactory results. An image has complex geometric structures. Some techniques have tried to estimate these structures and use them in the interpolation. In [5], [34], directional spatial interpolation is proposed for better interpolation. The pixel is reconstructed via interpolation of two nearest pixels from the correctly received neighborhood blocks in a predicted direction. In [5]

two nearest surrounding pixel layers of a missing block are used to estimate an edge passing through the missing block and interpolation is performed along this edge direction. In [34] the Hough transform is used to find the direction of interpolation. Another simple technique of spatial interpolation is to replace the erroneous blocks by the average of the DC coefficients of the correctly received blocks [32].

2.3. Data Hiding for Error Protection

2.3.1. Data Hiding

People have tried to hide information for various purposes since the archaic period. Steganography is the general name given to data hiding. Steganography can be formally defined as a process of embedding some data into a host signal imperceptibly to get a composite signal. The growth of digital media has made the information hiding problem important. Depending on the application, steganography can be viewed as three types: data hiding, copyright marking, and fragile watermarking. Data Hiding is primarily used for error concealment by transporting some error concealment data from encoder to the decoder. The type of steganography used for authentication of digital content is called fragile watermarking. Since steganography can be applied to any kind of digital data, it has attracted many researchers for many reasons. Steganography has found its place in many applications such as covert communication [39], broadcast monitoring [38], copyright protection, transition tracking [37], content

authentication [8], copy control, and also error concealment [40-43], [13]. Steganography can be used in covert communications by embedding some secret information transparently into the image or video and then sending it to the receiver. Unintended receivers do not even realize the presence of the hidden message [39]. Steganography has been used for owner identification of an original Work [37]. Since the embedded data is invisible and reduction in visual quality is very minimal, steganography has found its importance in the application of copyright protection.

Digital media can be easily and perfectly modified or altered, thus steganography can be used for content authentication. Here a signature is embedded into the digital content such that even a slight alteration of the content would destroy the signature. Thus it would be easy to detect when the content is not authenticated [8].

Error concealment in multimedia communications has become a new application of steganography. Some recent methods for error concealment using data hiding are proposed in [13], [33], [40-43]. Error concealment using data hiding approach is a combination of source coding and decoder error concealment, where some data about the image blocks is embedded into the bit stream at the encoder side and is transmitted to the decoder. The decoder then retrieves the hidden information and makes use of the information in concealing the error. In [5], [34], the edge direction is estimated at the decoder which is a complex computation for a decoder. Instead of estimating the edge

direction at the decoder the edge direction can be embedded in the entropy coded bit stream. The decoder then can retrieve the edge direction of the blocks and interpolate the edge blocks along its edge direction. This reduces a lot of computational effort at the decoder.

2.3.2. Implementation of Data Hiding

Many different methods have been proposed on how to implement data hiding depending on the application. The earlier approaches were based on modifying the least significant bits (LSB). However embedding in LSB may not be the best approach, since the data hidden can be lost by simple changes in the host image. Spread spectrum techniques [44] were proposed to counter this problem, where data is added to the image in the spatial or transform domain. Later, methods based on quantization of the host signal called Quantization Index Modulation (QIM) were proposed and were shown to perform better than spread spectrum approach [45], [46], [47], [48]. Modifying the Non-zero least significant DCT coefficients is also one of the methods used to hide the data. This approach is termed as *Even-Odd signaling* method. This method is very simple to implement hence widely used [49].

2.3.3. Utilization of Data Hiding In Error Concealment

Figure 2.1 shows the working of error concealment technique using data hiding. The methods explained in section 2.2.2. try to estimate the error block from the available correctly received block. Simple interpolation using the available neighbor blocks does not yield good results because an image has high variations and complex structures such as edges.

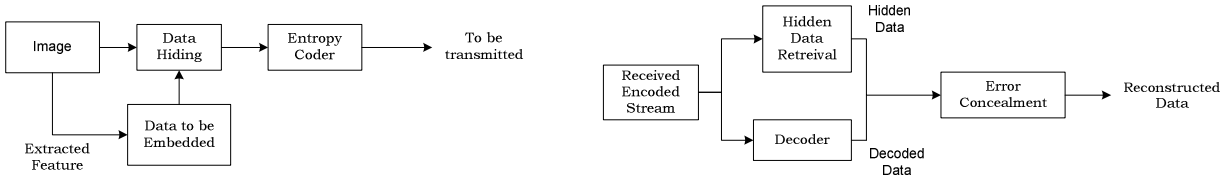

Figure 2.1 Procedure at the Encoder and the Decoder

In [5], the edge orientation of the erroneous block is estimated from the available neighboring blocks to interpolate along the edge direction. The edge direction in this case is estimated at the receiver, which makes the decoding part complex. In [11] the major edge direction of each block is estimated at the encoder and is embedded into the neighboring blocks. This way the decoder does not have to estimate any possible edges and this prior knowledge of the edge direction yields better results. In [12], coefficients of low quality version of each block are hidden in the neighboring blocks. This approach was devised to avoid interpolating the spectral coefficients of the error block at the decoder.

But a large number of bits are required to embed all the coefficients, which is not acceptable. In [50], instead of embedding some secondary information in a host image, the host image is embedded into itself. In this approach the embedding is done in the block DCT domain. The DC coefficient, which carries majority of the information of a block, is embedded into one of its neighboring blocks. The DC coefficient is extracted at the decoder and is used in combination with some post-processing technique that takes advantage of interblock correlation. In this method again the required number of embedded bits is higher because representation of each DC coefficients requires a large number of bits. Further, the AC coefficients are completely neglected in this method which can deteriorate the image quality.

Another important problem in multimedia communications is loss of synchronization at the decoder. This problem can be countered up to some extent by using data hiding technique. The authors of [18] used the technique of hiding the resynchronization data, such as length of a block into neighboring blocks. This way the decoder can resynchronize as soon as an error is detected and resume decoding in the correct sequence.

2.4. VLC for Error Protection

Loss of synchronization is one of the main problems faced in multimedia communications. Most of the compression standards including JPEG [1] use variable length coding, where in a single bit error can cause the decoder to lose synchronization.

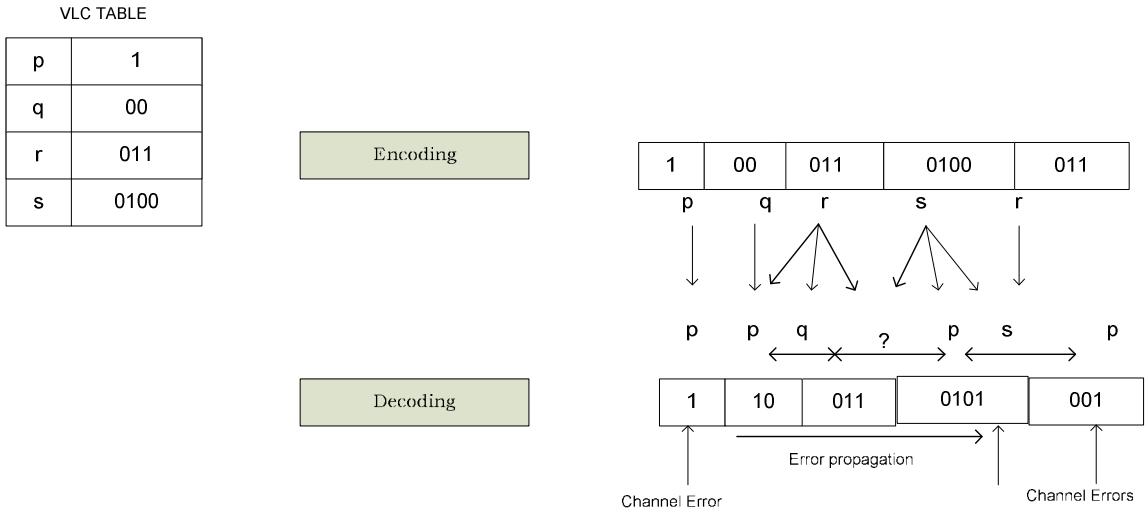

Figure 2.2 Error Propagation in VLC

Consider, Figure 2.2. The transmitted bit stream was “pqrsr” but due to a single bit error the receiver lost synchronization and the decoded bit stream becomes “ppq?psp”. In this example, after decoding “ppq” the decoder could not find a valid binary code word so it simply discards all the bits until it finds

a valid binary code word to decode. Many methods were proposed in the past to counter this synchronization problem. Some of them are discussed in this section. In both [51] and [52] frequent restart markers were used in order to stop the error propagation in VLC. Whenever a restart marker is observed by the decoder, it resumes decoding correctly from that point thus avoiding the error propagation to more number of bits. This method however increases the overhead due to the addition of extra bits in the form of restart markers. In [53] a similar approach is followed where the restart markers are placed at the ends of lines and is combined with error detection and error concealment. The approaches mentioned above increases the bit rate which is not acceptable in many practical applications. Therefore VLC techniques that provide re-synchronization without increasing the bit rate are needed. One such approach is proposed in EREC as explained in [3]. In this approach the VLC code words are reorganized into blocks of know length such that the each blocks start at a know position, and the decoder is synchronized at the start of each block. An alternative approach was devised by Wen and Villasenor [54], where the VLC code words are symmetric and the encoded stream thus obtained can be decoded in both forward and backward direction to maximize the amount of data decoded. This method leads to increased complexity at the decoder end. In order to develop an approach that has re-synchronizing properties and also does not increase the bit rate, researchers looked into the concept of re-synchronizing variable length codes. A re-synchronizing variable length code

contains one or more synchronizing code words that synchronize the decoder regardless of the previous data. Rudner [55] developed an algorithm to construct codes with minimum redundancy and optimum synchronization property. But the applications of this optimal re-synchronizing VLCs were restricted to sources with alphabets less than 30 [55]. Therefore non-optimal re-synchronizing VLCs were designed, which make use of optimal re-synchronizing VLC in combination with additional synchronizing code words such as Ad-Hoc Marker Code words [23], and Extended Synchronizing Code word [56]. In [56] Hemami proposed re-synchronizing VLCs formed by designing re-synchronizing Huffman code (RHC) with added Extended Synchronizing Code word (ESC). The compression is non-optimal in this approach as the average length of the code word is increased as a result of addition of the ESC. In [4] Titchener illustrated the synchronization performance of T-codes with expected synchronization delay as the objective function. T-codes are a family of self synchronizing codes which can be constructed in a recursive manner. It has been claimed that T-codes synchronize within two to three code words [4]. Unfortunately not much work has been done on the utilization of T-codes in error resilient image transmission. The utilization of T-codes as entropy coding in error resilient image transmission of an image is one of the major motivations of this work.

2.5. Our General Approach

Many compression standards such as JPEG [1] uses variable length codes (example Huffman codes) to perform compression. In our approach we use T-codes for this purpose utilizing its self synchronizing properties. We also make use of data hiding to assist the decoder in efficient error concealment. Many methods have been proposed on using data hiding in conjunction with decoder side error concealment [40-43], [13], [33]. Our Approach follows a similar technique as used in [11]. However in our work we propose a novel edge-pattern classification method to detect and correct errors. Edge information of each block is estimated at the encoder side and blocks are classified into different edge *classes* based on their edge pattern. The edge *class* of each block is embedded into the low frequency DC coefficients in some other blocks. Data hiding is performed using the *odd-even signaling* method [49]. Usually the side information is embedded into neighboring blocks. In our approach, we follow a novel *scrambling* algorithm to hide the data into three different blocks. These blocks are chosen to be far from the current block and from each other in order to protect the hidden data from the burst errors. Since the hiding process does not add any redundancy hiding in multiple blocks does not add extra overhead. The embedded data is then retrieved from the received encoded bit stream and erroneous blocks are detected by comparing the de-embedded information from all the three blocks that it has been embedded into. Then the decoder performs

error concealment by interpolating the erroneous blocks along their edge direction. The concealment is done by simple spatial interpolation using the neighboring blocks.

Chapter 3

Comparative Study of T-Codes and Huffman Codes

3.1. T-codes

Entropy coding is an important step in image communications. The main objective is to ensure that the redundancy of the source is minimized. Other desired properties in entropy coding are developing a coding process that is suitable for all kinds of data and also constructing codes that are capable of re-synchronizing the decoder. Irrespective of nature of channels, codeword synchronization is a very important aspect of digital media communication. In these years of advancement in image compression, many algorithms were proposed to construct self-synchronizing codes [20], [21], [54]. Our approach uses a class of VLCs that poses an extraordinary capability for self-synchronization, called T-Codes [6], [28].

Titchener devised an algorithm [28] for the construction of prefix free code words which can be summarized by the defining T-codes. T-codes are generated by recursive *T-augmentation* [4] of elements in an alphabet. The more formal definition of T-augmentation and T-codes according to [28] is as follows

Definition of T-Augmentation [6]:

Denote N by the set of natural numbers and let $N^+ = N\backslash\{0\}$. Let A be a finite alphabet. The number of elements in A is denoted by $|A|$. The set A^* denotes the free monoid generated by A under concatenation. The elements of A^* are called *strings*. The length of a string $x \in A^*$ is denoted by x_l. λ denotes the empty string, $\lambda_l = 0$. A set $A^*\backslash\lambda$ is denoted by A^+.

For a code set $S \subset A^*$ a string $p \in S$ and a positive integer $k \in N^+$ the generalized T-Augmentation of S, denoted by S_p^k, is given by

$$S_p^k = \left\{p^i u \mid i = 0,1,\dots,k \;\; u \in S\backslash\{p\}\right\} \cup \{p^{k+1}\} \qquad (3.1)$$

The generation of S_p^k from S is known as T-Augmentation, where p is called the *prefix*. k is known as the *expansion parameter*. Our approach uses a simple augmentation, where $k = 1$.

Definition of T-Codes [28]:

The alphabet A and sets derived from A by one or more *T-augmentations* as shown in equation (3.1) are called T-Code sets.

T-codes in general have the following properties [56]:

(a) Variable Length Codes with synchronizing code words spread all over the code set.

(b) High potential to show self-synchronizing property when an error occurs, even when the error cannot be recognized as such.

(c) Synchronization is re-established after the decoder recognizes an error.

In [6] Titchener explains synchronization process using the T-augmentation equation (3.1), in which it is evident that all but one elements in S, i.e., $(|S|-1)$, where $|S|$ denotes the number of elements in S, are included in the new set S_p^k, $k+1$ times, with the prefixes $p^0, p^1, \dots, p^k$ respectively. These, together with the element p^{k+1}, give

$$\left|S_p^k\right| = (k+1)(|S|-1)+1 \qquad (3.2)$$

The above definitions of T-augmentation and T-codes could be easily understood with the help of an example. Suppose a binary alphabet $S = \{0,1\}$, to obtain the first set of code words from S we apply T-augmentation on it, i.e. we remove one of the element from S and augment it with both the elements of S. Suppose if we remove the element '0' from$S$, the new set, $S_{(0)} = \{1, 00, 01\}$. Application of T-augmentation to $S_{(0)}$ results in more code words. In case we remove element '1' from $S_{(0)}$and T-augment it, the new set obtained is $S_{(0,1)} = \{00, 01, 11, 100, 101\}$. In a similar way another code set can be generated by eliminating the element '01' from $S_{(0)}$. The new code set formed if we eliminate '01' from $S_{(0)}$ in the previous example is $S_{(0,01)} = \{1, 00, 011, 0100, 0101\}$. And in similar way if we eliminate '00' we get a new set $S_{(0,00)} = \{1, 01, 001, 0000, 0010\}$. Thus for each augmentation level n there are $x_c = \prod_{i=0}^{n-1}(2^i + 1)$ different code sets, each consisting of $(2^n + 1)$ code words.

