République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Ferhat Abbas Sétif

THESE
Présentée à la Faculté des Sciences de l'Ingénieur
Département d'Electronique
Pour l'obtention du Diplôme de

Doctorat d'ETAT

En Electronique
par
KHELIL KHALED

THEME

CONCEPTION CONJOINTE DE CODAGE SOURCE-CANAL

Soutenue publiquement le 12/12/2007 devant le jury composé par :

Prof.	Abdelhafid Khellaf	U. F.A. Sétif	Président
Prof.	Raïs El'hadi Bekka	U. F.A. Sétif	Rapporteur
Prof.	Noureddine Doghmane	U. Annaba	Examinateur
Prof.	Djamel Chikouche	U. M'sila	Examinateur
Prof.	Nabil Khenfer	U. F.A. Sétif	Examinateur

2007

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ferhat Abbès University of Sétif

A Thesis submitted to the Graduate School
in partial fulfillment of the requirements
for the Degree

Docteur d'ETAT

JOINT SOURCE-CHANNEL CODING: APPLICATION TO MULTIPLE DESCRIPTION IMAGE CODING

By

KHELIL KHALED

Committee in charge:

Chairman:	A. Khellaf	Prof.	U. Sétif
Advisor:	R. E. Bekka	Prof.	U. Sétif
Members:	N. Doghmane	Prof.	U. Annaba
	D. Chikouche	Prof.	U. M'sila
	N. Khenfer	Prof.	U. Sétif

2007

DEDICATION

ACKNOWLEDGMENTS

I wish to express my sincere thanks and gratitude to my supervisor Professor R. E. BEKKA for providing guidance for this thesis work. His insight and motivation were invaluable.

My gratitude is extended to Prof. A. KHELLAF, Prof. N. DOGHMANE, Prof. D. CHIKOUCH and Prof. N. KHENFER for serving as members on the thesis committee and for their comments.

ABSTRACT

Source coding is expected to remove the redundancy from a signal in order to make the transmission more efficient. On the other hand, channel coding adds back some redundancy to the signal, in order to make it possible to correct the errors occurred during the transmission. As these two coding schemes act contrarily, the idea of doing them jointly gets meaning. The considered problem in this thesis is the design of a joint source channel (JSC) image coder within the context of multiple description coding (MDC).

Transmitting images via a variety of communication networks is both common and important to today's increasingly technology driven society. As such, there is growing demand for high speed, error-free transmission of these images. Unfortunately, with increasingly congested networks and long distance links, this is not always possible. Ideally, image data would be split into packets and sent to a destination with all packets arriving intact at the receiver's end. However, a packet is invariably dropped or scrambled due to congestion or error on its path through the network, necessitating a resend of the lost packet. This puts an even larger load on an already congested network, as the receiver must first send a message back to the sender indicating packet loss, followed by a retransmission by the sender. As such, it could take an unacceptable length of time before all image packets are received intact. The problem is magnified when the communications link is very long (e.g. from Earth to a satellite orbiting Mars) or the backwards link from receiver to sender is non-existent.

The proposed solution is to encode image data with multiple description coding which is a technique that can be considered a JSC coding for erasure channels. It is recognized as an effective method to protect multimedia information transmitted over networks subject to erasures. In the MDC approach, the source data is encoded into two or more correlated bitstreams called descriptions. The correlation/redundancy introduced between descriptions helps to minimize the distortion of the recovery image when only a subset of descriptions is received. Here, the case of four descriptions is considered. Using all descriptions, high quality reconstruction level is obtained. In case of channel failure, lower but still acceptable quality reconstruction can be obtained from any subset of received descriptions.

The purpose of this thesis is twice. First, a current Multiple Description Transform Coding (MDTC) is deeply studied and a different way in forming the different descriptions is proposed. It is shown that the employed technique leads to an improvement in performance in terms of rate/distortion.

Second, is to propose a new and simple MDTC scheme based on the Discrete Wavelet Transform (DWT) instead of the Discrete Cosine Transform (DCT) as it is the case in other prevalent works. It is shown that the new proposed MDTC coding scheme, when applied to image coding, outperforms the existing MDTC image coder in terms of the distortions and quality of the reconstructed images and complexity of implementation.

RESUME

Le codage de source est sensé réduire la redondance d'un signal, afin de diminuer la quantité de données. D'autres part, le codage de canal rajoute de la redondance au signal pour le rendre plus robuste vis-à-vis des erreurs de transmission. Comme ces deux schémas de codage agissent de manière contraire sur la redondance, il serait probablement bénéfique de faire les deux actions conjointement. Dans cette thèse, la conception conjointe source-canal(Joint Source Channel Coding :JSCC) d'un codeur d'images est considérée dans le contexte du codage par descriptions multiples (Multiple Description Coding :MDC).

La transmission des images sur des réseaux de transmission est à la fois commune et importante pour une société de plus en plus menée par la technologie. Ainsi, il y a demande croissante en transmission à grande vitesse et sans erreur de ces images. Malheureusement, avec des réseaux de plus en plus encombrés et des liaisons de grande distance, ce n'est pas toujours possible. Dans le meilleur des cas, des données image seraient partagées en paquets et envoyées à une destination avec tous les paquets arrivant intact au récepteur. Cependant, un paquet est invariablement perdu pour une raison de congestion du réseau ou d'erreur sur son chemin à travers le réseau, rendant nécessaire un renvoi du paquet perdu. Ceci met une charge encore plus grande sur un réseau déjà encombré, car le récepteur doit d'abord envoyer un message de nouveau à l'émetteur indiquant la perte de paquet, suivi d'une retransmission par l'émetteur. En tant que tels, il pourrait prendre une durée inacceptable avant que tous les paquets d'image soient reçus intacts. Le problème devient pus sévère pour des communications très distantes (par exemple de la terre à un satellite) ou lorsque la boucle de rétroaction du récepteur à l'émetteur est inexistante.

La solution proposée est de coder les données image avec le codage par descriptions multiples (MDC) qui est considéré comme un codage conjoint source-canal pour des canaux à effacements. Le codage par descriptions multiples consiste à créer deux ou plusieurs représentations distinctes mais corrélées, d'une même source, qui ont la propriété de se raffiner mutuellement. La corrélation/redondance introduite entre les descriptions aide à améliorer la qualité de reconstruction de l'image lorsque seulement certaines descriptions (et non pas toutes) sont reçues. Dans cette thèse, le cas de quatre descriptions est considéré. La qualité optimale est obtenue, quand toutes les quatre descriptions sont reçues. Sinon, une qualité de reconstruction inférieure mais acceptable peut être obtenue à partir de n'importe quel sous-ensemble de descriptions reçues.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ARQ	Automatic Repeat reQuest
bpp	bits per pixel
CWT	Continuous Wavelet Transform
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DWT	Discrete Wavelet Transform
ECUQ	Entropy Coded Unbounded Uniform Quantization
FEC	Forward Error Correction
FFT	Fast Fourier Transform
i.i.d.	Independent Identically Distributed
IA	Index Assignment
JPEG	Joint Photographic Experts Group
JSCC	Joint Source Channel Coding
KLT	Karhunen-Loeve transform
LOT	Lapped Orthogonal Transform
MD	Multiple Description
MDC	Multiple Description Coding
MDSQ	Multiple Description Scalar Quantization
MDTC	Multiple Description Transform Coding
MRA	MultiResolution Analysis
MSE	Mean Square Error
PCM	Pulse Code Modulation
pdf	Probability Density Function
PSNR	Peak Signal to Noise Ratio
QMF	Quadrature Mirror Filter
SNR	Signal to Noise Ratio
STFT	Short Time Fourier Transform
TCP	Transmission Control Protocol
VLC	Variable Length Code
VQ	Vector Quantization

CHAPTER 1

INTRODUCTION

1.1 Problem statement

With the rapid development of Internet applications, transmitting images has become both common and important. As such, there is growing demand for high speed, error-free transmission of these images. Unfortunately, with increasingly congested networks and due to the heterogeneous aspect of such networks, this is not always possible. A typical scenario might require data to move from a fiber link to a wireless link, which necessitates dropping packets to accommodate the lower capacity of the latter. Also, packets are invariably dropped or scrambled due to congestion or interference on their path through the network. The most common way to communicate an image over the internet is to use a progressive encoding system and to transmit the coded image as a sequence of packets with a protocol known as TCP [1] (Transmission Control Protocol). When there are no packet losses, the receiver can reconstruct the image as the packets arrive; but if a packet is lost, the whole reconstruction stops until that packet is resent and finally received. For a TCP-based network, the delay for resending a packet is often much longer than the time between the arrivals of packets. If one packet is lost, all packets after it become useless for the reconstruction of the image until the retransmitted lost packet is received. Therefore, the use of unreliable channels, such as wireless networks or the current Internet, implies that error-free delivery of data packets can only be achieved by allowing retransmission of lost or damaged packets, through error control mechanisms such as Automatic Repeat reQuest (ARQ) [2].

If packet loss is unavoidable and their retransmission is not an option (e.g., due to a delay constraint or a lack of feedback channel), one has to devise a way of making the received data meaningful despite the loss. This problem finds its natural solution in the so-called multiple description framework [1]. This thesis addresses the problem of transmitting data images over erasure channels within the context of Multiple Description Coding (MDC).