Tree Generation using the above example:

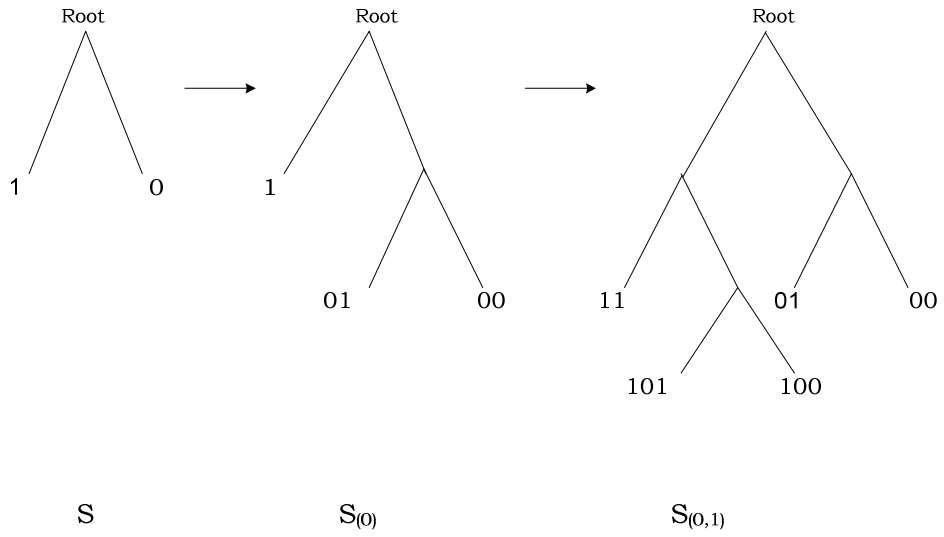

Figure 3.1 Generation T-Code trees $S_{(0)}$ and $S_{(0,1)}$ from $S = \{0, 1\}$

Shorter code words may not be the best prefixes because they result in uniform tree generation. Therefore, there is a need to develop an algorithm to choose the best T-codes that give a better compression. We proposed a novel algorithm to select best T-codes from the huge data base of T-code sets.

The following table gives a numerical explanation for the expressions given for number of Codeword sets and number of Code words within each code set. The table is shown for augmentation levels 0 to 9.

Augmentation Level (n)	Code words within each code set $(2^n + 1)$	No. of different Code sets (x_c)
0	2	1
1	3	2
2	5	6
3	9	30
4	17	270
5	33	4590
6	65	15140
7	129	9845550
8	257	120075950
9	513	326409519150

Table 3.1: Size and number of code sets for each augmentation level.

3.1.1. Self-synchronization of T-codes

Consider a string $x = x_1 x_2, \dots, x_t \in S$ and let the codewords formed as a result of T-Augmentation of S be $y_1, y_2, \dots y_j, y_{j+1}, \dots \in S_p^k$. From the standpoint of the decoder, any non-prefix element $u \in S \backslash \{p\}$ encountered in the decoding of the string x must, according to the equation (3.2), mark a boundary between the

codeword pair $\ldots y_j y_{j+1} \ldots$ thus any non-prefix element is a synchronizing pattern, automatically establishing resynchronization at the decoder. This can be explained using an example.

Consider a binary T-code set $S = \{0,10,11,110,1110,1111\}$.

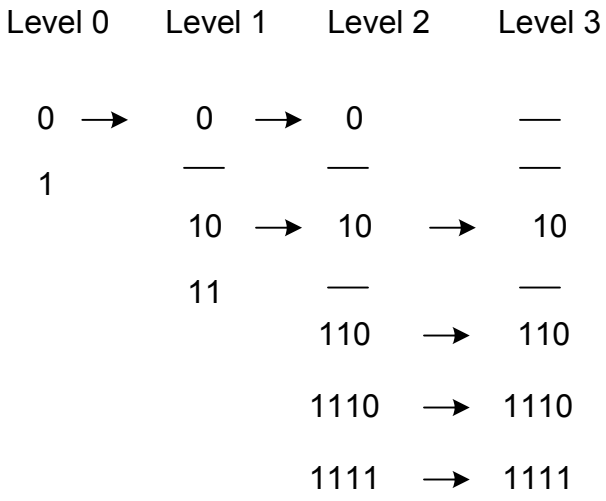

Figure 3.1.1: T-code synchronization tree

Following a synchronization loss, the decoder will revert itself to the lowest level code word state, level 0. Once in level 0, the decoder moves onto the next level upon the receipt of any non-prefix element $u \in S\backslash\{p\}$, where p is the element used as the prefix. In the above example the decoder moves to level 1 upon receipt of 0. Otherwise the decoder must remain in the current level. Thus any non-prefix element $u \in S\backslash\{p\}$ acts as a synchronizing pattern. Once the highest level of code word state (corresponding to the degree of augmentation of the final set) is reached, synchronization is assured. In the example, using the set $S(1,11,0)$, we assume that a synchronization loss is

occurred and the decoder is reverted to level 0. In level 0 a receipt of 0 will lead to a transition to level 1. In level 1 receipt of 10 or 11 will lead to a transition to level 2. In level 2, a 110, 1110 or 1111 will carry the decoder to next level. Now this is the highest codeword state, at which time the synchronization process will, with certainty, be known to be complete. From the example we can see that any non prefix element that belongs to the code set $S\backslash\{p\}$ acts as a synchronizing pattern, thus assuring resynchronization at the decoder.

3.1.2. Selection of T-Codes

When performing T-augmentation, choosing shorter code words as the prefix may not yield best results in terms of compression, because a tree suitable for uniform distribution is generated. Choosing longer code word as the prefix may not also yield good compression, because a tree suitable for skewed distribution is generated. Selection of best T-codes is an important step when using T-codes as entropy codes. In [51], Higgie proposed an algorithm to select database of best T-codes. These databases are derived using a complex algorithm that calculates the ASD of all the T-codes up to augmentation level of 8. In this work we proposed a novel T-code selection algorithm that is simple to implement.

Selection Algorithm: The novel selection algorithm was developed on the basis of observations made on T-augmentation. The key observations are

- (1) Choosing longest codeword for T-augmentation results in a highly skewed tree
- (2) Choosing the shortest codeword results in a balanced tree (equal-length code), suitable for uniform distribution
- (3) We can use an idea similar to Binary Search to select the best T-Code for a given input

Let L_i be average code length. The important steps that were implemented in our T-code selection algorithm are organized as

- Step 1: L_1 ← Compress input using (1);
- Step 2: L_2 ← Compress input using (2);
- Step 3: L_M ← Compress input using the median codeword for the T-augmentation;
- Step 4a: If ($L_1 <= L_M$) OR ($L_1 - L_M <= L_2 - L_M$)
  - Repeat (1) to (4) between L_M and L_1 (That is, use $L_1 = L_1$, $L_2 = L_M$)
- Step 4b: If ($L_2 <= L_M$) OR ($L_2 - L_M <= L_1 - L_M$)
  - Repeat steps 1 to 4 between L_M and L_2 (That is, use $L_1 = L_M, L_2 = L_2$)
- Step 4c: If ($L_1 = L_M = L_2$)
  - Stop. Use T-code at L_M as the coding tree.

Thus, at the stopping stage, we have only one code tree. The time taken would be logarithmic.

Figure 3.1.2 shows the ideal case for selection T-codes, where T-code set formed by using codewords of length L_M as prefix is selected.

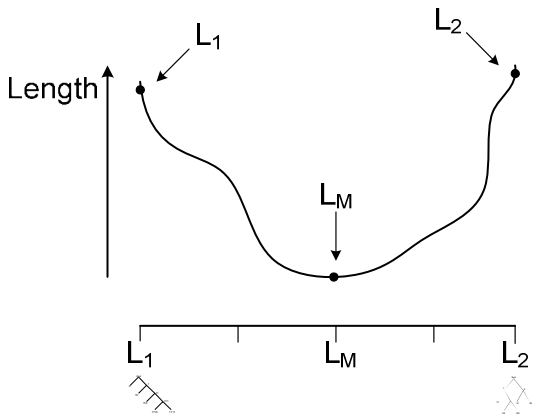

Figure 3.1.2: Ideal case for selection of T-codes.

Figure 3.1.3 shows a case where the range between L_1 and L_2 has been reduced by applying proposed T-code selection algorithm. The algorithm is recursively applied until L_1 and L_2 are converged to a single point L_M, which would be the best T-code set to use.

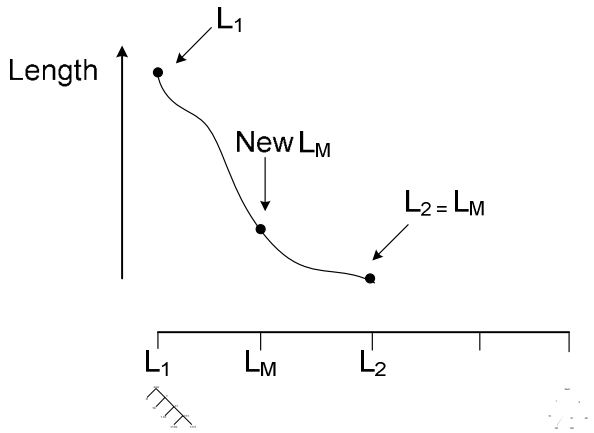

Figure 3.1.3: Reduced range after applying proposed algorithm

3.2. Huffman Codes

Huffman Coding is a prefix-free entropy coding algorithm invented by David A. Huffman in 1952 [57]. Huffman coding is known for its optimum compression efficiency for a symbol to symbol coding with a known probability distribution. Huffman codes are efficient when the probability of each symbol is negative power of two (i.e., dyadic distribution). The worst case performance of a Huffman code is found when the probability distribution is given by $p_i = 2^{-i}$, for the i_{th} symbol. Almost all the compression techniques including JPEG [1] uses Huffman coding for compression. Many researchers have extensively studied Huffman codes due to its best compression performance and thus various variations and modifications have been proposed [57], [62], [58], [59], [61], [60].

Binary Huffman Codes.

Huffman codes aim at achieving a minimum redundancy codes by minimizing the average message length. According to [57] the following restrictions are made to a code to achieve minimum redundancy

(1) No two messages will consist of identical arrangements of coding digits

(2) Once the starting point of a message sequence is known, no additional information should be required to know where a particular message code ends or begins.

(3) If there are N messages and P is the probability distribution of the N messages. The messages are assumed to be ordered in such a way that

$$P(1) \geq P(2) \geq P(3) \geq \cdots P(N-1) \geq P(N)$$

For a minimum redundant code the additional condition is

$$L(1) \leq L(2) \leq L(3) \leq \cdots L(N-1) = L(N)$$

(4) At least two of the messages with code length L(N) should have codes that are alike except for the last bit.

The design of basic binary Huffman codes follows the above restrictions to obtain a minimum redundancy code or optimum code [57]. According to the restriction (3) the least two probable messages should have equal length codes. Restriction (4) makes it necessary that the last digits of codes of Nth and (N-1)th messages are 0 and 1. Though the codes of the least probable messages

are undetermined it can be said that the codes of these messages are equivalent to a single message and its code will be the common prefix of order L (N-1) of these two messages. And its probability should be the sum of the two messages from which it was created. The least probable messages are then replaced with their composite message whose probability is the sum of the probabilities of the two messages. The messages are again ordered in descending order of their probabilities. The codes of each of the least probability messages are assigned to be 0 and 1. This technique is applied again and again until the number of messages becomes two. The sum of probabilities of these two messages should be 1 which can also be called as *root.* The code for each message can be obtained by tracing the path to the message from the *root.* The above explanation could be better understood with the help of an example. Let us assume that the messages are S= {A, B, C, D, E} with the probability distribution of {0.4, 0.25, 0.15, 0.1, 0.1}. The schematic of the method is shown in Figure 3.2

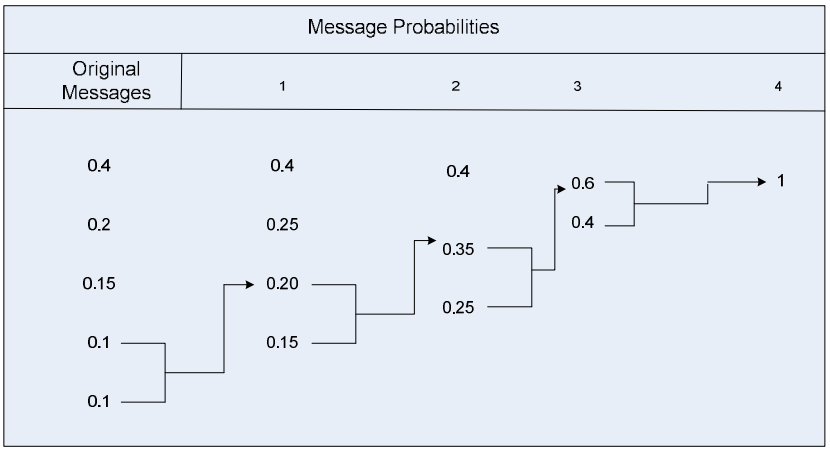

Figure 3.2: Huffman Coding Procedure.

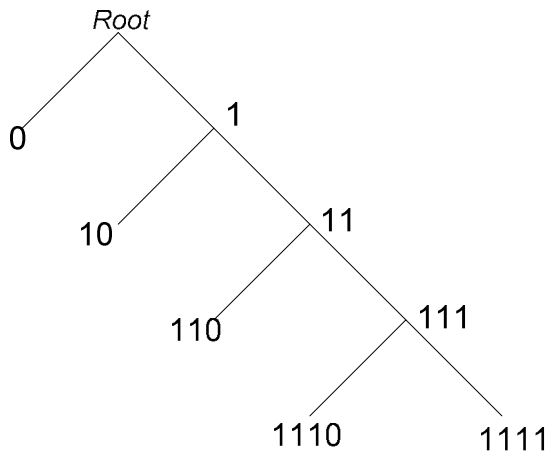

Figure 3.3: Huffman Tree Construction

Figure 3.3 shows the construction of a Huffman tree for assigning the codewords to each of the messages. The truncated leaf nodes are the generated Huffman Codes which are assigned to the messages or the symbols.

The Huffman table can be generated using the methodology explained.

S	P(i)	L(i)	P(i)*L(i)	*Code*
A	0.4	1	0.4	0
B	0.2	2	0.4	10
C	0.15	3	0.45	110
D	0.1	4	0.4	1110
E	0.1	4	0.4	1111

$L_{avg} = 2.05$

Table 3.2: Huffman Table

Table 3.2 shows the average length of a code in a Huffman encoded data.

Almost all VLCs exhibit synchronization properties up to some extent. Huffman's coding design makes it optimum as per compression efficiency is concerned, but provides no assurance on as to how the decoder reacts to the channel errors. The Performance of Huffman against T-codes in various aspects like Average Synchronization Delay (ASD), PSNR, MSE, MAE, Compression rate, and encoding-decoding times is studied in this work. The details of the work and the results obtained are presented in the next section of the chapter.

3.3. Details of the Comparative study

The efficiency of Huffman coding and T-codes are studied in this work. A comparative study is done in various aspects, especially in the self-synchronizing aspect. The study shows the behavior of both Huffman codes and t-codes in an erroneous environment. The various aspects studied are briefly discussed in this section.

Compression ratio or Compression rate: is an important parameter when measuring the efficiency of a compression scheme. In Image Compression, Compression rate is measured in terms of bits per pixel (bpp) Huffman coding gives optimum performance in terms of compression. T-codes also exhibit good performance in compression aspect, which is close to that of Huffman coding.

Encoding and Decoding Times: are another set of parameter that decides the applicability of a scheme in real life environment. The computational and operational complexity of encoder and decoder is a major factor when designing a model. Due to the design structure of T-codes, they show a higher encoding and decoding time when compared to that of Huffman coding.

Average Synchronization Delay: The main purpose of this study is to analyze the performance of T-codes over Huffman codes in the aspect of synchronization and hence error resilience. The efficiency of synchronization can be measured in terms of Average Synchronization Delay (ASD). Synchronization Delay is defined as the number of bits or symbols after which the decoder regains synchronization. G. R. Higgie studied T-codes from the ASD point of view in [56], in which he calculates ASD and standard deviation of SD when a single bit error is occurred in a compressed pdf data. In this work we have studied the ASD aspect of T-Codes in comparison with Huffman codes and also the variations of ASD against different parameters like Average length of the code, Different error rates, and Entropy are studied. T-Codes show a clear dominance over Huffman coding in this aspect. From this result comes the motivation for the usage of T-Codes in the application of multimedia transmission where synchronization property is very much needed, which is also one of the major contributions of this work.

Quality Analysis: The effect of superior performance in synchronization and thus error resilience is reflected in the decoded image at the decoder. Peak

Signal to Noise Ratio (PSNR), Mean Square Error (MSE), and Mean Absolute Error (MAE) are the three major parameters that indicate the closeness of the reconstructed image to the original image. All three performance parameters are studied against different error probabilities for both Huffman Coding and T-coding.

Visual Comparison: "*A Picture is worth a thousand words.*" The visual results tell us much more about the performance of an image transmission system. Reconstructed images were plotted against different error probabilities for both Huffman coding and T-Coding.

From the results obtained in this detailed study, it can be said that in situations where codeword synchronization is needed, T-Codes can be used instead of Huffman Codes without sacrificing much coding efficiency. A detailed analysis with the help of the results obtained in the study is shown in the next section of this chapter.

3.4. Results of the Study

The Main objective of this study is to demonstrate the superiority of T-Codes over Huffman Codes in synchronizing the decoder without much compromise in compression efficiency. The parameters used to analyze this performance are MSE, PSNR, MAE, ASD, and Compression Ratio.

In all the following equations x is the original image pixel and x' is the reconstructed image pixel. M and N are the dimensions of the Image

MSE: Mean square Error, MSE is calculated by using the formula shown below

$$\text{MSE} = \frac{1}{255^2}\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}(x_{m,n} - x'_{m,n})^2 \quad (3.3)$$

PSNR: Peak Signal to Noise Ratio, PSNR calculation is shown below

$$\text{PSNR} = 10\log_{10}\left(\frac{1}{MSE}\right) \quad (3.4)$$

MAE: Mean Absolute error, the formula to calculate MAE is given below

$$\text{MAE} = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}(|x_{m,n} - x'_{m,n}|) \quad (3.5)$$

The experiments were performed in a MATLAB 7.0.1 environment, using a PC running at 1.2 GHz, with 512GB of RAM.

3.4.1. Results for Encoding Time and Decoding Time

Table 3.3 shows the encoding time of both Huffman and T-codes calculated for three sizes of images. The time shown is in seconds. A matlab command *cputime* is used to calculate the time taken to encode and decode.

Test Images	Huffman	T-Codes
Lena 128x128	0.8934	1.2117
Lena 256x256	5.909	28.18
Lena 512x512	80.21	203.36
Baboon 128x128	3.57	4.27
Baboon 256x256	17.6	80.672
Baboon 512x512	300.66	409.825
CameraMan 128x128	0.728	2.21
CameraMan 256x256	5.306	29.4224
CameraMan 512x512	70,95	217.5266
Airplane 128x128	0.9293	2.141
Airplane 256x256	3.43	19.454
Airplane 512x512	71.172	249.849

Table 3.3 Encoding Time in seconds

Table 3.4 shows the decoding time for Huffman coding and T-codes.

Test Images	Huffman	T-Codes
Lena 128x128	3.216	5.33
Lena 256x256	5.323	28.18
Lena 512x512	35.6513	58.626
Baboon 128x128	3.939	6.66
Baboon 256x256	11.451	35.691
Baboon 512x512	89.272	113.75
CameraMan 128x128	1.4921	4.74
CameraMan 256x256	5.4178	19.808
CameraMan 512x512	33.538	62.28
Airplane 128x128	1.702	5.608
Airplane 256x256	5.63	19.65
Airplane 512x512	31.865	66.31

Table 3.4 Decoding Time in seconds

3.4.2. Results for Compression Rate

Compression Ratio is the measure of compression efficiency of a coding scheme. It is defined as

CR = (Original Image Size) / (Compressed Image Size). The Higher the compression ratio, the higher is the compression efficiency. Huffman codes are known for their optimum compression efficiency. In table 3.5 compression ratios of Huffman codes, T-codes (shortest code word prefix), T-codes (Proposed Selection Algorithm) and T-codes (Longest code word prefix) are shown.

Test Images	Huffman	T-Codes (Shortest Code)	Proposed	T-Codes (Longest Code)
Lena 128x128	7.0965	2.6748	6.2451	1.4738
Lena 256x256	9.578	4.9354	7.9859	2.649
Lena 512x512	11.543	5.852	8.9369	3.1846
CameraMan 128x128	8.2805	3.9143	6.8452	1.8634
CameraMan 256x256	9.1853	4.1709	7.5114	2.7301
CameraMan 512x512	11.5729	4.9657	9.4605	3.0743
Airplane 128x128	7.037	2.492	5.8278	1.2795
Airplane 256x256	9.003	4.1989	7.4144	2.3075
Airplane 512x512	11.595	5.09	9.3808	3.1639

Table 3.5 Compression Ratio

3.4.3. Results for PSNR, MSE, MAE

PSNR, MSE and MAE are the quality measures for the reconstructed image. PSNR and MSE show a similar picture of the quality performance. MAE shows the absolute difference between the original image and the reconstructed image. Figures 3.4.1 - 3.4.4 show the plot of PSNR vs. Error Rate. The plot shows the variations of both Huffman and T-codes. It can be observed that the T-Codes show better PSNR when compared to Huffman codes.

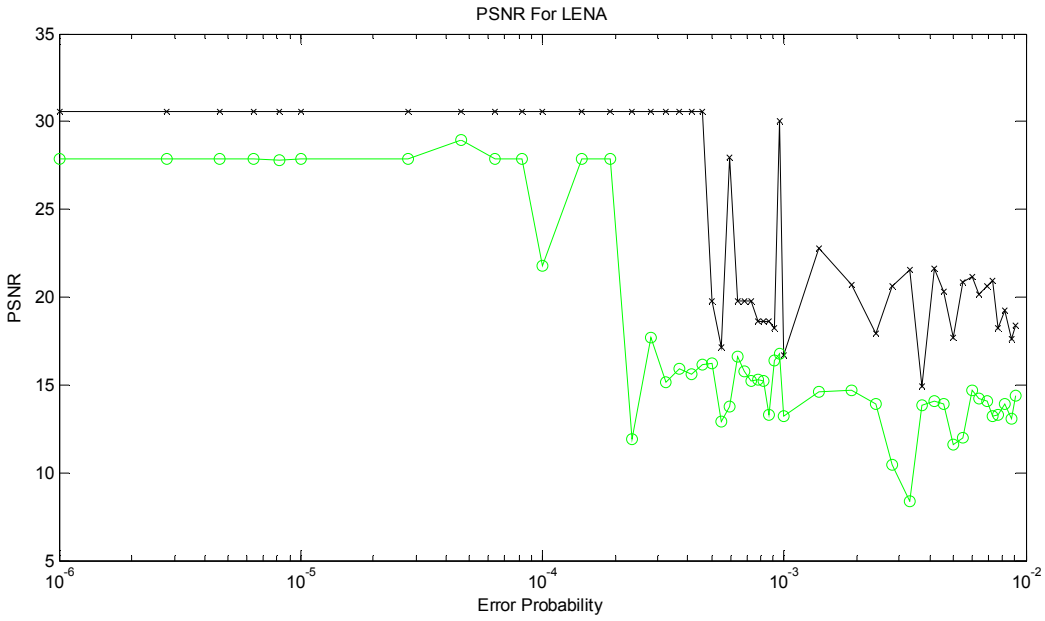

Figure 3.4.1 Variation of PSNR with Error Rate for Lena

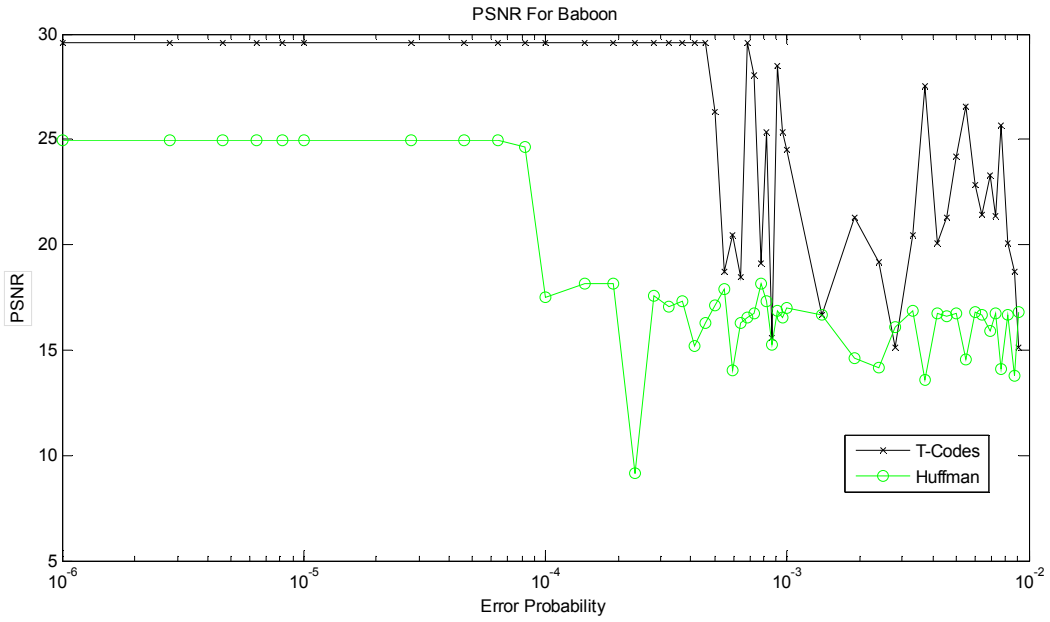

Figure 3.4.2 Variation of PSNR with Error Rate for Baboon

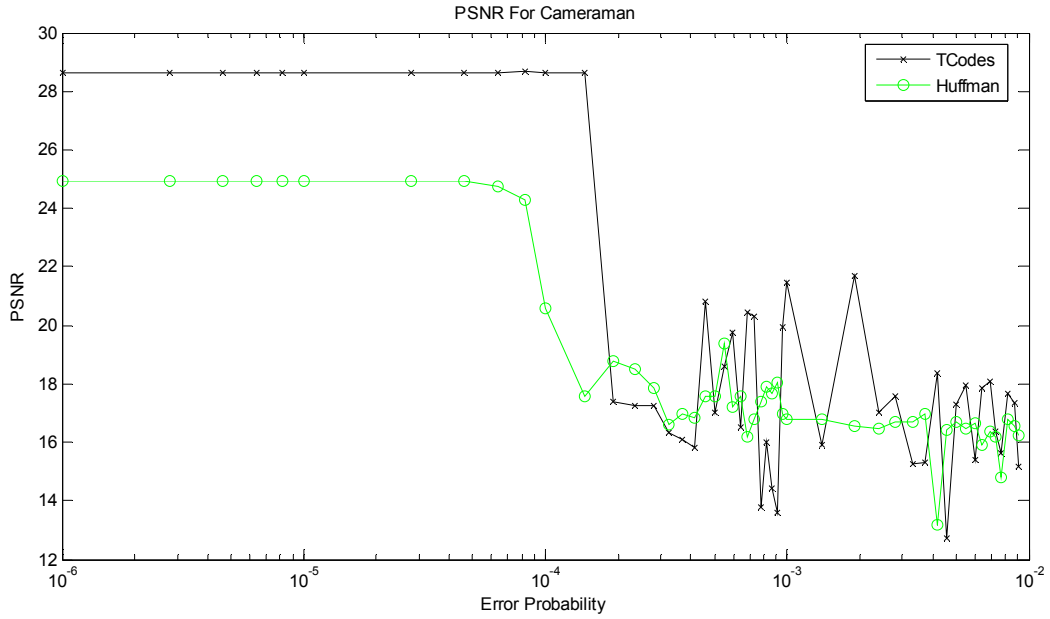

Figure 3.4.3 Variation of PSNR with Error Rate for Cameraman Image

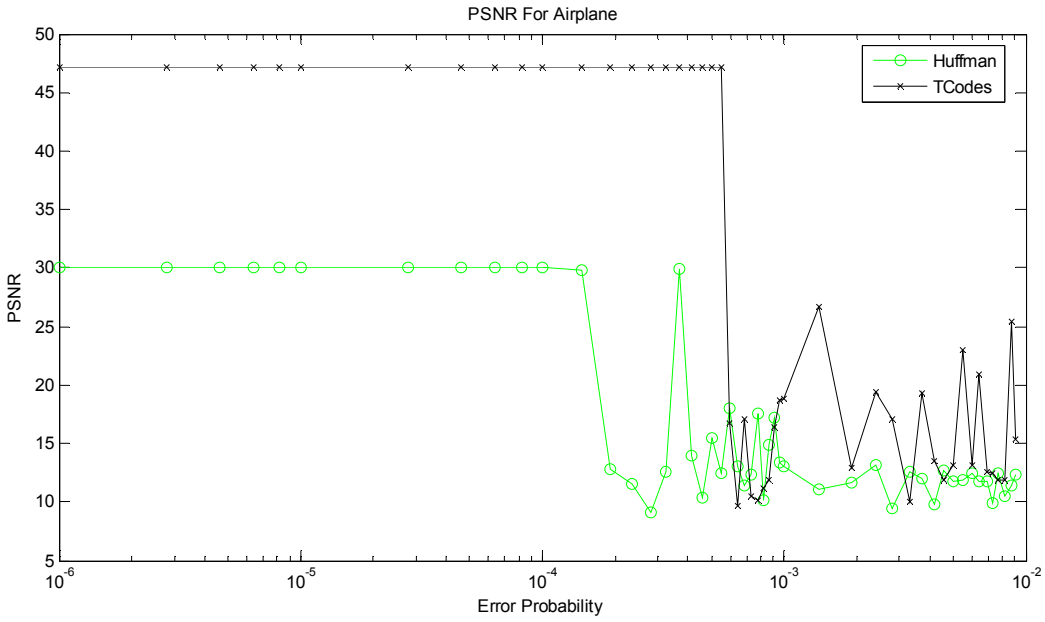

Figure 3.4.4 Variation of PSNR with Error Rate for Airplane Image

Figures 3.5.1-3.5.4 shows the plot of MSE against the Error rate for both the schemes.