Generally, the goal of error control is to make the data bitstream resilient to transmission errors. The channel noise can occur in the form of random bit errors, burst bit errors or packet losses. To make such bitstream resilient to transmission errors one must add redundancy to the stream, so that it is possible to detect and correct errors. Typically, this is done at the channel by using Forward Error Correction (FEC) [3, 4, 5, 6, 7, 8]. The classical Shannon information theory [9] states that one can separately design the source and channel coders, to achieve error-free delivery of a compressed bitstream, so long as the source is represented by a rate below the channel capacity. Therefore, the source coder should compress a source as much as possible (below the channel capacity) for a specified distortion, and then the channel coder can add redundancy through FEC to the compressed stream to enable the correction of transmission errors. The standard FEC presents high performance when developed for constant channels and channels that do not present burst errors. Unfortunately, this is not the case in many situations where channels have a highly variable quality.

In fact, the typical error control techniques, ARQ, FEC or even both together [10], cannot be easily adapted to real time transmissions. Therefore, Joint Source-Channel Coding (JSCC) is often preferred. Such kind of scheme allocates a total amount of redundancy between source and channel coding to recover from erroneous or missing bits.

1.2 Joint source-channel coding

Source coding and channel coding are essential functions in any communication system. The source coding block is designed to remove as much redundancy as possible from the source while the channel coding block adds control redundancy to the compressed source. For practical and existing systems, these two blocks are separately optimized. This was motivated both by Shannon “separation theorem” [9] and by the conceptual simplicity of considering only one or the other. However it is well known that the Shannon theorem requires codes of infinite lengths (and hence infinite complexity and delay) for both source coder and channel coder.

The limitations of separate source and channel coding have lead to the problem of designing joint source-channel coding (JSC) coders. JSC coding can lead to performance gains under complexity and/or delay constraints and offer robustness against channel variation.

1.3 Multiple description coding

Multiple description coding is a technique which can be considered a JSC code for erasure channels. It is recognized [11] as an effective method to protect multimedia information transmitted over networks subject to erasures. In the MDC approach, two or more correlated descriptions of the same data are generated which can be independently decoded, and yield mutually refinable information. Therefore, the quality of the recovered signal is dependent only on the number of received descriptions, and not on the specific loss pattern.

In MD coding, the signal is encoded into more than one description and each description is packed into one packet. Each packet can have equal importance. When all packets are received, a signal with high fidelity is reconstructed. When one or more packets are lost, the quality of the reconstructed signal will degrade gracefully.

Many MD coding methods have been developed. The most popular ones are: MD quantization, MD correlation transform, and MD coding with frames. There is no obvious best choice from these methods. The complexity of these methods is often high and considerable research has been undertaken to improve these methods and to apply them to signal coding.

The purpose of this thesis is twice. First, a current Multiple Description Transform Coding (MDTC) [12, 13] is deeply studied and a different way in forming the different descriptions is proposed. It is shown that the employed technique leads to an improvement in performance in terms of rate/distortion.

Second, is to propose a new and simple MDTC scheme based on the Discrete Wavelet Transform (DWT) instead of the Discrete Cosine Transform (DCT) as it is the case in the work of Goyal et al. [12, 13]. It will be shown that the new proposed MDTC coding scheme, when applied to image coding, outperforms the existing MDTC image coder in terms of the distortions and quality of the reconstructed images and complexity of implementation.

1.4 Thesis overview

The remainder of this thesis is organized as follows. Chapter 2 describes a digital communication system and provides necessary background on image coding, information theory, quantization and source coding. Chapter 3 is dedicated to the presentation of the principle image coding transforms and their properties. In chapter 4, we address the problem of multiple description transform coding and its application to image coding. Based on the work of [12, 13], we propose a new technique in forming the descriptions, and then a new wavelet based MDTC scheme is developed. This chapter also presents the experimental results which include the objective and subjective qualities of reconstructed images using the two proposed coders. Finally chapter 5 summarizes the results of the study and provides suggestions for future work.

CHAPTER 2

BACKGROUND

In this chapter we provide some of the necessary background aimed at keeping this thesis self-contained.

2.1 Digital communication systems

Digital communication consists of the transmission of information from a source that generates the information to one or more destinations through a communication channel. In the following, we describe the basic elements of a generic digital communication system, which are illustrated in Figure 2.1.

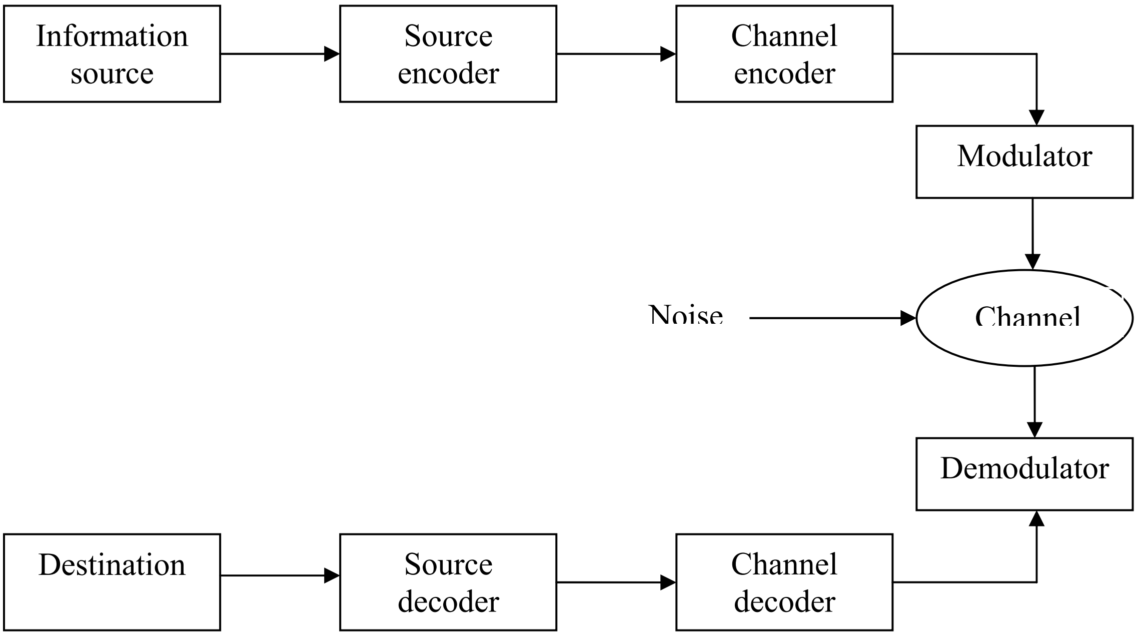


Figure 2.1: Block diagram of a generic digital communication system.

- Source encoder: The source encoder efficiently converts the output of the information source into a stream of binary digits called bitstream. In this context, efficiently means that the size of the bitstream that represents the source output is minimized. This is why the term compression is usually used as a synonym for source coding. There are two general classes of compression: lossless compression and lossy compression. We talk about lossless compression if the original signal can be perfectly recovered from the information bitstream that represents the compressed data. Otherwise, the compression is said to be lossy

- Channel encoder: The purpose of a channel encoder is to protect the information bitstream against errors introduced by a noisy communication channel. This can be achieved by inserting, in an intelligent manner, additional bits, called *redundancy*, into the information bitstream. The redundancy can be used at the receiver to detect and possibly correct errors. The bitstream at the output of a channel encoder is organized in binary strings called *code words.*

- Digital modulator: The digital modulator serves as the interface of the communication channel. It converts the code words outputted by the channel encoder into a waveform, which is the appropriate form for transmission through nearly all practical communication channels.

- Digital demodulator: The digital demodulator reconverts the waveform upon arrival from the communication channel into a digital signal that can be processed by the channel decoder.

- Channel decoder: The purpose of a channel decoder is to reconstruct the original information bitstream using the protection bits inserted by the channel encoder. Both the channel encoder strategy and the channel noise characteristics are taken into consideration in the design of the channel decoder.

- Source decoder: The aim of a source decoder is to reconstruct the signal originally sent from the information source and to deliver it to its destination.

It is worthwhile to indicate that error control in the above described system, which is a one-way system, is achieved using *forward error correction* (FEC). However, if a feedback channel is available, the data lost during the transmission may be retransmitted using *Automatic repeat request* (ARQ) schemes.

We note also that the system can be further simplified by combining the modulator, the physical channel, and the demodulator in one box called *discrete channel* or *coding channel* [14]. In fact, these three elements are usually jointly modeled using a unique mathematical model.

A key issue in evaluating the performance of a digital communication system concerns the efficiency with which information from a given source can be represented. Another key issue pertains to the rate at which information can be transmitted reliably over a noisy channel. The fundamental limits on these key aspects of system performance have their roots in information theory, which was originally known as the mathematical theory of communication. Information theory deals only with mathematical modeling and analysis of a communication system rather than with physical sources and physical channels.

2.2 Information theory

This section involves an overview of information theory and its importance in data transmission. Using these theories, a good understanding of what conditions are required for data representation can be gained and the theoretical limits of signal representation fully comprehended. Specifically, given an information source and a noisy channel, information theory provides limits on the minimum number of bits per symbol required to fully represent the source, and the maximum rate at which reliable communication can take place over the channel [15]. Information theory also serves in giving a quantitative measure of entropy (i.e. information content) of a digital signal. The Father of information theory was Claude E. Shannon, who during the late 1940s and the 1950s developed a theory of source coding in order to quantify the optimal achievable performance in data representation systems.