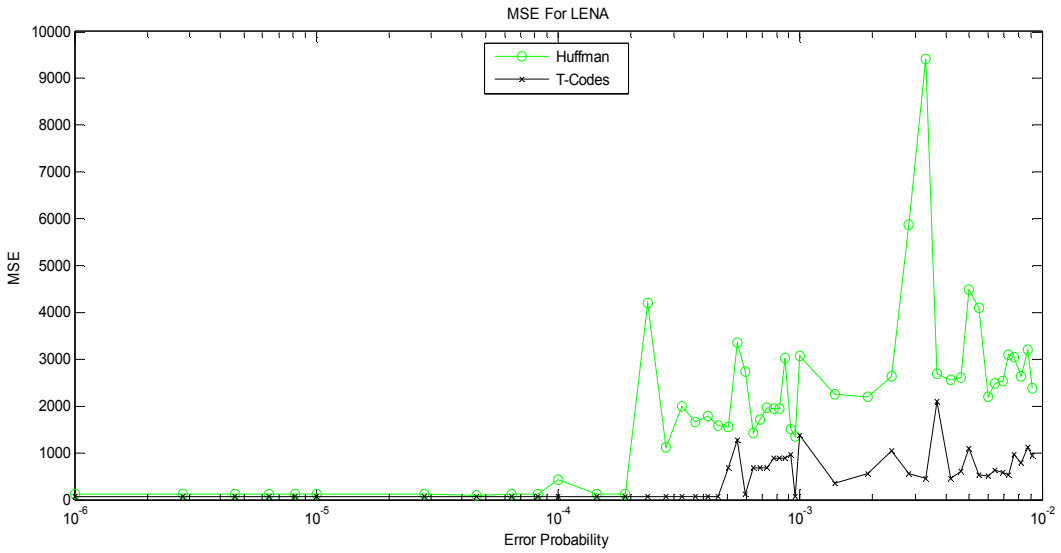

Figure 3.5.1 Variation of MSE with Error Rate for Lena Image

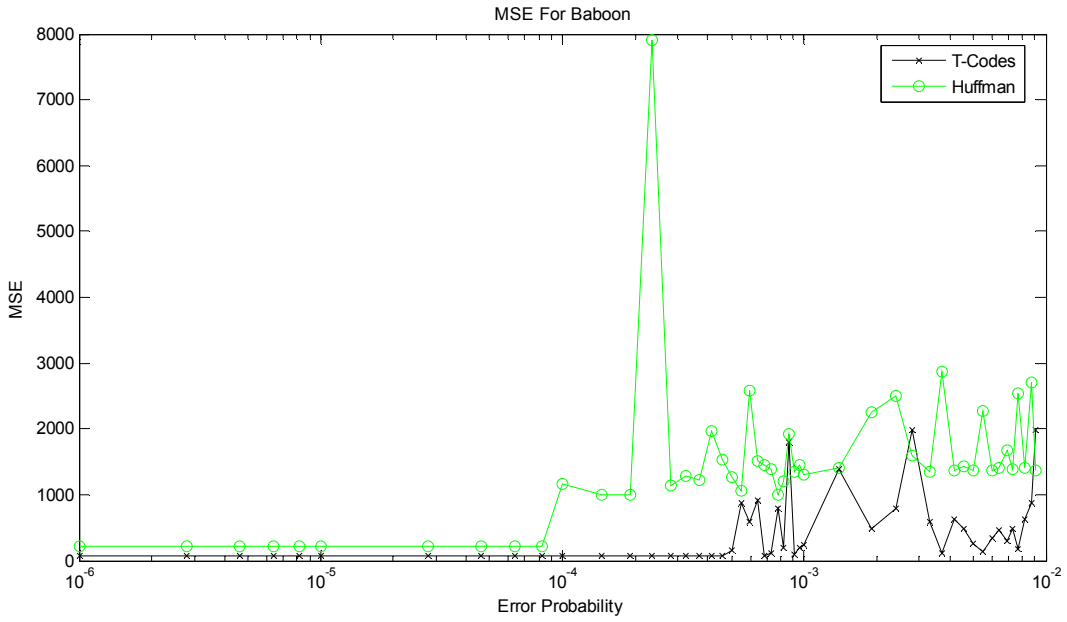

Figure 3.5.2 Variation of MSE with Error Rate for Baboon Image

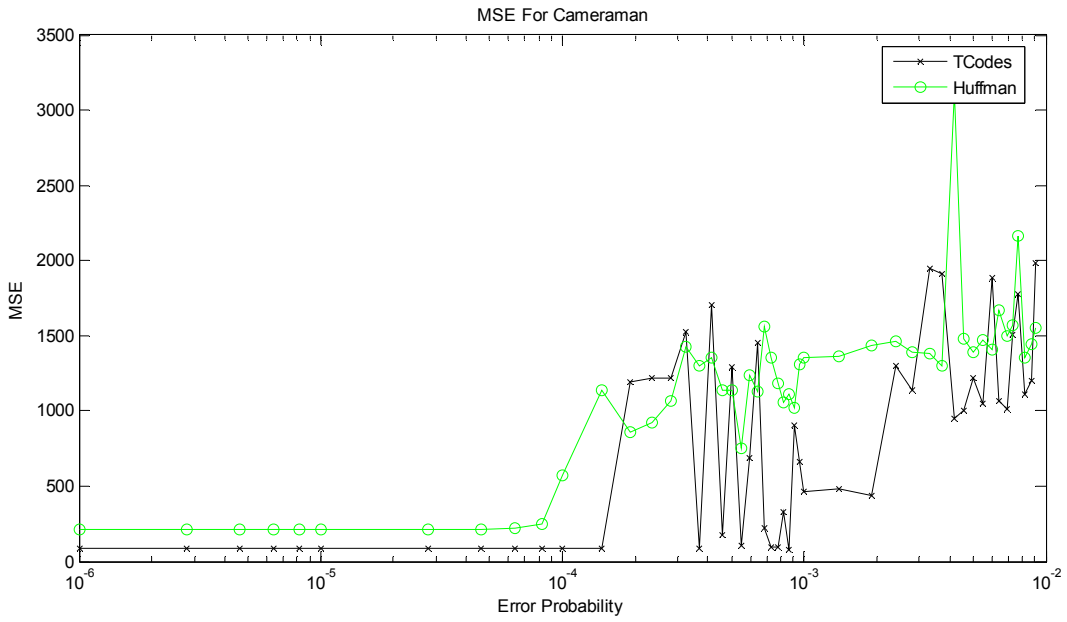

Figure 3.5.3 Variation of MSE with Error Rate For Cameraman Image

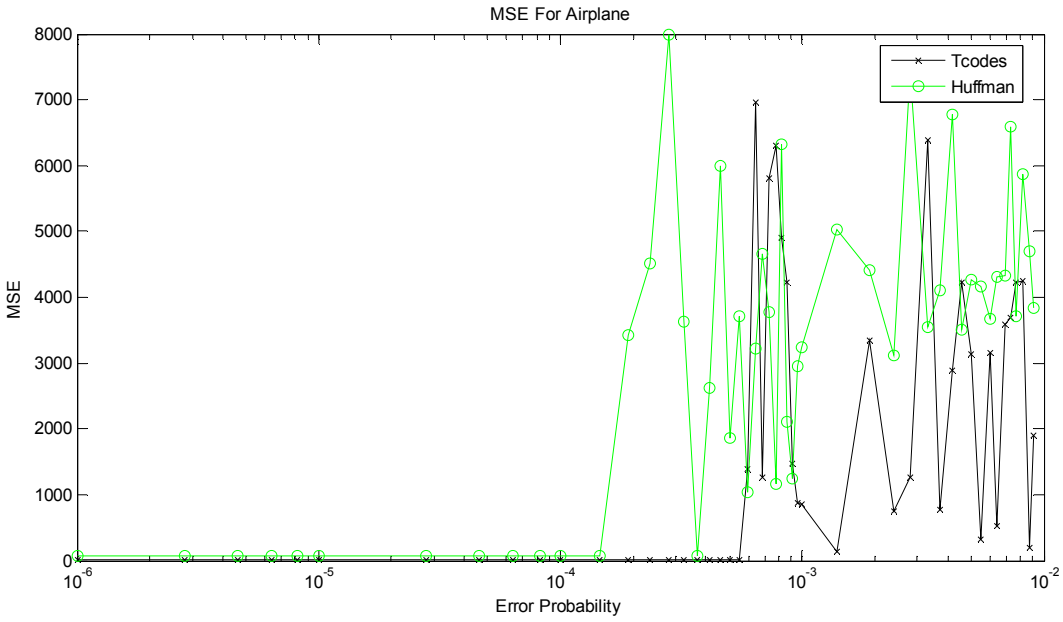

Figure 3.5.4 Variation of MSE with Error Rate For Airplane Image

Similar to the PSNR response, T-Codes show better MSE in comparison with Huffman Codes. Figure 3.6.1-3.6.4 shows the plot of MAE vs Error rate for both Huffman and T-codes.

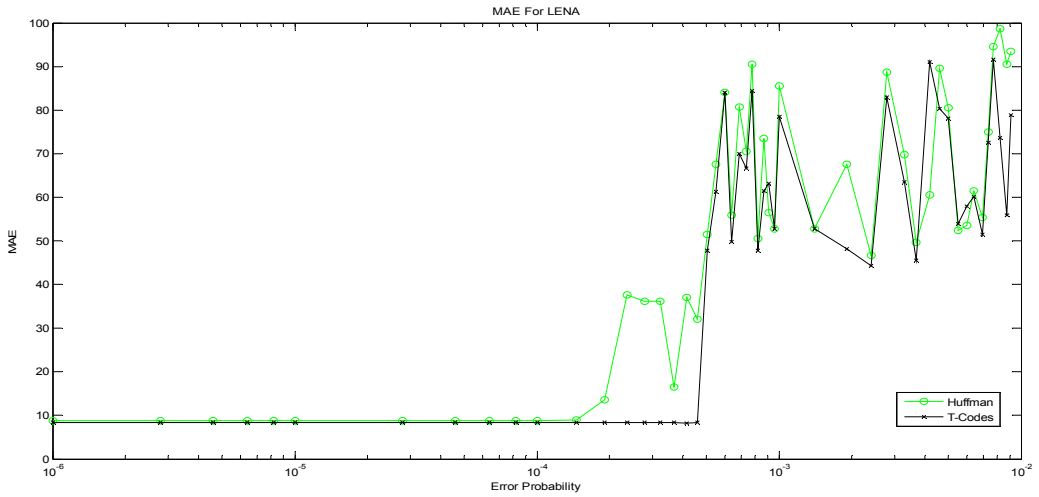

Figure 3.6.1 Variation of MAE with Error Rate for Lena Image

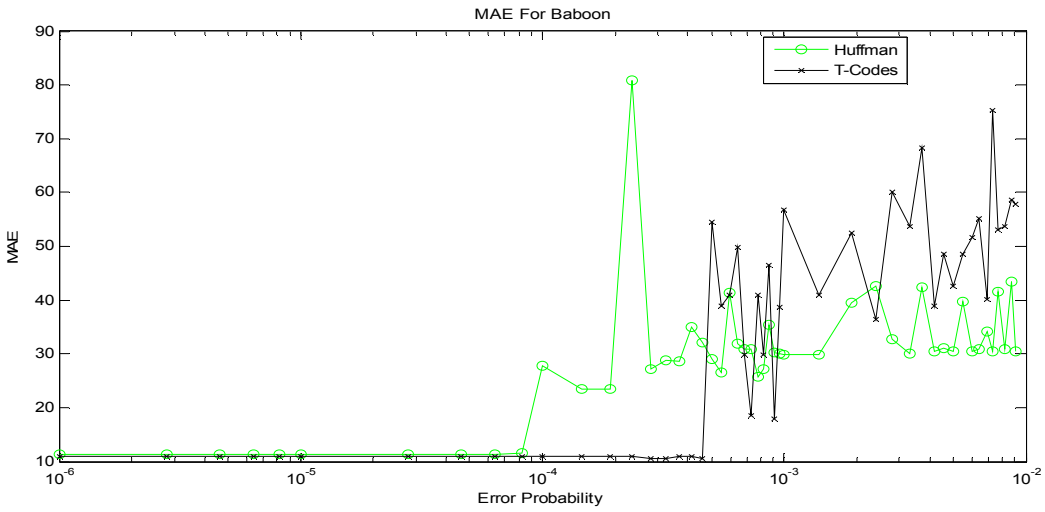

Figure 3.6.2 Variation of MAE with Error Rate for Baboon Image

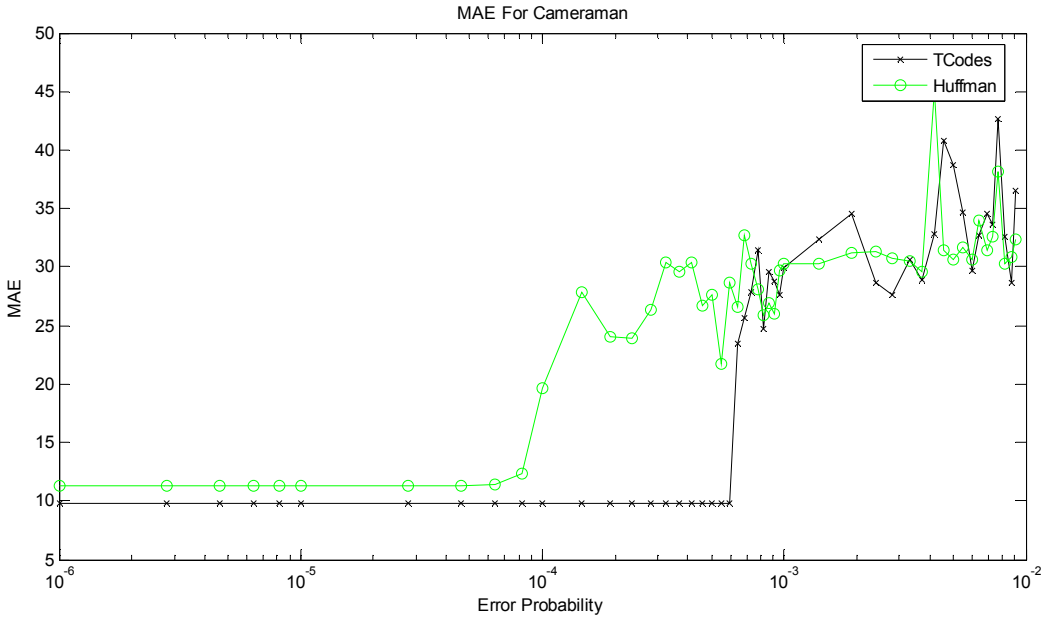

Figure 3.6.3 Variation of MAE with Error Rate For Cameraman Image

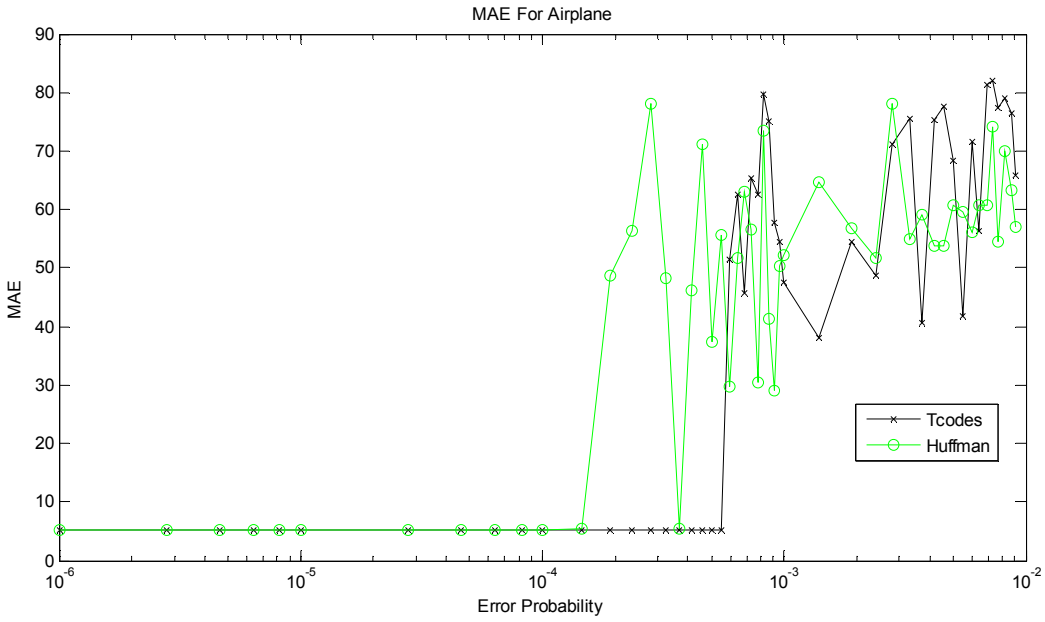

Figure 3.6.4 Variation of MAE with Error Rate For Airplane Image

By observing the above results one can say that the performance of T-Codes is good when compared to the standard Huffman codes.

3.4.4. Results for Average Synchronization Delay

Synchronization delay is defined as the number of symbols taken for the decoder to achieve synchronization. Average synchronization delay is a measure for the synchronization performance of a coding scheme. The average synchronization delay (ASD) depends on the error rate. Higher ASD is observed at higher error rates. This can be seen in the results shown in this section. The following results show that T-Codes offer better synchronization properties than that of Huffman. It can be observed that T-codes offer good synchronization at higher error rates, where Huffman starts degrading. Figures

3.7.1-3.7.4 show the plots of ASD versus error rate, both Huffman and T-Code behavior is shown.

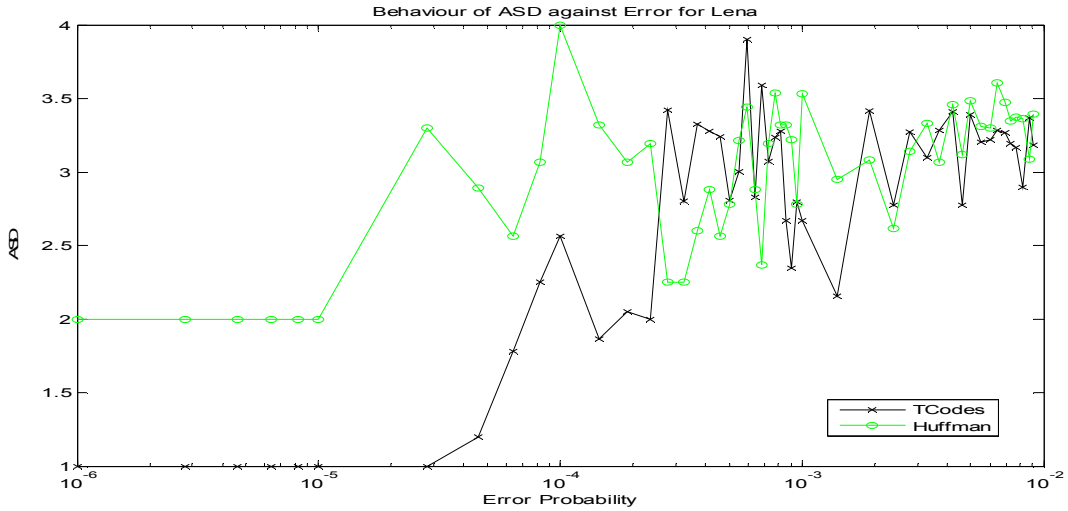

Figure 3.7.1 Behavior of ASD against Error rate for Lena Image

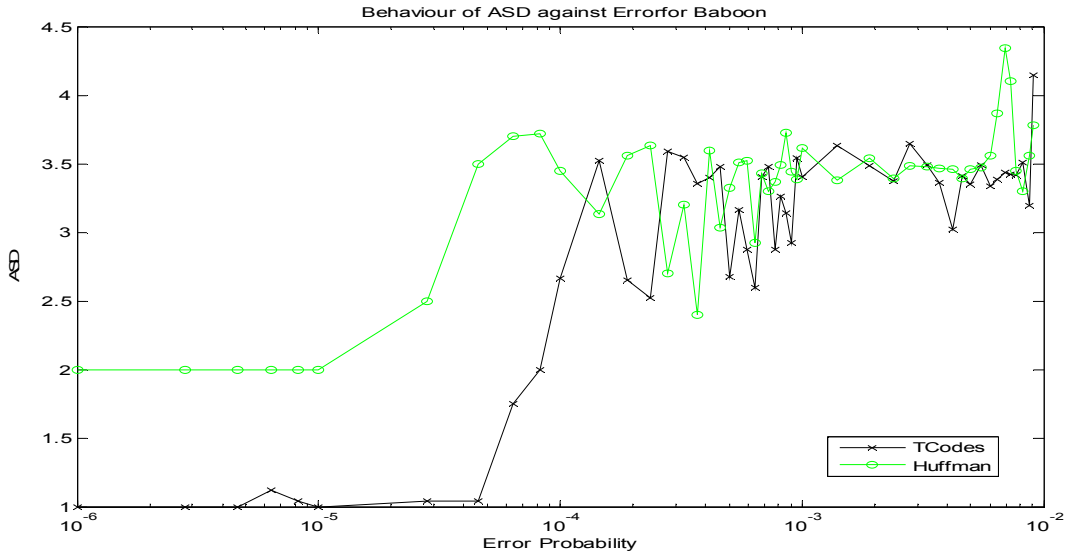

Figure 3.7.2 Behavior of ASD against Error rate for Baboon Image

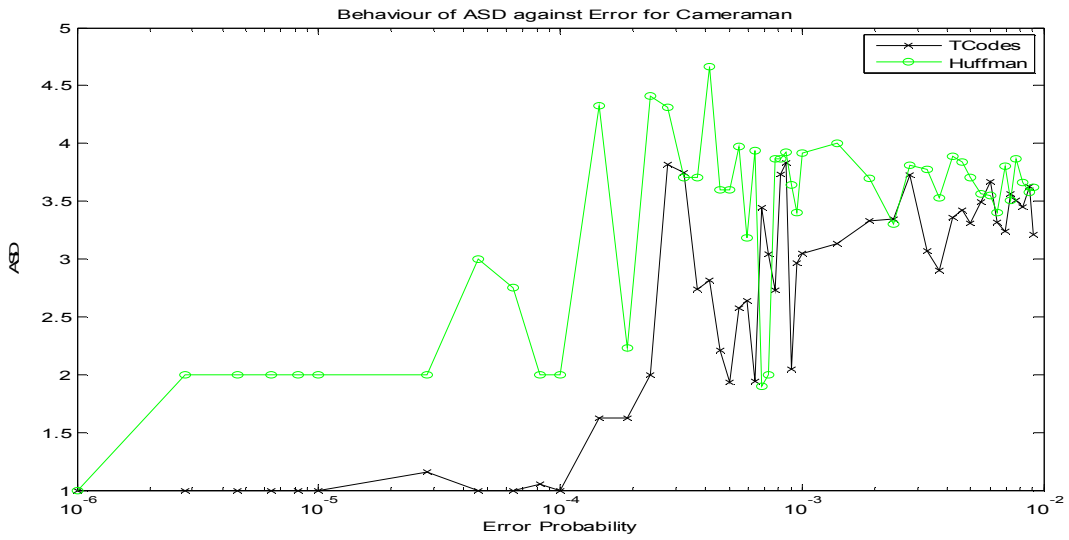

Figure 3.7.3 Behavior of ASD against Error rate for Cameraman Image

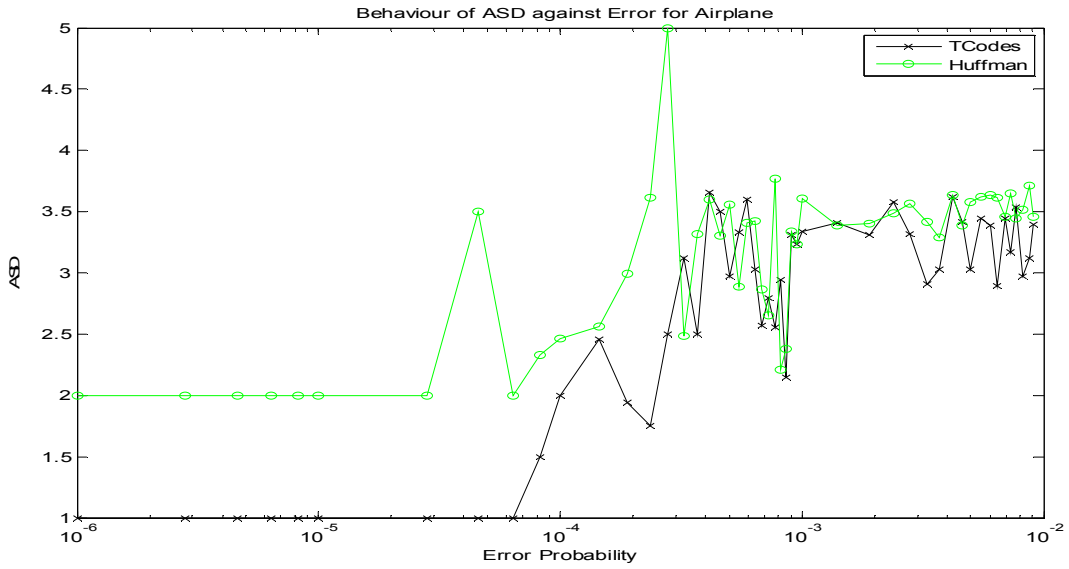

Figure 3.7.4 Behavior of ASD against Error rate for Airplane Image

3.4.5. Visual Results

For any kind of image coding scheme, visual analysis is very necessary because things which cannot be explained using mathematical analysis can be easily analyzed by visual perception. The following figures show some reconstructed images reconstructed from the two schemes in focus, Huffman codes and T-Codes. It can be observed from the figures that T-Codes are more resilient against errors than Huffman codes. Figures 3.8.1-3.8.4 shows the images of Lena that were reconstructed using both the schemes at different error rates.

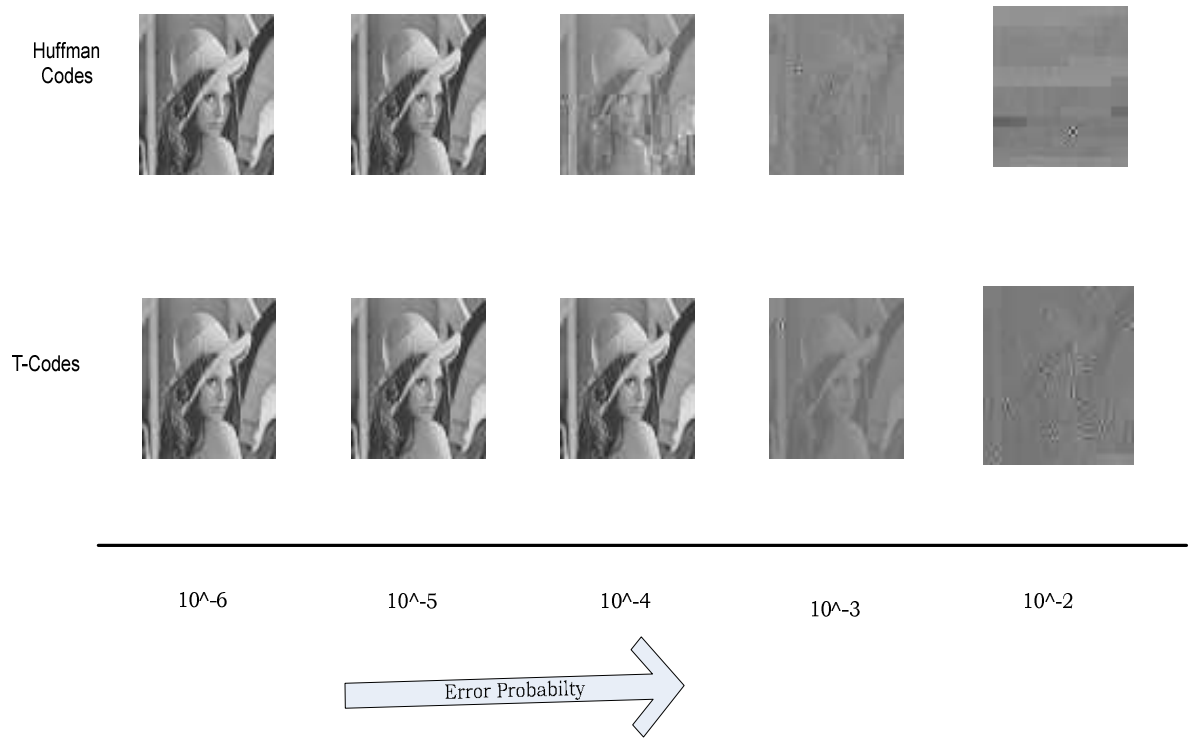

Figure 3.8.1 "Lena" Reconstructed by Huffman coding and T-coding at different error rates