2.2.1 Uncertainty, Information, and Entropy

Suppose that a probabilistic experiment involves the observation of the output emitted by a discrete source. The source output is modeled as a discrete random variable, X , which takes on symbols from a fixed finite alphabet

$$\varphi = \{x_0, x_1, \ldots, x_{K-1}\} \tag{2.1}$$

With probabilities

$$P(X = x_k) = p_k \qquad k = 0,1,\ldots,K-1 \tag{2.2}$$

where the set of probabilities must satisfy the condition

$$\sum_{k=0}^{K-1} p_k = 1 \tag{2.3}$$

The symbols emitted by the source are assumed to be statistically independent. A source having the properties just described is called a discrete memoryless source, memoryless in the sense that the symbol emitted at any time is independent of previous symbols.

How much information is produced by such a source? The idea of information is closely related to that of “uncertainty” and “surprise”.

Consider the event $X = x_k$, describing the emission of symbol x_k as defined in Equation 2.2. Clearly, if the probability $p_k = 1$ and $p_i = 0$ for all $i \neq k$, then there is no surprise and therefore no information when symbol x_k is emitted. If, in the other hand, the probability p_k is low, then there is more surprise and therefore more information when symbol x_k is emitted by the source than when symbol $x_i, i \neq k$, with higher probability is emitted.

The amount of information gained after observing the event $X = x_k$ which occurs with probability p_k is given by [14]

$$I(x_k) = \log_2\left(\frac{1}{p_k}\right) \tag{2.4}$$

The unit of information called the bit (a contraction of binary unit). When $p_k = \frac{1}{2}$, we have $I(x_k) = 1$ bit. Hence, one bit is the amount of information that we gain when one of two possible and equiprobable events occurs. Indeed, $I(x_k)$ is a discrete random variable that takes on the values $I(x_0), I(x_1), ..., I(x_{K-1})$ with probabilities $p_0, p_1, ..., p_{K-1}$, respectively. The mean value of $I(x_k)$ over the source alphabet φ is given by

$$\begin{aligned} H(\varphi) &= E[I(x_k)] \\ &= \sum_{k=0}^{K-1} p_k I(x_k) \\ &= \sum_{k=0}^{K-1} p_k \log_2(\frac{1}{p_k}) = -\sum_{k=0}^{K-1} p_k \log_2 p_k \end{aligned} \tag{2.5}$$

The important quantity, $H(\varphi)$, is called the ***entropy*** of a discrete memoryless source with source alphabet φ. It is a measure of the average information content per source symbol.

Note that the entropy $H(\varphi)$ depends only on the probabilities of the symbols in the alphabet φ of the source.

2.2.2 Source coding

An important problem in communication is the efficient representation of data generated by a discrete source. The source by which this representation is accomplished is called source encoding. For the source coding to be efficient, we require knowledge of statistics the source. In particular, if some source symbols are known to be more probable than others, then this feature could be exploited in the generation of a source code by assigning short code words to frequent source symbols, and long code words to rare source symbols. Such source code is referred to as a *variable-length code*. Two functional requirements should be satisfied by this source code:

- The code words produced by the encoder are in binary form.
- The source code is uniquely decodable, so that the original source sequence can be reconstructed perfectly from the encoded binary sequence.

Consider then the arrangement shown in Figure 2.2 that depicts a discrete memoryless source whose output x_k is converted by the source encoder into a stream of 0s and 1s, denoted by b_k. The source has an alphabet of K different symbols, and the kth symbol x_k occurs with probability p_k.

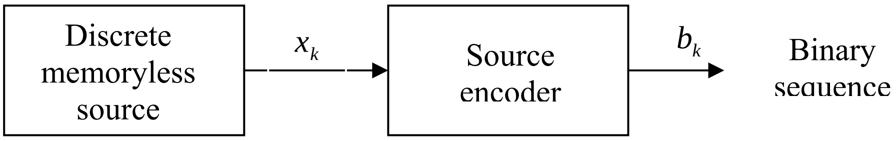


Figure 2.2: Source encoding.

Let the binary code word assigned to symbol x_k by the encoder have length l_k, measured in bits. The average code word length, $\overline{L}$, is defined as

$$\overline{L} = \sum_{k=0}^{K-1} p_k l_k \tag{2.6}$$

The parameter $\overline{L}$ represents the average number of bits per source symbol used in the source encoding process. Let $L_{\min}$ denote the minimum possible value of $\overline{L}$. The coding efficiency of the source encoder is defined as

$$\eta = \frac{L_{\min}}{\overline{L}} \tag{2.7}$$

With $\overline{L} \geq L_{\min}$, the minimum value $L_{\min}$ is defined in Shannon's first theorem: the source-coding theorem [14] stated as follows:

Given a discrete memoryless source of entropy $H(\varphi)$, the average code word length $\overline{L}$ for any source encoding is bounded as

$$\overline{L} \geq H(\varphi) \tag{2.8}$$

Accordingly, the entropy $H(\varphi)$ represents a fundamental limit on the average number of bits per source symbol, $\overline{L}$, necessary to represent a discrete memoryless in that $\overline{L}$ can be made as small as, but no smaller than, the entropy $H(\varphi)$.

2.2.3 Entropy coding

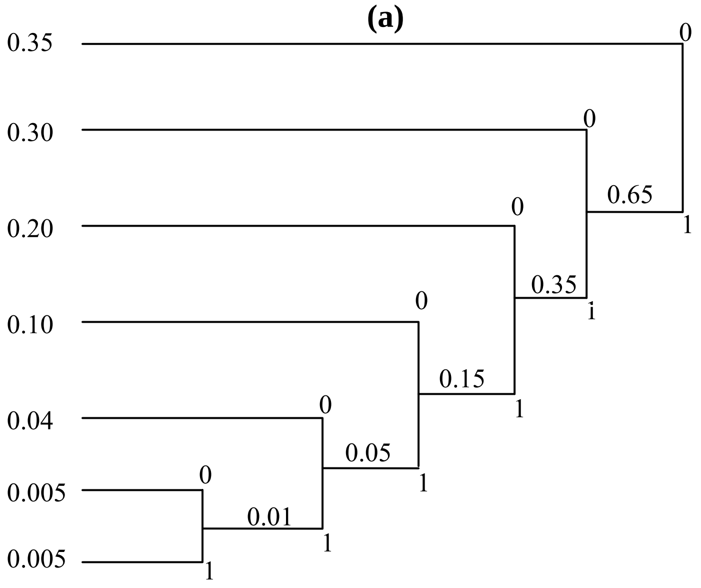
A variety of coding schemes have been developed over the years, these coding schemes play a very important part in the compression process as they relate the theoretical entropy to the practical entropy achieved after compression of the data. Two major entropy coding systems will be presented from a practical perspective namely Huffman and Arithmetic coding.

2.2.3.1 Huffman coding

Huffman coding was developed by D. A. Huffman [16] in 1952. This particular method of entropy coding yielded a performance quite close to the fundamental limit set by the entropy of a discrete memoryless source, namely $H(\varphi)$. The Huffman code is optimum in the sense that no other uniquely decidable set of code words has a smaller average code word length for a given discrete memoryless source [17]. We illustrate this encoding algorithm through an example.

Consider a discrete memoryless source with seven possible symbols $x_0, x_1, ..., x_6$ having the probabilities of occurrence illustrated in Figure 2.3. The source symbols are ordered in decreasing order of probabilities, i.e., $P(x_0) > P(x_1) > ... > P(x_6)$. We begin the encoding process with the two least probable symbols x_5 and x_6. These two symbols are tied together with the upper branch assigned a 0 and the lower branch assigned a 1. The two symbols are regarded as being combined into a new symbol with probability equal to the sum of the two original probabilities. Therefore, the list of source symbols is then reduced in size by one. The probability of the new symbol is placed in the list in accordance with its value. The procedure is repeated until we are left with a final list of source symbols with only two for which a 0 and a 1 are assigned as shown in Figure 2.3(a). The code words of the Huffman code for the source are tabulated in Figure 2.3(b).

(a)



(b)

Symbol	Probability	Code word
x_0	0.35	00
x_1	0.30	01
x_2	0.20	10
x_3	0.10	110
x_4	0.04	1110
x_5	0.005	11110
x_6	0.005	11111

Figure 2.3: Example of the Huffman encoding algorithm

The average code word length is therefore

$\overline{L} = 0.35(2) + 0.30(2) + 0.20(2) + 0.10(3) + 0.04(4) + 0.005(5) + 0.005(5) = 2.21$

Using Equation 2.5, the entropy of the specified discrete memoryless source is:

$H(\varphi) = 2.11$

2.2.3.2 Arithmetic coding

A major problem with Huffman coding is that the codes it generates are only an integral number of bits long. For example if the probability of a character is 1/3, the optimum number of bits to code the character is 1.6 as per Shannon's entropy equation (Equation 2.4). Huffman coding has to either assign a 1 or 2 bit code, with either choice leading to a longer compressed message than the theoretical optimum. A clever way to get around this problem is to employ Arithmetic coding. Arithmetic coding doesn't generate individual codes for each character but performs arithmetic operations on a block of data, based on the probabilities of the next character. Using this method it is possible to encode characters with a fractional number of bits, thus approaching the theoretical optimum. This comes from Shannon's second idea of information theory which states that data can always be encoded more efficiently if coded in vector (i.e. block) form. Arithmetic coding achieves this amazing feat by basing its coding on the fact that from any arithmetic coding process a single number is produced that is less than 1 and greater than or equal to 0.

$$0 \le n < 1 \tag{2.9}$$

Where n is the output number.