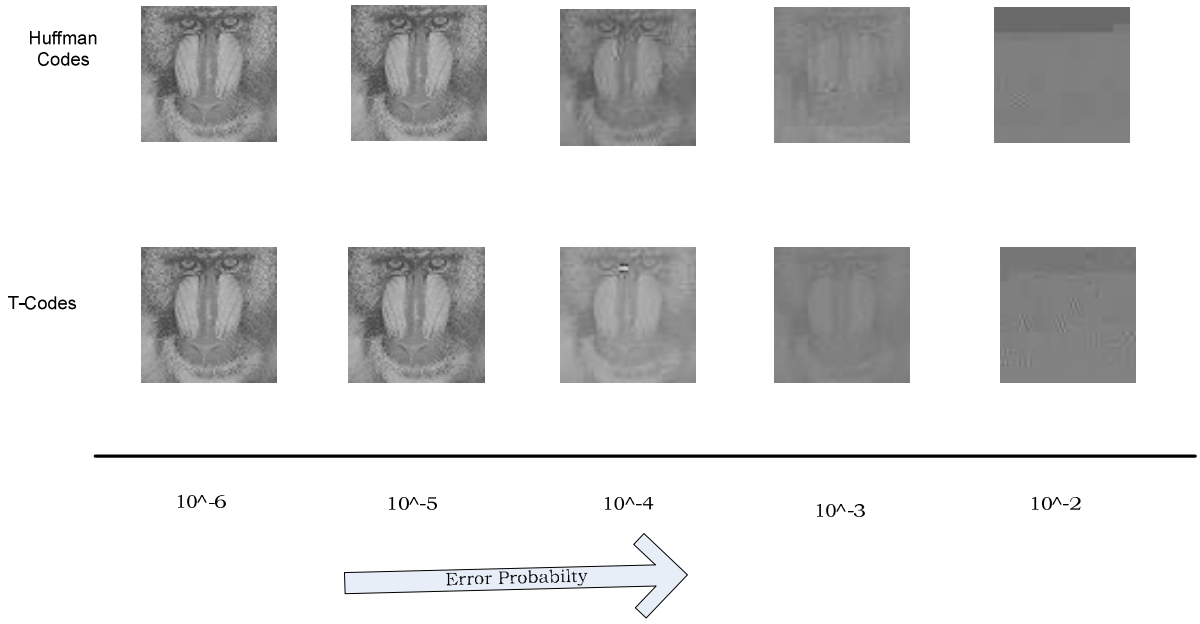

Figure 3.8.2 "Baboon" Reconstructed by Huffman coding and T-coding at different error rates

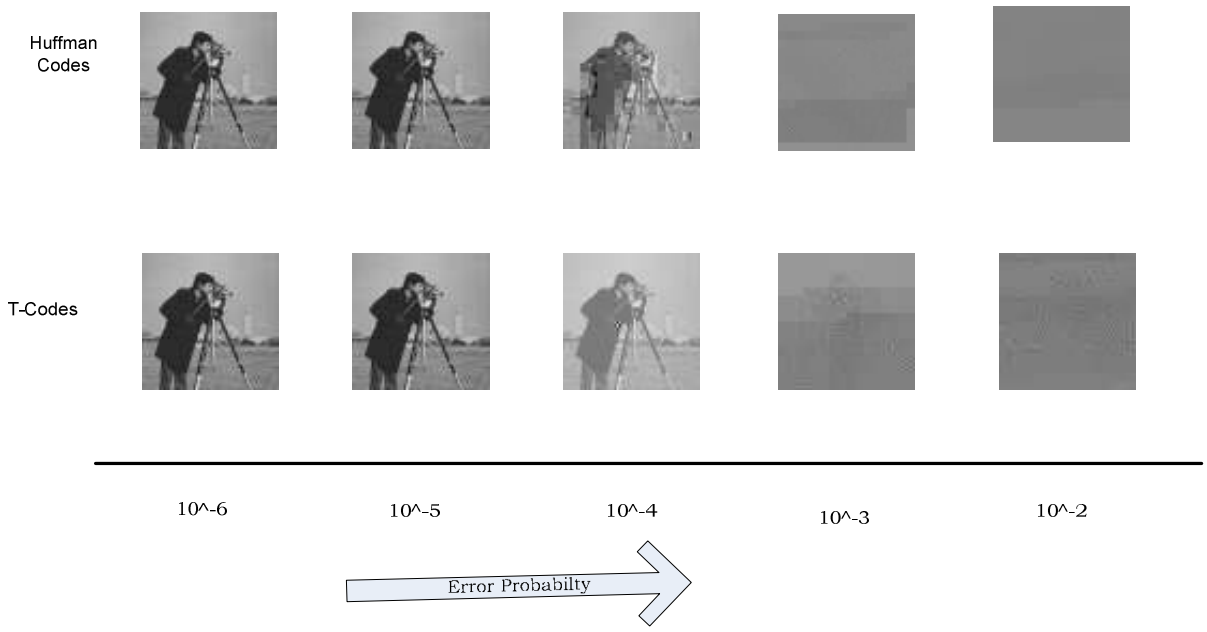

Figure 3.8.3 "Cameraman" Reconstructed different error rates By Huffman Codes and T-Codes

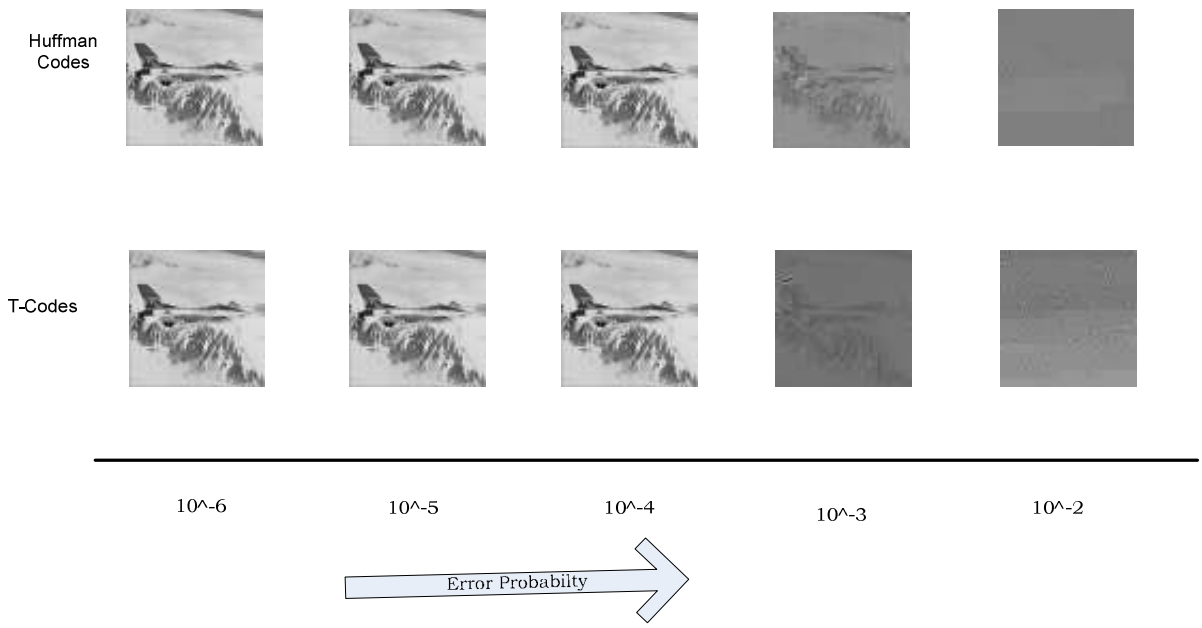

Figure 3.8.4 "Airplane" Reconstructed different error rates by Huffman Codes and T-Codes

Chapter 4

Error Resilience Using Edge-Based Data Hiding

4.1. Data Hiding Concept

Data hiding is the process of hiding some side information invisibly into host data. Data hiding has been used in many applications like covert communications, copyright protection, and content authentication [37], [39], [8]. Error concealment in multimedia communication has become a new application of data hiding in recent years.

In general original image data is unavailable at the decoder side; the only data available would be the correctly received data around the damaged data. If we could estimate some features from the correctly received data before-hand, it would be useful for a better concealment of the damaged blocks. The purpose of data hiding in error concealment is to hide some important features of the original data into itself and using them at the decoder end for concealing the error. Many methods were proposed on how to hide the data into the host signal [44], [45]. In our approach, we use *'odd-even'* method [49], in which the non-zero low frequency AC coefficients are modified to achieve data hiding. If

the data to be embedded is '0' then the AC coefficient is modified to an even number, otherwise it is modified into an odd number. It can be generalized as

$$p_i = hide\,(q_i, r_i)$$

$$= \begin{cases} q_i + 1, & if\ q_i > 0\ and\ (q_i - r_i) mod 2 \neq 0 \\ q_i - 1, & if\ q_i < 0\ and\ (q_i - r_i) mod 2 \neq 0 \\ q_i, & Otherwise \end{cases}$$

here p_i and q_i are i_{th} AC coefficients of the received and original images respectively, r_i is the message bit to be embedded. We have chosen to use the *'odd-even'* method mainly because of its simplicity and flexible nature.

Usually in all the previous works proposed on data hiding (for example see [11]), features extracted from a block are embedded into its surrounding block and is transmitted. Since the transmitted bit stream could experience a lot of channel errors, the hidden data may also get corrupted in the process. This may result in improper detection of the hidden feature and thus improper correction of error. In our approach we propose a novel scheme, in which we embed features of a block into three other blocks which are far away from each other using a novel data scrambling method. This way even if hidden data from one of the blocks is damaged, it can be retrieved from the other two blocks. This data scrambling method is also used in the detection of error which is explained in detail in the next section.

To implement the novel data scrambling method, we created a *scramble table* that shows, for each block in an image, the set of three blocks in which data of the particular block should be embedded.

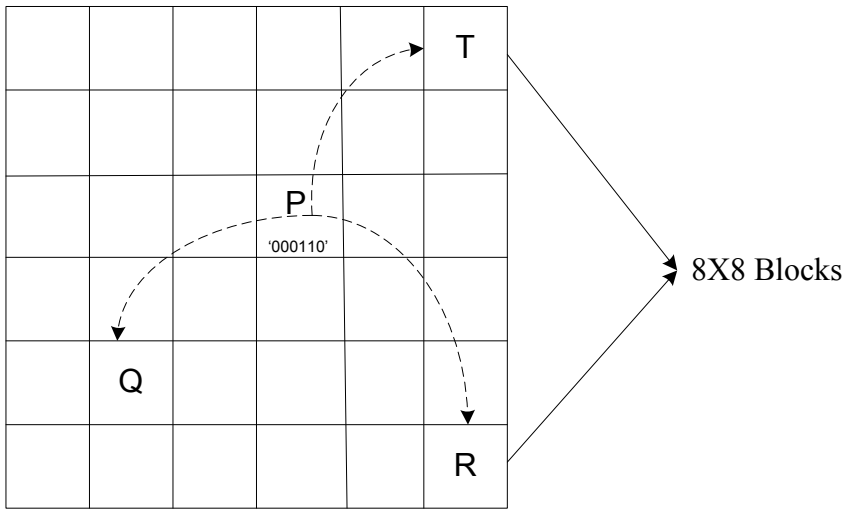

Figure 4.1 Novel Scrambling technique

Figure 4.1 shows an example of the data scrambling approach. Let the feature extracted from an 8x8 block 'P' be assigned a binary code '000110'. By looking up the *scramble table* it can be determined that the set of three blocks in which the data from 'P' is to be embedded are Q, R, and T. As you can see the scramble table is created in such a way that all the three blocks in a set are far from each other. This way a burst error involving one of the blocks does not affect the other two blocks. Finally after the features of all the blocks have been

embedded into other blocks, each block now holds hidden information of three other blocks. Figure 4.2 shows a view of a block after the scrambling and embedding process.

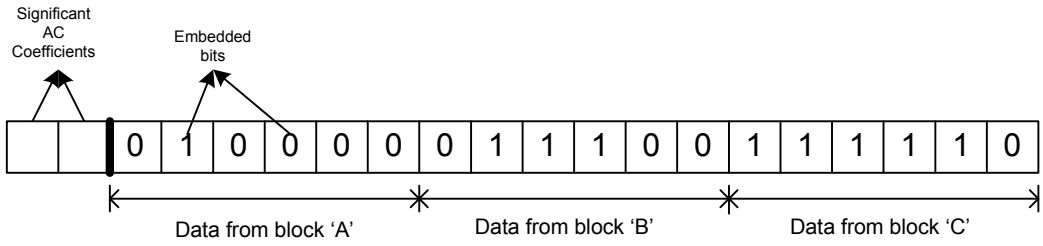

Figure 4.2 Example of a DCT block with embedded bits.

The first two AC coefficients are left untouched when hiding the data because they are assumed to contain most of the information of all the AC coefficients. In our approach we use 6 bits to represent the feature of the block, as shown in Figure 4.3.

4.2. Edge-Based Error Detection

In the above discussion we referred to embedding a particular 'feature' of a block into other blocks. It is very important to determine what kind of feature we need to embed in order to achieve a better error detection and error concealment. A natural image consists of high variations in intensities especially due to edges etc. and ordinary bilinear interpolation may not yield better results when interpolated across edges. A bilinear interpolation which is

edge directed, that is, interpolation along the edge direction yields better results [5]. A prior knowledge of edge direction will thus improve efficiency of concealment. So we consider edge information as the 'feature' to be embedded.

Error detection is a major step before going into concealment of the error. A good error detection scheme promises better concealment. In our approach we make use of the same embedded data to detect the errors and to conceal the errors.

Edge Pattern Classification: Firstly, we know that edges may be in different directions, so our approach makes use of an edge-pattern classification scheme that classifies edges into different classes based on their direction. The different edge patterns considered in this work are shown in figure 4.3.

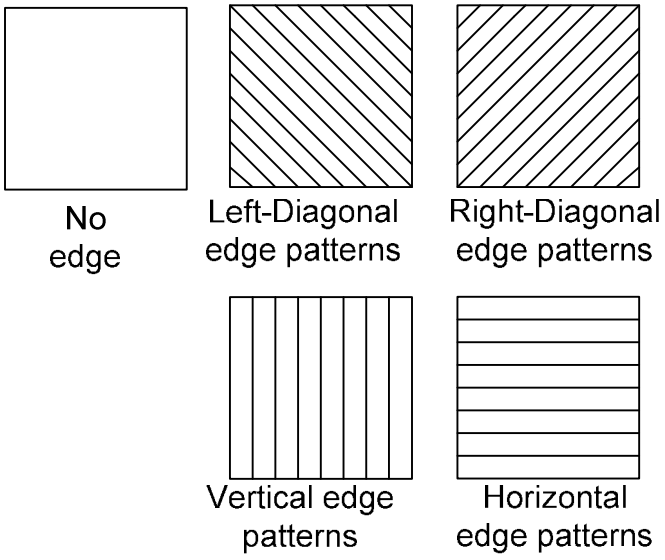

Figure 4.3 Different types of edge patterns

Each edge pattern is assigned a binary code. Edge patterns of all the blocks in an image are estimated and the corresponding binary code is then embedded into three different blocks using our novel scrambling algorithm. The image is then transmitted over a potentially noisy channel where it gets degraded due to the channel errors. After receiving the image, the decoder starts to de-embed the hidden data from all the blocks. The decoder can de-embed all the three copies of the embedded data of a particular block. By using the scramble table, the decoder can determine the locations of the three copies of embedded data of a particular block. The decoder compares all the three copies and determines the edge direction of the block by taking the majority of the copies. Figure 4.4 shows the general block diagram of edge-based data hiding.