This single number n can be uniquely decoded to create the exact stream of symbols that went into its construction. To construct the output number n, the symbols are assigned a set of probabilities. A low and high value between 0 and 1 is then allocated to these symbols based on their probabilities. For example, if it is to encode the random message "BILL GATES", it would have a probability distribution that looks like this [18]

Character	SPACE	A	B	E	G	I	L	S	T
Probability	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.1

Once the character probabilities are known, the individual symbols need to be assigned a range along a "probability line", which is nominally 0 to 1. It doesn't matter which characters are assigned which segment of the range, as long as it is done in the same manner by both the encoder and the decoder. The nine character symbol set use here would look like this

Character	SPACE	A	B	E	G	I	L	S	T
Probability	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.1
Range	0.0 - 0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.8	0.8-0.9	0.9-1.0

Each character is assigned the portion of the 0-1 range that corresponds to its probability of appearance. Note also that the character "owns" everything up to, but not including the higher number. So the letter 'T' in fact has the range 0.90 - 0.9999....
The most significant portion of an arithmetic coded message belongs to the first symbol to be encoded. When encoding the message "BILL GATES", the first symbol is "B". In order for the first character to be decoded properly, the final coded message has to be a number greater than or equal to 0.20 and less than 0.30. What we do to encode this number is keep track of the range that this number could fall in. So after the first character is encoded, the low end for this range is 0.20 and the high end of the range is 0.30.

After the first character is encoded, we know that our range for our output number is now bounded by the low number and the high number. What happens during the rest of the encoding process is that each new symbol to be encoded will further restrict the possible range of the output number. The next character to be encoded, 'I', owns the range 0.50 through 0.60. If it was the first number in our message, we would set our low and high range values directly to those values. But 'I' is the second character. So what we do instead is say that 'I' owns the range that corresponds to 0.50-0.60 in the new subrange of 0.2 - 0.3. This means that the new encoded number will have to fall somewhere in the 50th to 60th percentile of the currently established range. Applying this logic will further restrict our number to the range 0.25 to 0.26.

The algorithm to accomplish this for a message of any length is shown below:

```
Set low to 0.0
Set high to 1.0
While there are still input symbols do
    get an input symbol
    code_range = high - low.
    high = low + range*high_range(symbol)
    low = low + range*low_range(symbol)
End of While
output low
```

Following this process through to its natural conclusion with our chosen message looks like this:

New character	Low value	High value
	0.0	1.0
B	0.2	0.3
I	0.25	0.26
L	0.256	0.258
L	0.2572	0.2576
SPACE	0.25720	0.25724
G	0.257216	0.257220
A	0.2572164	0.2572168
T	0.25721676	0.2572168
E	0.257216772	0.257216776
S	0.2572167752	0.2572167756

So the final low value, 0.2572167752 will uniquely encode the message "BILL GATES" using our present encoding scheme.

Given this encoding scheme, it is relatively easy to see how the decoding process will operate. We find the first symbol in the message by seeing which symbol owns the code space that our encoded message falls in. Since the number 0.2572167752 falls between 0.2 and 0.3, we know that the first character must be "B". We then need to remove the "B" from the encoded number. Since we know the low and high ranges of B, we can remove their effects by reversing the process that put them in. First, we subtract the low value of B from the number, giving 0.0572167752. Then we divide by the range of B, which is 0.1. This gives a value of 0.572167752. We can then calculate where that lands, which is in the range of the next letter, "I".

The algorithm for decoding the incoming number looks like this:

```
get encoded number
Do
    find symbol whose range straddles the encoded number
    output the symbol
    range = symbol low value - symbol high value
    subtract symbol low value from encoded number
    divide encoded number by range
until no more symbols
```

2.3 Quantization

Digital transmission of information has come to dominate the design of communications systems to such an extent that information has become synonymous with bits. The generation of bits from a continuous valued source inevitably involves some form of quantization, which is simply the approximation of a quantity with an element chosen from a discrete set. Assuming the interface between a source encoder and a channel encoder is discrete, a finite-rate source coder is synonymous with a quantizer.

The communication of any information, continuous or discrete, through quantized values seems very natural today. However, this idea was rather new in Shannon's time. Pulse-Code Modulation (PCM), which is nothing more than quantization for subsequent digital communication, was patented in 1939 [19]. The first fully deployed use of PCM was for a military communication system used in 1945 [20] and described openly in 1947 [21].

The simplest form of quantization is fixed-rate quantization of a scalar source. Here a real-valued source is mapped to one of a finite number of values. Symbolically, the quantizer is a mapping $Q : R \rightarrow C$ where

$$C = \{y_1, y_2, ..., y_K\} \subset R,\ y_1 < y_2 < ... < y_K \tag{2.10}$$

is called the *codebook*. This can be decomposed into *encoding* or “quantization”

$$\varepsilon : R \rightarrow \{1,\ 2,\ ...,\ K\} \tag{2.11}$$

and decoding or “inverse quantization”

$$d : \{1,\ 2,\ ...,\ K\} \rightarrow C \tag{2.12}$$

operations, though this is usually not necessary. Each inverse image $Q^{-1}(y_i)$ is called a *cell*, and the cells together form the *partition* induced by the quantizer.

Except in the degenerate case where the source takes on no more than K values, the input and output of the quantizer will differ. This difference is called the quantization error. Naturally, this error should be made small, so we minimize a nonnegative measure of the error. The most common distortion measure is the mean-squared error (MSE), defined for the random variable *X* and the quantizer *Q* by

$$D = E\left[(X - Q(X))^2\right] \tag{2.13}$$

where $E[.]$ denotes expectation.

With the MSE distortion measure, there are two simple necessary conditions for an optimal quantizer:

- *Nearest neighbor encoding:* Consider the codebook to be fixed. In encoding a source sample *x*, one should choose the codebook element closest to *x*. This is written as

$$Q(x) = \arg\min_{y_i \in C} |x - y_i| \tag{2.14}$$

In this case the cell is called a voronoi cell.

- *Centroid condition:* With the encoder mapping fixed, the decoder minimizes the distortion by decoding to the average value of the cell:

$$d((y_i) = E[x / Q(x) = y_i] \tag{2.15}$$

Except for a few examples with simple probability density functions, optimal quantizer design cannot be done analytically. Lloyd [22] and Max [23] independently suggested that a quantizer be designed iteratively by alternating enforcements of the above conditions. A quantizer designed in this manner is called a Lloyd-Max quantizer.

The output of a Lloyd-Max quantizer is a random variable taking values in a discrete set of size *K*. The average number of bits required to transmit such a random variable can usually be reduced by entropy coding. There is no reason to believe that cascading a Lloyd-Max quantizer and an entropy coder would give the best trade-off between average rate and distortion [24]; in fact, a joint design of the quantizer and entropy coder is beneficial. The result of a joint design is called an entropy-constrained scalar quantizer. Its first numerical investigation was by Wood [25]. Optimality conditions for MSE distortion were provided by Berger [26].

Vector quantization is merely the extension of scalar quantization to multidimensional domains, but the implications of this extension are profound. The source coding of any discrete domain source like an image, audio segment, or video segment can be considered a single vector quantization operation. This is in contrast to having many scalar quantization operations, say one for each sample of an audio signal. Vector quantization is by definition the ultimate coding scheme because it includes all forms of source coding.

Vector Quantization (VQ) is actually the structure used by Shannon [27] in his theoretical studies of coding with a fidelity criterion, thus it is as old rate-distortion theory. Interest in VQ rose in the 1980's as the use of digital computers made its implementation more practical. It should be noted, however, that unconstrained VQ is feasible for vectors of much lower dimension than an entire source realization (image, audio segment, video segment) described above.

The two scenarios for scalar quantization considered above, optimum for a fixed codebook size and entropy-constrained, have been addressed for VQ as well. For the first problem, a design algorithm which generalizes the Lloyd-Max iteration was given by Linde, Buzo, and Gray [28]. It is called the generalized Lloyd, or LBG for the authors, algorithm. An algorithm for entropy-constrained VQ (ECVQ) design was given by Chou, Lookabaugh, and Gray [29]. The main drawback of VQ is its complexity, both in time (number of operations) and space (amount of memory). Unless some structure is imposed on the codebook, each codeword must be stored independently. For a codebook of size K, the nearest neighbor encoding process requires K distance computations and the selection of the smallest of the K distances. Table 2.1 summarizes a complexity comparison between scalar and vector quantization for an N-dimensional source coded at R bits per component. It is assumed that the scalar codebooks used for each component are distinct but equal in size [24].

	Scalar quantization	Vector quantization
Space complexity:		
Codebook size	$N2^R$	2^{NR}
Time complexity:		
distance calculations	$N2^R$	2^{NR}
complexity of distance calculation	$O(1)$	$O(N)$
overall	$O\left(N2^R\right)$	$O\left(N2^{NR}\right)$

Table 2.1: Complexity comparison between scalar and vector quantization. N represents the dimensionality of the source and R the rate in bits per source component. It is assumed that different scalar quantizers are used for each vector component; otherwise, codebook size is reduced from $N2^R$ to 2^R.

Many techniques for reducing time and space complexity by placing constraints on the codebook design (using a suboptimal codebook) or by replacing the nearest neighbor encoding rule (using suboptimal encoding) have been proposed. The most important variants are described in [30], a very interesting text on VQ.

The design of optimal quantizers can rarely be completed analytically. While this is a practical problem in its own right, it poses distinct difficulty in the design of systems consisting of more than just a quantizer. It is useful to be able to model the quantization process in a general way. Several approaches that lead to tractable analysis are to assume the quantization is fine, uniform, non-uniform or dithered [24].

2.3.1 Fine quantization approximation

Consider first, Lloyd-Max quantization, i.e., scalar quantization without entropy coding. If the size of the codebook is large and the probability density function (p.d.f.) of the source is smooth, then the p.d.f. is approximately constant over each quantization cell. This approximation, attributed to Bennett [31], facilitates analytical approximations of the distortion as a function of the size of the codebook *K* and point density function of the quantizer [30]. In any cell, the point density function is approximately the width of the cell divided by *K*.

The optimum point density function is proportional to the cube root of the p.d.f. and yields distortion of

$$D \approx \frac{1}{12K^2}\left(\int_{-\infty}^{+\infty} f_X(x)^{1/3}\,dx\right)^3 \tag{2.15}$$

For a Gaussian random variable with variance σ^2, this becomes

$$D \approx \frac{\sqrt{3}\pi\sigma^2}{2K^2} \tag{2.16}$$

Upon relating the rate to the number of cells with $R = \log_2 K$, we obtain

$$D_{Gaussian,Lloyd-Max} \approx \frac{\sqrt{3}\pi}{2}\sigma^2 2^{-2R} \tag{2.17}$$

Bennett's approximation can be used to analyze entropy-coded scalar quantization (ECSQ) as well. The results are rather startling: Gish and Pierce [32] showed that under weak assumptions on the source p.d.f. and on the distortion criterion, the asymptotically optimal point density is constant on the support of the source p.d.f. For a Gaussian source, one obtains

$$D \approx \frac{\pi e}{6}\sigma^2 2^{-2R} \tag{2.18}$$

For comparison, the distortion-rate function for a memoryless Gaussian source with the same variance is

$$D(R) = \sigma^2 2^{-2R} \tag{2.19}$$

2.3.2 Uniform quantization

Uniform quantization is usually not optimal, but it is very common in practice because of its simplicity. A uniform quantizer is distinguished by having equally spaced codewords. Two types of uniform quantizers are of interest: one with a finite number of codewords (a bounded quantizer) and one with a countably infinite number of codewords (an unbounded quantizer).

A bounded uniform quantizer is characterized by the number of cells *K*, the quantization step size Δ, and any single codebook value, say y_1. The remaining codebook entries are given by $y_k = y_1 + (k-1)\Delta$. The interval $I_{granular} = \left\lfloor y_1 - \Delta/2, y_K + \Delta/2 \right\rfloor$ is called the granular region. Source samples in this interval will be approximated within $\pm\Delta/2$ by their quantized values if midpoint reconstruction is assumed. Samples in the overload region $I_{overload} = \mathbb{R} \setminus I_{granular}$ face quantization error greater than $\Delta/2$ in absolute value. Overload distortion and granular distortion refer to the expected value of each type of distortion.

For a fixed value of *K*, there is a trade-off in selecting the value of Δ. Decreasing Δ diminishes the granular distortion, but also contracts the granular region, enhancing the overload distortion. Similarly, attempting to minimize the overload distortion enhances the granular distortion. The size of Δ can be described by a dimensionless quantity called the loading factor. The *loading factor* is the length of I_{granular} divided by twice the standard deviation of the source. For Gaussian and Laplacian sources, the optimal loading factor as a function of the bit rate is shown in Figure 2.4. The difference is due to the heaviness of the distribution tails [24].

An unbounded uniform quantizer is described by a quantization step size Δ and an offset $a \in [-\Delta/2, \Delta/2)$. The quantization function is then given by

$$\begin{aligned} Q_{\Delta,a}(x) &= n\Delta - a \text{ if } x \in \left[\left(n - \frac{1}{2}\right)\Delta - a, \left(n + \frac{1}{2}\right)\Delta - a \right) \\ &= [x+a]_\Delta - a \end{aligned} \tag{2.20}$$

where $[.]_\Delta$ represents rounding to the nearest multiple of Δ. This is most commonly used with a equal to zero or $-\Delta/2$. An unbounded uniform quantizer only makes sense when followed by an entropy code.

Except in the small Δ limit, few analytical calculations can be made regarding the performance of entropy-coded unbounded uniform quantization (ECUQ). The use of Bennett approximations for small Δ yields the optimality of ECUQ mentioned previously. At high

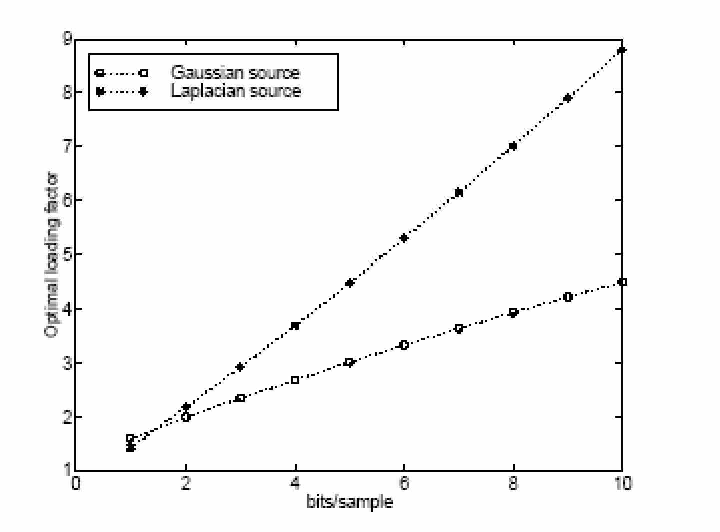


Figure 2.4: Optimal loading factors for bounded uniform quantization of Gaussian and Laplacian sources.

rates, ECUQ performs within $\frac{1}{2}\log_2\left(\pi e/6\right) \approx 0.255$ bits per sample of the rate-distortion function for any memoryless source [32]. A numerical study has shown that for a wide range of memoryless sources, ECUQ is within 0.3 bits per sample of the rate-distortion function at all rates [33].

2.3.2.1 Example of uniform scalar quantization

An example of a uniform scalar quantizer is depicted in figure 2.5 where the quantization step size Δ is equal to 1. The granular region is divided into intervals $I_i = [x_{i-1}, x_i]$. The output values y_i are typically chosen in the intervals I_i. The set $\{y_i\}$ and y_i are respectively the dictionary (codebook) and centroid.

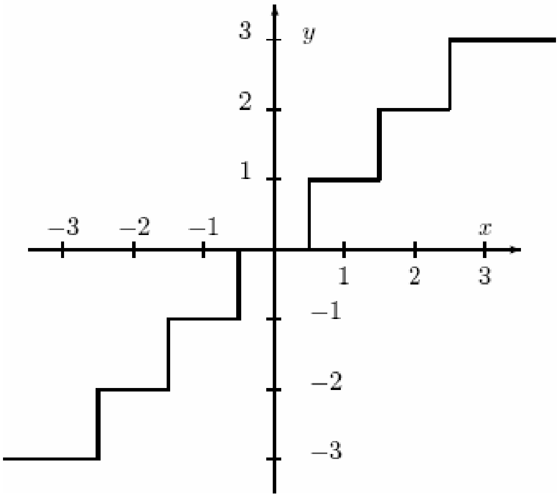


Figure 2.5: Uniform scalar quantization.

Let $x(n)$ a discrete-time signal uniformly distributed in the interval $[a,b]$. It is considered as a realization of a centered random process $X(n)$ (this signal is, in fact, a non realistic signal especially in the case of images, but it will help to simplify computations and give a concise formula of the quantization error power in terms of signal power and resolution). The resolution $R = \lceil \log_2 N \rceil$ is a function of the number of intervals N dividing the granular region $[a,b]$.

To define a uniform scalar quantizer, one should:

1. divide the interval $[a,b]$ into $N = 2^R$ distinct intervals of equal length $\Delta = {(b-a)}\big/{2^R}$;
2. associate a number "i" to each interval;
3. define a representing value "y_i" for each interval, for example the midpoint of the interval.

The coding operation consists of finding the interval to which belongs a given input value $x(n)$ and associating the interval number $i(n) \in \{1 \ldots N\}$ to this value. The decoding operation consists of assigning to the number $i(n)$ the corresponding $y_{i(n)}$ from the dictionary.

$$\hat{x}(n) = y_{i(n)} \tag{2.21}$$

Thus, the coding is a two-step operation:

$$x(n) \xrightarrow{coding} i(n) \xrightarrow{decoding} y_{i(n)}$$

and the quantization error is given by

$$\varepsilon(n) = x(n) - \hat{x}(n) \tag{2.22}$$

where $x(n)$ and $\hat{x}(n)$ are respectively the input and output of the quantizer.