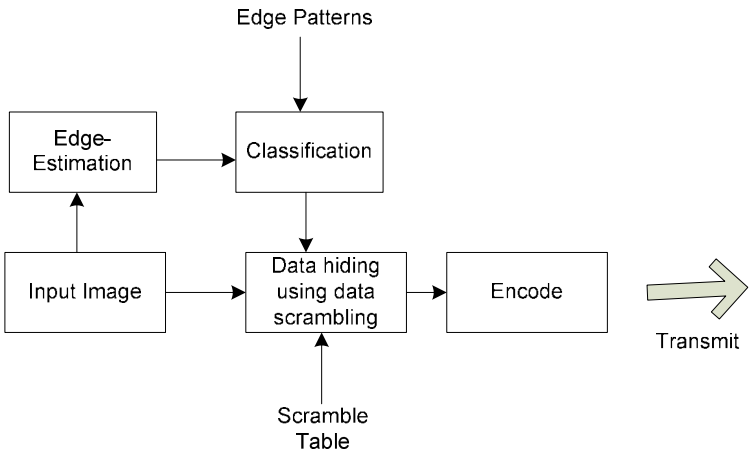

Figure 4.4 Edge-Based Data Hiding

For example, let us consider that the edge information of a block 'P', '000100', has been embedded into a set of three blocks Q, R, and T. Suppose a block, say, Q has been damaged due to the channel error and R, T are correctly received, as shown in Figure 4.5. The de-embedded information obtained from Q will be different from that of R and T. So the decoder determines that the block Q is an erroneous block, accomplishing an important task of error detection.

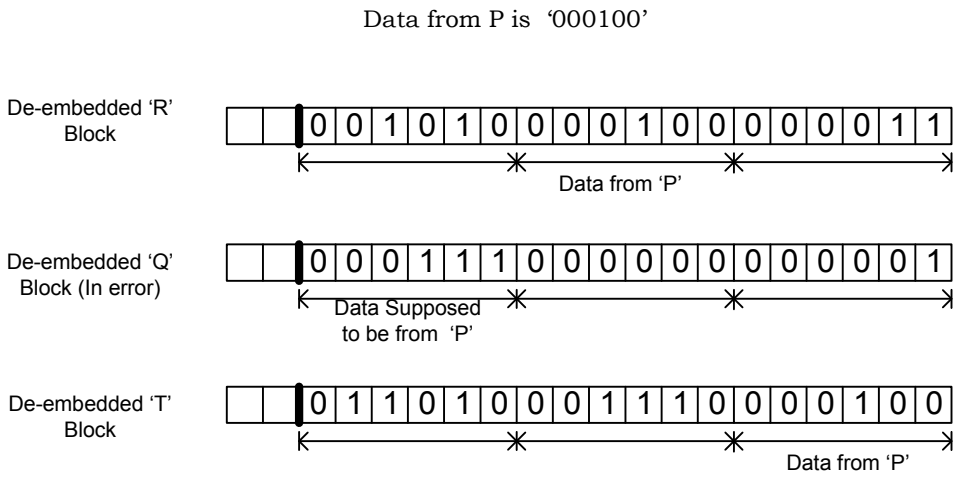

Figure 4.5 Error detection by comparison of the three copies

To determine the edge direction of block 'P', the decoder uses a majority rule between the three copies of embedded data obtained from Q, R, and T. since R and T are received correctly, the de-embedded data should also match with each other. The decoder now can say that the de-embedded data from R and T corresponds to the edge direction of the block 'P'. Here we assume that only

one of the blocks that carry the same embedded information is damaged. The process can be illustrated using a flowchart as shown in Figure 4.6. The Flow chart shows a sequence of operations performed by the decoder to detect and correct errors.

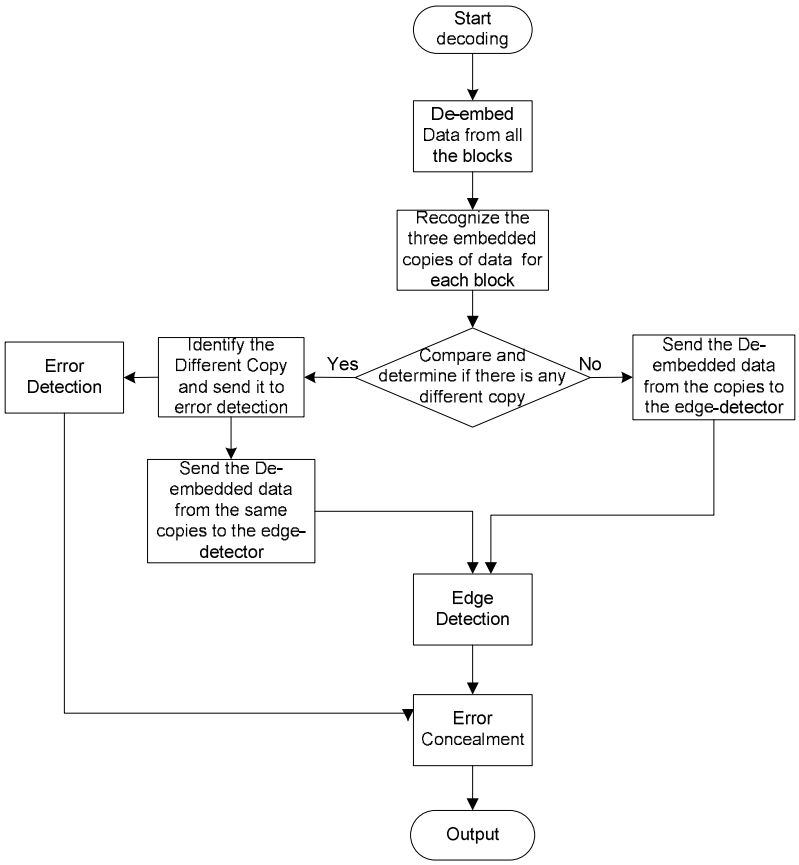

Figure 4.6 Flow chart of Decoder operations.

4.3. Edge-Based Interpolation

After the successful detection of errors, the next step would be concealing the errors. The most basic technique used to conceal an error is bilinear interpolation in the spatial domain. In this method the lost pixel is interpolated using the four nearest neighbors in all four directions. The averaging is done on weight basis; the weights are dependant on the respective distances of the neighboring pixels from the lost pixel. Closer pixels are given more weightage. However, due to high variations of intensities in an image, ordinary spatial interpolation does not always produce good results, especially across edges. Therefore interpolation is done in the direction of an edge. Figure 4.7 shows the basic bilinear interpolation and basic directional interpolation.

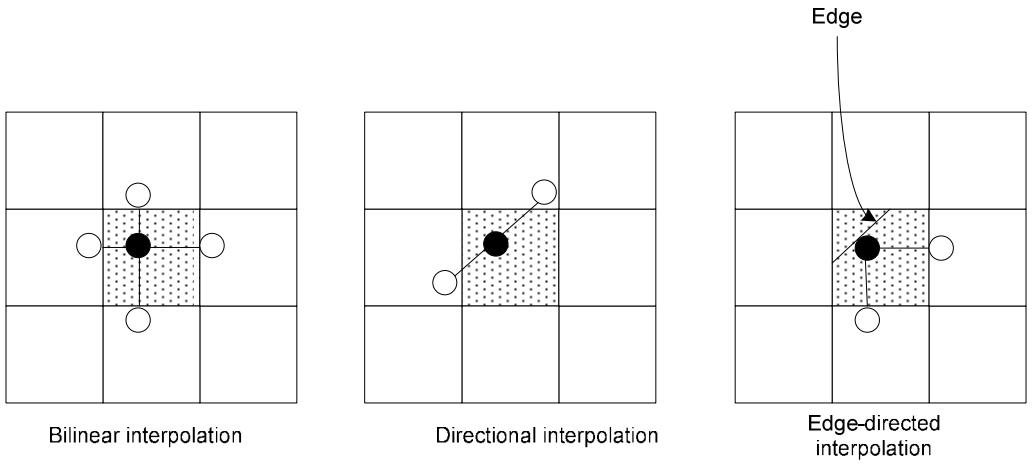

Figure 4.7 Basic bilinear interpolation, directional interpolation and Edge-directed Interpolation

In Figure 4.7, the dark circle corresponds to the pixel to be interpolated and the hollow circle indicates pixels used for interpolation. As explained in the previous section the edge information is embedded into the AC coefficients of the original image and is transmitted over a channel. At the receiver end, the hidden data is retrieved and is utilized in the error concealment. The block diagram of the proposed approach is shown in Figures 4.8 and 4.9.

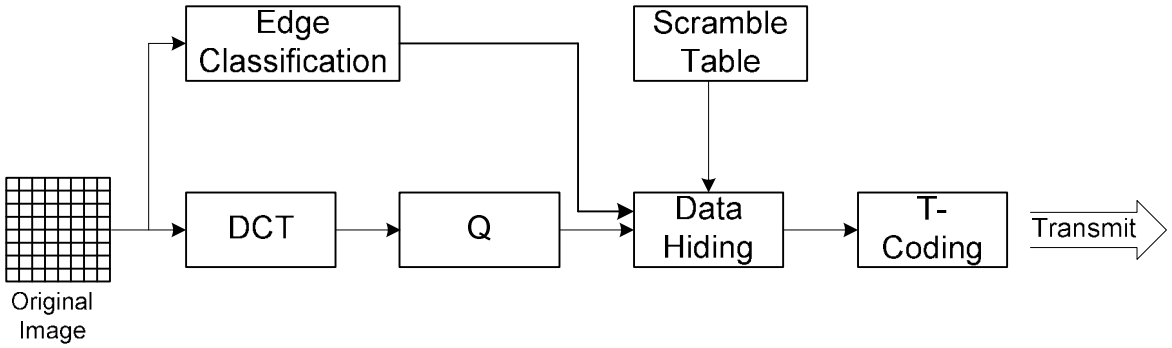

Figure 4.8 Encoder end operations

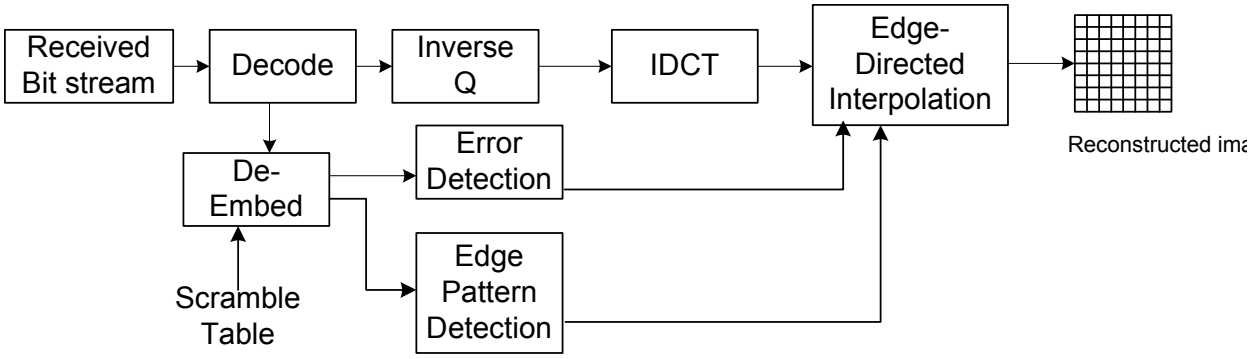

Figure 4.9 Decoder end operations

In our approach we use *odd-even* data hiding technique to embed edge information at the encoder. The error detection is done using a novel scheme explained in the previous section. Once the error detection is done, edge blocks are identified from the erroneous blocks and their corresponding edge direction is obtained from the hidden data. The error concealment is done using directed interpolation along the edge. Erroneous smooth blocks are corrected using normal bi-directional interpolation technique. Each pixel in the erroneous block is interpolated depending on the edge information in as shown in Figure 4.10

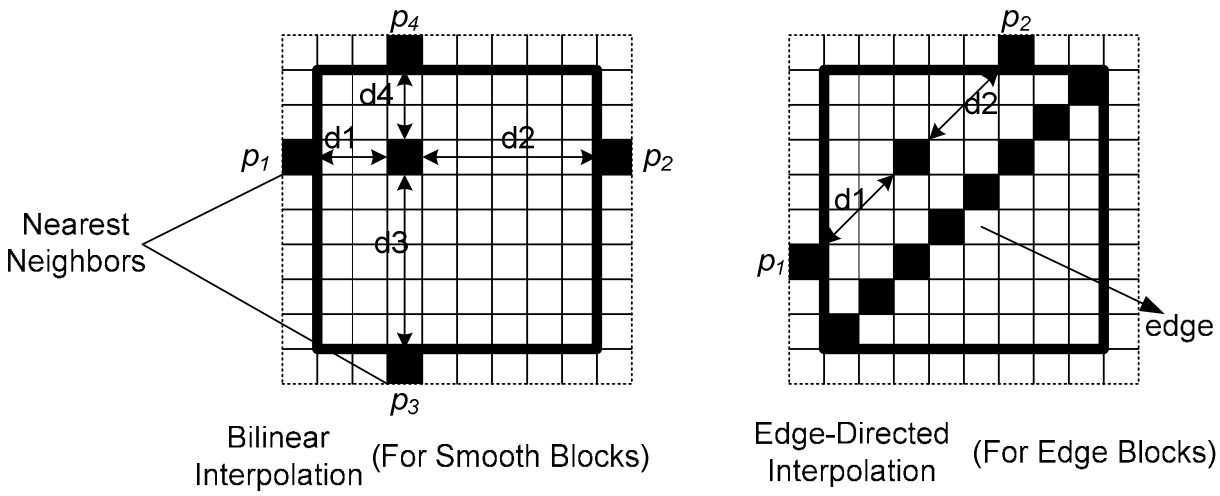

Figure 4.10 Linear Interpolation for smooth blocks and for edge blocks

Let *p* be the lost pixel in an 8x8 block and *p1, p2, p3, p4* be the nearest neighbors from all the directions, *d1, d2, d3, d4* being the respective distances

from the neighbor pixel to the lost pixel p. In the bilinear interpolation technique the lost pixel p is replaced with the weighted average of all the four nearest pixels as follows

$$p = \frac{(d1 * p1) + (d2 * p2) + (d3 * p3) + (d4 * p4)}{d1 + d2 + d3 + d4}$$

For an edge directed interpolation only the nearest pixels that are in the direction of the edge are considered for the interpolation.

$$p = \frac{(d2 * p1) + (d1 * p2)}{d1 + d2}$$

For each given edge pattern, we have a corresponding set of neighboring pixels and their respective weights based on the distance from pixel under consideration.

4.4. Results

The quality performance of the scheme is tested by calculating the PSNR before interpolation and after interpolation. In Table 4.1 PSNR 1 represent the PSNR after the interpolation and PSNR 2 represent PSNR before the interpolation when no protection is provided.

Test Images	PSNR 1	PSNR 2
Lena	39.5749	32.9911
Baboon	35.7223	33.7277
Cameraman	33.2671	31.0761
Airplane	33.5299	30.8173

Table 4.1 PSNR values before and after the edge-directed Interpolation

We captured some erroneous 8x8 blocks before the interpolation and after the interpolation to show the performance with respect to visual quality. The 8x8 blocks are taken from Lena image.