The quantizer performance is, generally, measured through the MSE which we often seek to minimize or in other sense maximizing the signal to noise ratio (SNR)

$$D = E\left[\left|X(n) - \hat{x}(n)\right|^2\right] \tag{2.23}$$

$$SNR = \frac{E\left[X^2(n)\right]}{E\left[\left|X(n) - \hat{x}(n)\right|^2\right]} \tag{2.24}$$

The quantization error $\varepsilon(n)$ is a deterministic function of $x(n)$ with a uniform distribution [34]. The quantization error mean is zero and its variance is given by

$$\sigma_{\varepsilon}^{2} = E\left[\varepsilon^{2}\right] = \int_{-\Delta/2}^{\Delta/2} x^{2} \frac{1}{\Delta} dx \tag{2.25}$$

For N intervals representing the whole interval $\left[a, b\right]$, the variance σ_{ε}^{2} becomes

$$\sigma_{\varepsilon}^{2} = \frac{\Delta^{2}}{12} = \frac{1}{12}\frac{(b-a)^{2}}{N^{2}} = \frac{1}{12}\frac{(b-a)^{2}}{2^{2R}} \tag{2.26}$$

The variance of the signal $X(n)$ is given by

$$\sigma_{X}^{2} = E\left[X^{2}(n)\right] = \frac{(b-a)^{2}}{12} \tag{2.27}$$

then, the SNR in dB is

$$SNR = 10\log_{10}\frac{E\left[X^{2}(n)\right]}{E\left[\varepsilon^{2}\right]} = 10\log_{10} 2^{2R} = 6.02R \tag{2.28}$$

thus, the SNR of uniform quantization increases 6 dB for each increase of one bit/pixel.

2.3.3 Dithered quantization

Dithering is the addition of a random signal prior to quantization. The dither signal may or may not be subtracted after quantization, yielding subtractive and nonsubtractive dithered quantization, respectively. The purpose of this operation is to manipulate the statistical properties of the quantization noise, e.g., to make them independent of the signal.

In subtractive dithered quantization (SDQ), an unbounded uniform quantizer and a white dither signal uniformly distributed on $[-\Delta/2, \Delta/2)$ is assumed. This SDQ is equivalent to the use of Equation (2.20) with randomized offset a. The quantization noise is signal-independent, white, and uniformly distributed, regardless of the signal p.d.f. and Δ.

The first use of SDQ was by Roberts [35] to improve the perceptual quality of PCM-encoded images. Subtractive and nonsubtractive dithered quantization is surveyed in [36, 37].

2.4 Digital image preliminaries and distortion measures

2.4.1 Preliminaries of digital images

In contrast with digital speech and audio data, image data spans over space rather than over time. The image signals are usually band-limited analog signals, which are converted to digital form through sampling at or above the Nyquist sampling rate [38]. As images are two-dimensional, discretisation is done in both dimensions. Each discretised point in an image is known as a picture element or a pixel.

For grayscale images, each pixel has a certain scalar quantized value which represents the luminance. For a b-bit grayscale images, the luminance goes from 0 (black) to $2^b - 1$ (white). Therefore, for a one-bit image, luminance values are either 0 or 1. Such binary images are used commonly for facsimiles [38]. Grayscale images used in research are commonly 8-bit images, where each pixel has a luminance within the range of 0 to 255 and all values in between are a certain level of grey. An example of an 8-bit image is given in Figure 2.6. The rectangular arrangement of pixels in an image allows them to be represented as a two dimensional array. This representation allows the image to be specified as a data type for use in coding algorithms. Individual pixels can be quantized using scalar quantizers or differential pulse code modulation (DPCM), which exploits spatial redundancy from neighboring pixels. Rows and columns of an image can be extracted as one dimensional vector from the array. This allows direct manipulation by various algorithms such as subband and wavelet coding, where row and column filtering can be performed separately. Likewise, due to the two dimensional nature of image data, statistical dependencies exist across both dimensions, which suggests that square or

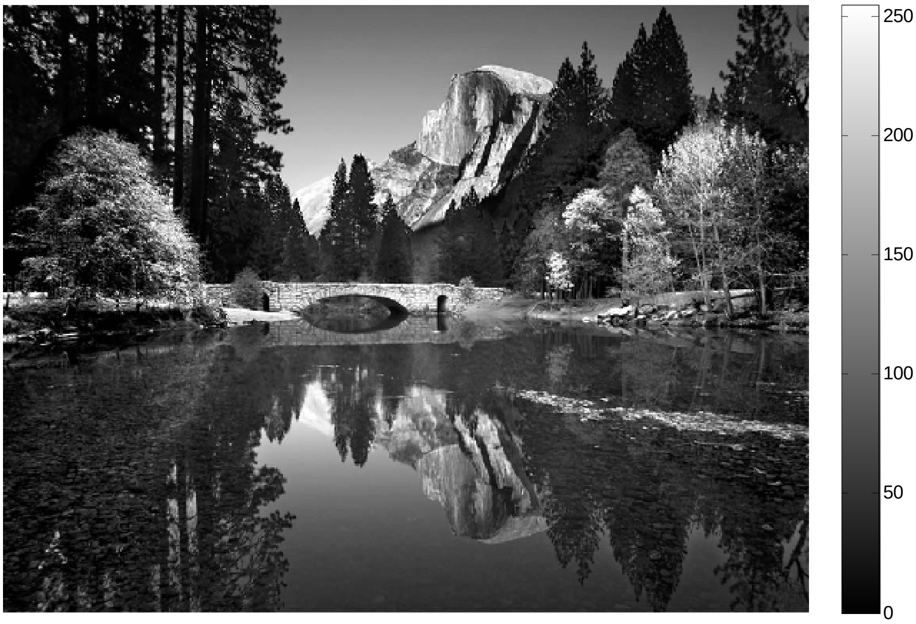


Figure 2.6: A typical 8-bit grayscale image “Yosemite” with mapping between color and luminance value.

rectangular blocks of pixels are a suitable atomic data representation for block-based quantization schemes, such as transform coding and vector quantization, to exploit these dependencies [39].

Assume an isotropic covariance function for image data [40]

$$\sum(i, j) = \sigma^2 e^{-\alpha\sqrt{i^2+j^2}} \tag{2.29}$$

where σ^2 is the variance of the image and i and j is the distance away from the reference pixel. It can be shown, if the correlation is assumed uniform in both dimensions, that the covariance function of any pixel, which indicative of spatial correlation, decreases rapidly beyond the vicinity of 8 neighboring pixels [40]. Therefore, a popular block size used in block-based image coding is $8 \text{ x } 8$ pixels, which form vectors of dimension 64.

2.4.2 Distortion measures for image coding

For images, the amount of compression is usually expressed as the bitrate which is measured in bits per pixel (bits/pixel or bpp). In other words it is the average number of bits required to represent one pixel.

In general, distortion measurements are used on lossy compressed (whereas in lossless compression, no information is lost during compress and hence the distortion is zero) or transmitted images to try and quantify the quality of a picture. This quality is a very subjective measure as it may mean that the reproduced image is to be used for casual human viewing (i.e. TV image) or it may be needed for precise, quantitative measurement of some sort (i.e. medical image). These types and degrees of degradation become quite dependent on the situation for which the picture is being used. Getting a quantifiable measure of the distortion between two images is very important as one can try and minimize this distortion so as to better replicate the original image.

There are many ways of measuring the fidelity of a reproduced image $\hat{I}$ to its original I. One of the simplest and most popular methods is to use the difference between I and $\hat{I}$. A common measure for the loss of information, in image coding, is the mean squared error (MSE), which is calculated using the following equation [41].

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left[I(i,j) - \hat{I}(i,j)\right]^2 \qquad (2.30)$$

where *m* and *n* are the dimensions of the image, $I(i,j)$ and $\hat{I}(i,j)$ are the pixel values at location (i,j) in the input and output image, respectively. A human observing two images affected by the same type of degradation will generally judge the one with the smaller MSE to be closer to the original. A very small MSE can be taken to mean that the image is very close to the original. However the MSE has some problems when images with different types of degradation are compared, the one with the smallest MSE will not necessarily seem closest to the original.

Signal-to-noise ratio (SNR) is another measure often used to compare the performance of reproduced images which is defined, in decibels, by

$$SNR = 10\log_{10}\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[I(i,j)\right]^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[I(i,j) - \hat{I}(i,j)\right]^2} \qquad (2.31)$$

or in terms of the mean square error

$$SNR = 10\log_{10}\frac{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left[I(i,j)\right]^2}{MSE} \qquad (2.32)$$

A common method of measuring the difference between two images is the Peak Signal to Noise Ratio, or PSNR which is defined by

$$PSNR = 10\log_{10}\frac{\left(2^b - 1\right)^2}{MSE} \qquad (2.33)$$

where *b* is the number of bits used to represent a pixel in the input image. The numerator is the square of the largest possible pixel value in the input image. For an 8-bit grayscale image, the PSNR is given by

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \qquad (2.34)$$

In our thesis, we have chosen to use PSNR as the quantitative method of comparing the image quality of a processed image to the original image. Unfortunately, PSNR does not necessarily take into account the perceived quality of an image, rather it only takes into account the overall similarity of pixel values between the original and processed image. For example, in Figure 2.7, the difference in PSNR between images (a) and (b) (about 18 dB) is much greater than that between images (b) and (c) and images (c) and (d) (approximately 5 dB, and 2 dB respectively), even though the perceivable degradation in quality between images (b) and (c) and between images (c) and (d) is much greater than that between images (a) and (b). Therefore, in addition to the objective analysis based on the computation of the PSNR, we will be using a subjective analysis of reconstructed image quality, using visual comparison, which will allow us to better evaluate the performances of the image coding techniques used in our thesis.

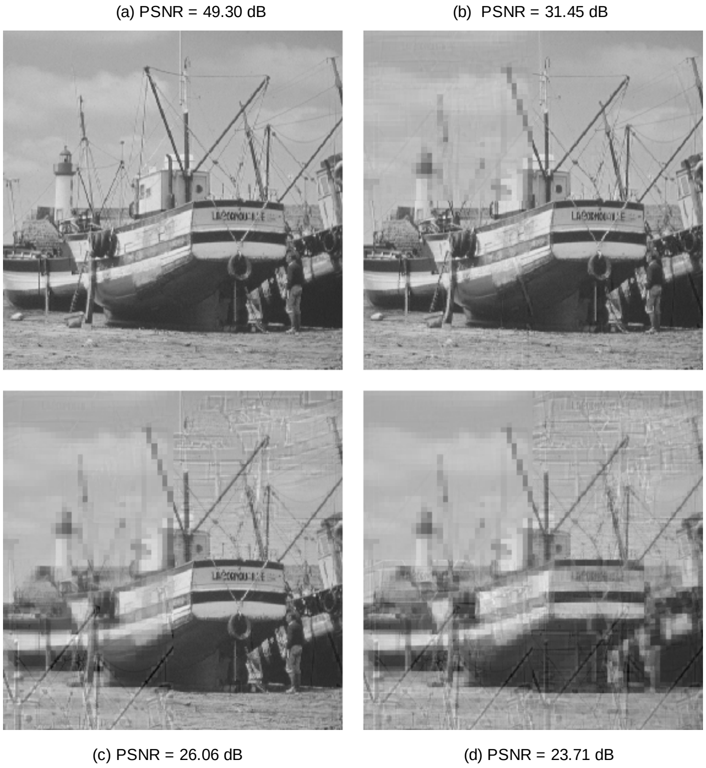


Figure 2.7: PSNR versus perceived image quality.

CHAPTER 3

TRANSFORM CODING OF IMAGES

Transform coding constitutes an integral component of contemporary image/video processing applications. Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels. Similarly in a video transmission system, adjacent pixels in consecutive frames show very high correlation. Consequently, these correlations can be exploited to predict the value of a pixel from its respective neighbors. A transformation is, therefore, defined to map this spatial (correlated) data into transformed (uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the information content of an individual pixel is relatively small i.e., to a large extent visual contribution of a pixel can be predicted using its neighbors.

A typical image transmission system is outlined in Figure 3.1 [42]. The objective of the source encoder is to exploit the redundancies in image data to provide compression. In other words, the source encoder reduces the entropy, which in our case means decrease in the average number of bits required to represent the image. On the contrary, the channel encoder adds redundancy to the output of the source encoder in order to enhance the reliability of the transmission.

As mentioned previously, each sub-block in the source encoder exploits some redundancy in the image data in order to achieve better compression. The transformation sub-block decorrelates the image data thereby reducing (and in some cases eliminating) interpixel redundancy [42].

The quantizer sub-block utilizes the fact that the human eye is unable to perceive some visual information in an image. Such information is deemed redundant and can be discarded without introducing noticeable visual artifacts. Such redundancy is referred to as psychovisual redundancy [42]. This idea can be extended to low bitrate receivers which, due to their stringent bandwidth requirements, might sacrifice visual quality in order to achieve bandwidth efficiency. This concept is the basis for rate distortion theory, that is, receivers might tolerate some visual distortion in exchange for bandwidth conservation. Lastly, the entropy encoder

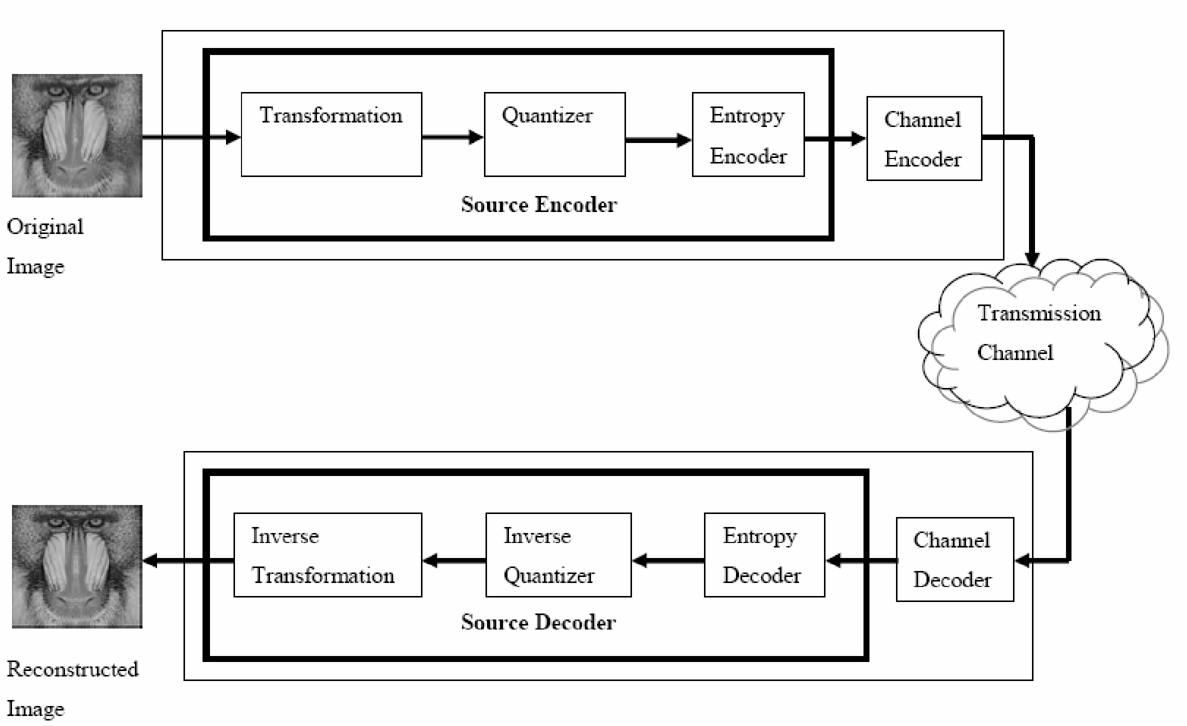


Figure 3.1: Components of a typical image transmission system.

employs its knowledge of the transformation and quantization processes to reduce the number of bits required to represent each symbol at the quantizer output.

In addition, transform coding is a method in which the image is decomposed into several components that are then coded according to their specific characteristics. The idea behind this method is to compact the information in the image into a few elements (called coefficients) that are easier to code. A simple way to illustrate this method would be to consider the data scatter plot shown in Figure 3.2 [43] where two actual data points are taken together to form a 2-D vector describing a single point in the 2-D plane. Now it can be seen that the data is mostly concentrated around the α axis when we use the $\alpha - \beta$ co-ordinate system. So, if the points are resolved along these axes, then their spread is smaller along one of the axes (the β axis). The "information" contained in two values of the original data now sequence gets more concentrated in one of the values and the change in the variance of the signal along the two directions leads to better compression [30]. Note that the transform itself does not compress the signal; it merely expresses the signal in a form that is more conducive to compression.

Mathematically, a transform projects the signal onto a set of basis functions. If $x = (x_1, x_2 ..., x_k)$is a *k*-dimensional vector in the original space and ***T*** is a k x k matrix of

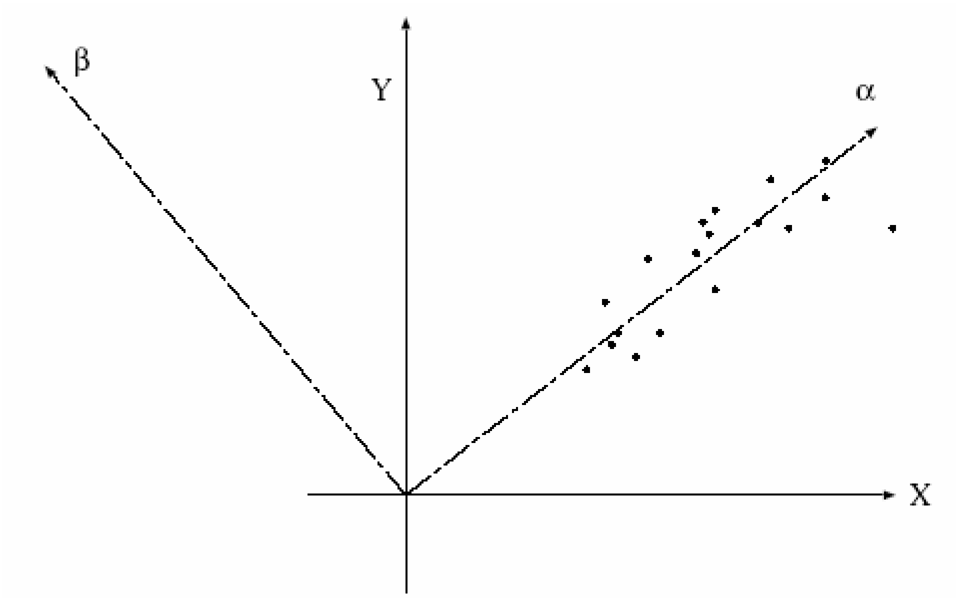


Figure 3.2: Transform coding: a geometric perspective.

basis vectors, then the projection $y = (y_1, y_2 ..., y_k)$, of x onto the basis functions is given by

$$y = Tx \tag{3.1}$$

The inverse transformation can be obtained by multiplying the vector y, by the inverse of the matrix T, T^{-1}

$$x = T^{-1} y \tag{3.2}$$

If the transformation is unitary, then $\left(T^*\right)^t T = I$, where $\left(T^*\right)^t$ is the transpose of the complex conjugate of the matrix *T* and *I* is the identity matrix. This property is useful in coding, because this means that the quantization error magnitude in the transform domain is the same as that in the signal domain and the quantizer can be optimized in the transform domain. When the signal is two dimensional (like in an image), the inputs and outputs of the transform coding system are described as matrices. Hence, now the forward transform equation becomes

$$Y = T^t X T \tag{3.3}$$

where *X* is an input matrix and *Y* the corresponding transformed matrix. Usually, separable transforms are used for image coding applications. This allows the 2-D transforms to be split into two 1-D transforms, which can then be applied separately along the horizontal and vertical directions.