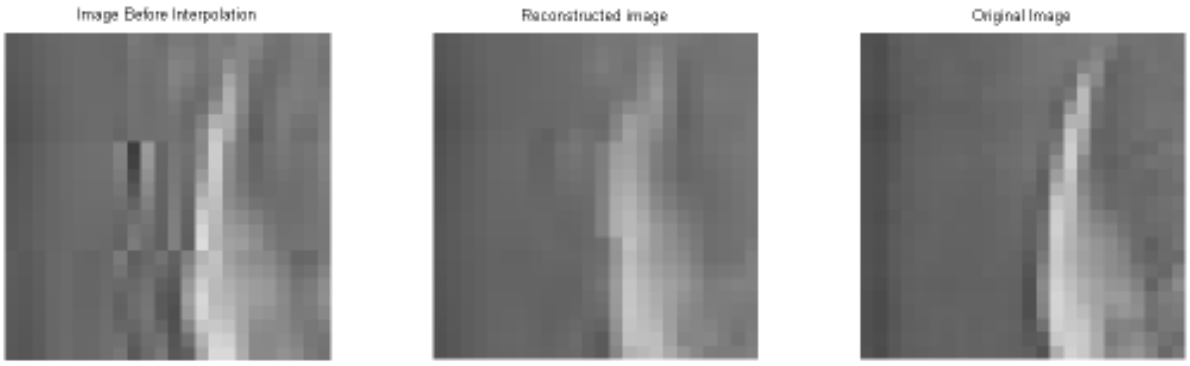

Figure 4.11 Zoomed 8x8 *Block from Lena*

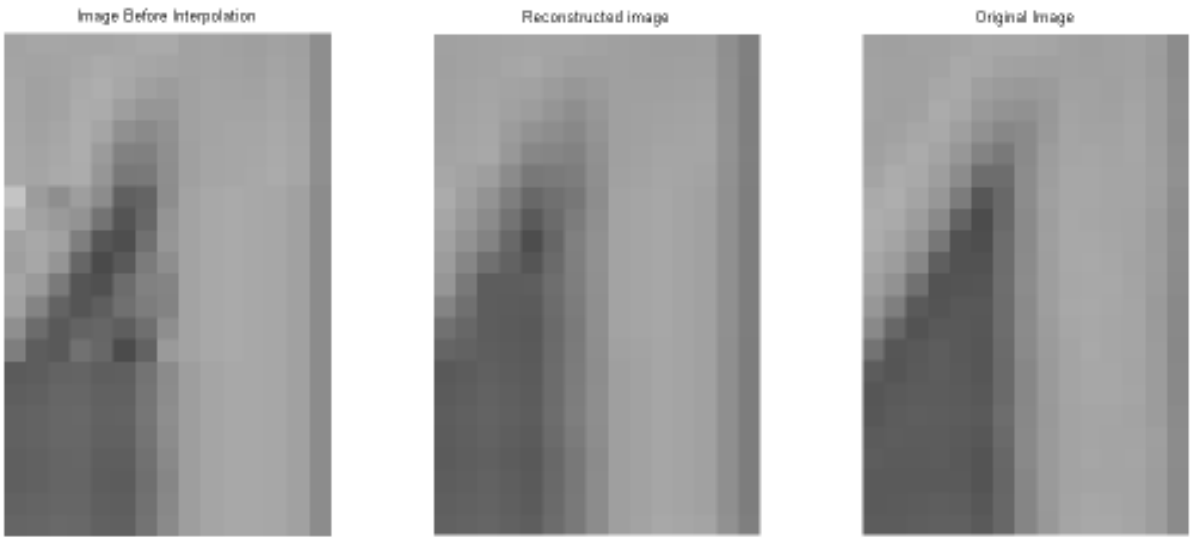

Figure 4.12 Zoomed portion of Lena containing an erroneous 8x8 *block*

Figure 4.13 Zoomed portion of Lena Image containing an erroneous 8x8 *block*

The above figures 4.11-4.13 show that the edges are interpolated along their direction. The above results show that our scheme produces good results making use of the edge-based interpolation.

4.5 Comparative results

For measuring the comparative performance, we take results from algorithms previously implemented and compare them with the results we obtained.

Table 4.2 shows the comparative PSNR results calculated when a Lena Image is transmitted over a noisy channel with error rate 10^{-3}.

ECDH [52]	P.Yin, B. Liu 'Error Concealment using Data Hiding' [11]	Y. Liu, Y.Li 'Error concealment for digital Images using Data Hiding' [50]	Proposed Scheme
34.07 dB	32.32 dB	33.28 dB	33.72 dB

Table 4.2 Comparative PSNR results at error rate 10^{-3}

Chapter 5

Conclusions

An image error concealment method, which achieves error detection and image reconstruction using data hiding, is proposed. A novel data scrambling scheme is proposed in order to achieve error detection and also to protect the hidden data. We have also studied the synchronization performance of two entropy coding schemes, namely T-codes and Huffman codes.

Simulations show that T-codes give better synchronization performance when compared to Huffman codes, with some sacrifice in compression efficiency. Motivated by these results, we have chosen to use T-codes as the entropy coding technique in our error concealment method. The reconstructed image is recovered after channel error using edge-based interpolation. The edge directional information is transmitted to the receiver by hiding it in the DCT coefficients. A novel scrambling algorithm is proposed to detect the errors in the received bit stream and also to protect the hidden data. Simulation results on images transmitted over a binary symmetric channel (BSC) using T-codes and edge embedding algorithm, demonstrate the performance of the proposed scheme.

The proposed algorithm uses a simple augmentation of T-codes, where the T-codes tree expansion is limited. To increase the number of T-codes generated, a higher augmentation level can be chosen. To improve the compression performance of the scheme, best T-codes can be chosen from the huge code set generated from the higher augmentation of T-codes. We developed a novel algorithm for the selection of best T-codes. Our selection algorithm achieved good compression results that are comparable to that of Huffman codes.

In its current form, the protection of DC coefficients is one limitation acknowledged in the proposed work. The proposed scheme can be made more efficient by adding some kind of FEC to the DC coefficients of the image blocks. The results obtained are encouraging for further study on the edge embedding algorithm using data hiding for video signals transmitted over real channels.

References

[1] Digital Compression and Coding of Continuous-Tone Still Images, CCITT Commendation T.81, 1992.

[2] Diego Santa-Cruz, Raphaël Grosbois and Touradj Ebrahimi. JPEG performance and assessment Signal Processing: Image Communication, pages 113-130, volume 17, no. 1, Jan. 2002.

[3] D.W. Redmill and N.G. Kingsbury, "The EREC: an error-resilient technique for coding variable-length blocks of data" IEEE Trans. Image Processing, Vol. 5, pp. 565-574, Apr. 1996.

[4] M. R. Titchener, "The synchronization of variable-length codes," IEEE Trans. Inform. Theory, vol. 43, pp. 683–691, Mar. 1997.

[5] W. Zeng, and B. Liu, .Geometric-structure-based error concealment with novel applications in block-based low bit rate coding,. IEEE Transactions on Circuits and Systems for Video Technology, vol. 9, no. 4, pp. 648-665, June 1999.

[6] M. R. Titchener and J. J. Hunter, "Synchronization process for the variable-length T-codes," in Proc. Inst. Elec. Eng. pt. E, Computers and Digital Techniques, vol. 133, no. 1, pp. 54–64, 1985.

[7] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," Proc. of the IEEE, vol. 86, no. 5, pp. 974-997, May 1998.

[8] Min Wu and Bede Liu, "Watermarking for image authentication," Proceedings of the 1998 IEEE International Conference on Image Processing (ICIP '98), vol. 2, pp. 437-441, 4-7 October 1998.

[9] M. Ramkumar, A. N. Akansu, and A. A. Alatan, "On the choice of transforms for data hiding in compressed video," Proc. of IEEE ICASSP '99, pp. 3049-3052, 1999.

[10] A. Yilmaz and A. A. Alatan, "Error concealment of video sequences by data hiding," in Proc. Int. Conf. Image Processing, vol. 3, Sept. 2003, pp. 679–682.

[11] P. Yin, B. Liu, and H. H. Yu, "Error concealment using data hiding," Proc. of. ICASSP, vol. 3, pp. 1453-1456, May 2001.

[12] S. Yafei, Z. Li, W. Guowei, and L. Xinggang, "Reconstruction of missing blocks in image transmission by using self-embedding," Proc. of Int. Symp. on Intel. Multim., Video, and Speech Proc., p. 535-538, May 2001.

[13] A. Piva, R. Caldelli, V. Cappellini, A. De Rosa, "Data hiding for transmission error detection in H.263 video," Tyrrhenian International Workshop on Digital Communications (IWDC 2002), Capri, Italy, 8-11 September, 2002.

[14] Y. Wang, S. Wenger, J. Wen, and A. K. Katsaggelos, "Error resilient video coding techniques," IEEE Signal Processing Magazine, vol. 17, no. 4, July 2000, pp. 61-82.

[15] V.K Goyal, .Compression Meets the Network., IEEE Signal Processing Magazine., September 2001.

[16] V Vaishampayan, .Design of Multiple Description Scalar Quantizers., IEEE Trans. On Inform. Theory, Vol. 39, pp. 821-834, May 1993.

[17] G. Davis and J. Danskin. Joint source and channel coding for image transmission over lossy packet networks. In Conf. Wavelet Applications to Digital Image Processing, Denver, CO, August 1996. SPIE.

[18] M. Pereira, M. Antonini, M. Barlaud, .Multiple descriptions image and video coding for wireless channels,. EURASIP Journal on Applied Signal Processing vol. 18, pp. 925.945, 2003.

[19] J. C. Maxted and J. P. Robinson, "Error recovery for variable length codes," IEEE Trans. Inform. Theory, vol. IT-31, pp. 794–801, 1985.

[20] T. J. Ferguson and J. H. Ranowitz, "Self-synchronizing Huffman codes," IEEE Trans. Inform. Theory, vol. IT-30, pp. 687–693, July 1984.

[21] P. G. Neumann, "Self-synchronizing sequential coding with low redundancy," Bell Syst. Tech. J., vol. 50, pp. 951–981, Mar. 1971.

[22] S.-M. Lei and M.-T Sun, "An entropy coding system for digital HDTV applications," IEEE Trans. Circuits Syst. Video Technol., vol. 1, pp. 147–154, Mar. 1991.

[23] W.-M. Lam and A. R. Reibman, "Self-synchronizing variable length codec for image transmission," in Proc. ICASSP'92, San Francisco, CA, Mar. 1992, pp. III477–480.

[24] B. L. Montgomery and J. Abrahams, " Synchronization of binary source codes," IEEE Trans. Inform. Theory, vol. IT-32(6), pp. 849-854, Nov. 1986.

[25] N. Zierler, "Linear recurring sequences," .I. Sot. Ind. Appl. Math., vol. I, pp. 31-48, Mar. 1959.

[26] E. N. Gilbert and E. F. Moore, "Variable length binary encodings," Be// Syst. Tech. J., vol. 38, p. 933, 1959.

[27] P. G. Neumann, "Efficient error-limiting variable length codes," IRE Trans. Inform. Theory, vol. IT-8, pp. 292-304, July 1962.

[28] M. R. Titchener, "Digital encoding by means of new T-codes to provide improved data synchronization and message integrity," Technical Note, IEE Proceedings, Volume: 131, Pt. E, Number: 4 , July 1984 Page(s): 51 - 53.

[29] S. Aign, ``Error concealment for MPEG-2 video," in Signal Recovery Techniques for Image and Video Compression and Transmission, A. K. Katsaggelos and N. P. Galatsanos, editors, ch. 8, pp. 235-268, Kluwer Academic Publishers, 1998.

[30] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," IEEE Trans. Image Proc., vol. 4, no. 4, pp. 470-477, Apr. 1995.

[31] G.-S. Yu, M. M.-K. Liu, and M. W. Marcellin, "POCS-based error concealment for packet video using multiframe overlap information," IEEE Trans. CAS for Video Tech., vol. 8, no. 4, pp. 422-434, Aug. 1998.

[32] M.C. Hong, L. Kondi, H. Scwab, and A. K. Katsaggelos, "Video Error Concealment Techniques," Signal Processing: Image Communications, special issue on Error Resilient Video, 14(6-8), pp. 437-492, 1999.

[33] Alper Koz and A. Aydın Alatan, "Foveated image watermarking," Proceedings of the 2002 IEEE International Conference on Image Processing, (ICIP '02), vol. 3, pp. 657-660, 24-28 June 2002.

[34] David L. Robie and Russell M. Mersereau, "The use of Hough transforms in spatial error concealment," Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00), vol. 4, pp. 2131-2134, 5-9 June 2000.

[35] R. Talluri, "Error resilient video coding in the ISO MPEG-4 standard," IEEE Communications Magazine, vol. 36, no. 6, pp. 112-119, June 1998.

[36] J. Liang and R. Talluri, "Tools for robust image and video coding in JPEG2000 and MPEG4 standards," Proceedings of the SPIE Conference on Visual Communications and Image Processing, vol. 3653, January 23-29 1999, San Jose, California, pp. 40-51.

[37] Stefan Katzenbeisser and Fabien A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Inc., USA, 2000.

[38] Ingemar J. Cox, Matthew L. Miller, and Jeffrey A. Bloom, Digital Watermarking, Academic Press, USA, 2002.

[39] Ingemar J. Cox, Joe Kilian, Tom Leighton, and Talal Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Transactions on Image Processing, vol. 6, no. 12, pp. 1673-1687, December 1997.

[40] David L. Robie and Russell M. Mersereau, "Video error correction using steganography" Proceedings of the 2001 IEEE International Conference on Image Processing, vol.1, pp.930-933, 7-10 October 2001.

[41] Jie Song and K. J. R. Liu, "A data embedding scheme for H.263 compatible video coding," Proceedings of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS '99), vol. 4, pp. 390-393, 30 May-2 June 1999.

[42] Peng Yin, Min Wu, and Bede Liu, "Robust error resilient approach for MPEG video transmission over internet," Visual Communication and Image Processing, SPIE, vol. 4671, pp. 103-111, January 2002.

[43] Teng Sing Wang, Pao-Chi Chang, Chih-Wei Tang, Hsueh-Ming Hang, and Tihao Chiang, "An error detection scheme using data embedding for H.263 compatible video coding," Coding of Moving Pictures and Associated Audio, ISO/IEC JTC1/SC29/WG11, MPEG99/N6340, July 2000.

[44] B. Chen and G. W. Wornell. Dither modulation: A new approach to digital watermarking and information embedding. In Proceedings of SPIE: Security and Watermarking of Multimedia Contents, January 1999.

[45] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. IEEE Trans. on Image Processing, 6(12): 1673-1687, December 1997.

[46] B. Chen and G. W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. on Information Theory, 47(4):1423-1443, May 2001.

[47] J. Chou, L. El Ghaoui, S. S. Pradhan, and K. Ramchandran. On the duality between distributed source coding and data hiding. In Proceedings of 33rd Asilomar Conference on Signals, Systems and Computers, November 1999.

[48] J. Chou, S. S. Pradhan, and K. Ramchandran. A robust optimization solution to the data hiding problem using distributed source coding principles. In Proceedings of CISS, March 2000.

[49] Mahalingam Ramkumar, Ali N. Akansu, and A. Aydın Alatan, "On the choice of transforms for data hiding in compressed video," Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99), vol. 6, pp. 3049-3052, 15-19 March1999.

[50] Y. Liu and Y. Li, "Error concealment for digital images using data hiding," presented at the 9th DSP Workshop, Hunt, TX, Oct. 2000.

[51] HIGGIE, G.R., 'Database of best T-codes', *IEE Proc. E. Comput. Digit. Tech.,* 1996, 143**,** pp. 213-218.

[52] Masayuki Kurosaki, Khairul Munadi, Hitoshi Kiya, 'Error correction using data hiding technique for JPEG2000 images'. ICIP (3) 2003: 473-476