3.1 Karhunen-Loeve Transform

A transform can also be viewed in terms of the changes in statistics between the original and transformed sequences [44]. In this perspective, a transform helps to decorrelate the samples of the signal (as mentioned before, a strong correlation exits between pixels of natural images). When the correlation between the samples is reduced, better signal compaction is possible. One such transform was proposed by Karhunen and Loeve and is known as the Karhunen-Loeve transform (KLT) [44].

The KLT is a linear orthogonal transform able to remove the correlation between pixels in an image. Let x be a vector with zero mean and autocorrelation matrix R_x. To remove the correlation between samples, a multiplication matrix A is to be found such that the elements of $y = Ax$ are uncorrelated. The autocorrelation matrix for y can be expressed as

$$R_y = E\left[(Ax)(Ax)^t\right] = AR_xA^t \tag{3.4}$$

Since R_x is already known, the task is then to find matrix A such that AR_xA^t is diagonal. A transform with A as the multiplication matrix is called a Karhunen-Loeve transform [45].

From the autocorrelation matrix R_y the orthogonal eigenvectors are found which by there own definition form the linear basis of the original data signal [46]. With these basis vectors it is possible to represent the original data signal as a linear combination of these basis vectors. This is very useful as the resultant transformed signal has had all linear correlation removed from it. If this transformed signal is then analyzed to see which transform coefficients contribute to the energy of the signal, a quantifiable measurement of redundant and essential information can be made. Since all linear transforms are unitary [46], the coefficients that contribute to the energy of the entire signal can be stored with the other redundant coefficients being thrown away or stored in some compressed way. If enough coefficients are found to be redundant then some compression can occur.

Though the KLT has optimal properties that are useful for transform coding, such as decorrelation and energy compaction, it has inherent disadvantages which have prevented it from being used in practical transform coders. Firstly, the transform is calculated from the source covariance (for zero-mean vectors the covariance matrix is equivalent to the autocorrelation matrix) matrix, and hence is source dependent [44, 45]. For the decoder to be able to perform the inverse transformation, it needs to know either the transformation matrix or the source covariance matrix, for the specific data to be decoded. These parameters need to be transmitted as side information, thus incurring transmission overhead. Alternatively, a static and global Karhunen-Loeve transform matrix can be derived from training data and used in both encoding and decoding. However, this leads to the problem of source mismatch and suboptimal coding for data that is not part of and is statistically different to the training set [47].

Secondly, there are sources where the covariance matrix is (or is close to being) singular which leads to the Karhunen-Loeve transform not being able to be uniquely defined [47]. It has been shown by Effros et al. [48] that for sources where the KLT is not unique, the `worst' KLT can give distortion performance that is 1.5 dB lower than the `best' KLT.

Lastly, the KLT can be a computationally complex operation. For a vector dimension of *n*, the Karhunen-Loeve transform requires $2n^2 - n$ operations [38]. It is not amenable to fast and efficient computation, unlike the fast Fourier transforms (FFT).

Fortunately, there is a substitute that achieves performance very close to that of the optimal KLT but yet has the advantage of a fixed set of basis vectors independent of the signal thus cutting computational complexity drastically. This transform is known as the Discrete-Cosine Transform (DCT).

3.2 Discrete Cosine Transform

The Discrete Cosine transform was introduced by Ahmed, Natarajan, and Rao in 1974 [49, 50]. The DCT is an orthogonal transform having some features of a transformation to the frequency domain. Like other transforms, the DCT attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency. This section describes the DCT and some of its important properties.

3.2.1 The One-Dimensional DCT

The most common DCT definition of a 1-D sequence $\{x_n\}$ of length *N* is

$$X(u) = \alpha(u) \sum_{n=0}^{N-1} x(n) \cos\left[\frac{\pi(2n+1)u}{2N}\right] \tag{3.5}$$

for $u = 0,1,2,...,N-1$. Similarly, the inverse transformation is defined as

$$x(n) = \sum_{u=0}^{N-1} \alpha(u) X(u) \cos\left[\frac{\pi(2n+1)u}{2N}\right] \tag{3.6}$$

for $n = 0,1,2,...,N-1$. In both Equations (3.5) and (3.6) $\alpha(u)$ is defined as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u \neq 0 \end{cases} \tag{3.7}$$

It is clear from Equation (3.5) that for $u = 0$, $X(u=0) = \sqrt{\frac{1}{N}} \sum_{n=0}^{N-1} x(n)$. Thus, the first transform coefficient is the average value of the sample sequence. In literature, this value is referred to as the DC Coefficient. All other transform coefficients are called the AC Coefficients.

To fix ideas, ignore the sequence $\{x_n\}$ and $\alpha(u)$ component in Equation (3.5). The plot of $\cos\left[\frac{\pi(2n+1)u)}{2N}\right]$ for $n = 0,1,...,N-1$ with $N = 8$ and varying values of u is shown in Figure 3.3. In accordance with our previous observation, the first the top-left waveform (u = 0) renders a constant (DC) value, whereas, all other waveforms $(u = 1,2,...,7)$ give waveforms at progressively increasing frequencies [51]. These waveforms are called the *cosine basis function*. Note that these basis functions are orthogonal. Hence, multiplication of any waveform in Figure 3.3 with another waveform followed by a summation over all sample points yields a zero (scalar) value, whereas multiplication of any waveform in Figure 3.3 with itself followed by a summation yields a constant (scalar) value. Orthogonal waveforms are independent, that is, none of the basis functions can be represented as a combination of other basis functions.

If the input sequence has more than N sample points then it can be divided into sub-sequences of length N and DCT can be applied to these chunks independently. Here, a very important point to note is that in each such computation the values of the basis function points will not change. Only the values of $\{x_n\}$ will change in each sub-sequence. This is a very important property, since it shows that the basis functions can be pre-computed offline and then multiplied with the sub-sequences. This reduces the number of mathematical operations (i.e., multiplications and additions) thereby rendering computation efficiency.

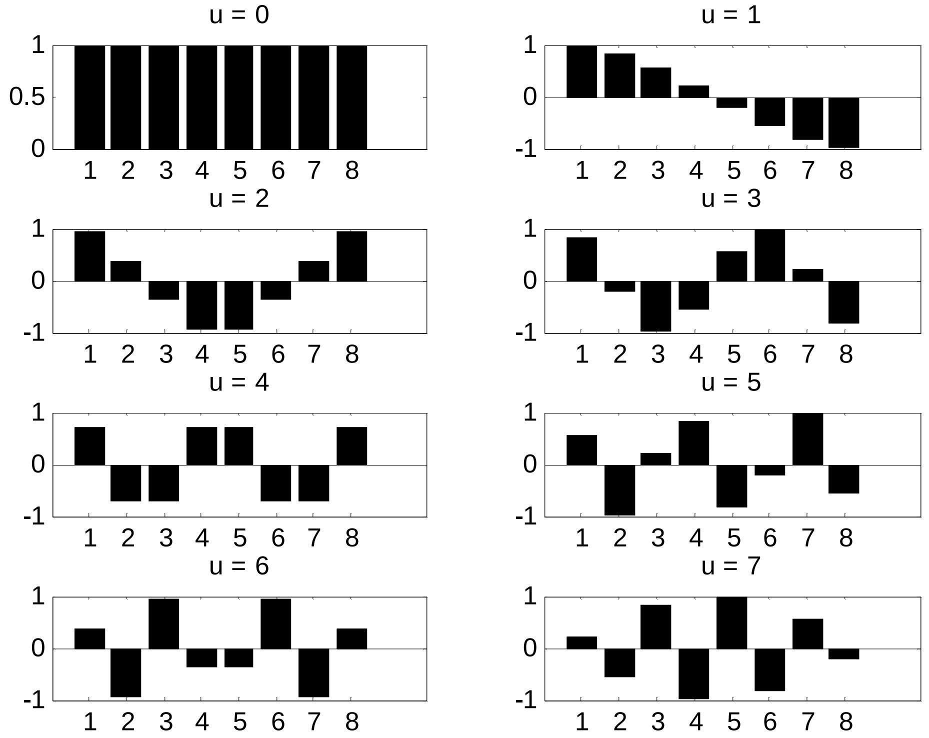


Figure 3.3: One dimensional cosine basis function (*N*=8).

3.2.2 The Two-Dimensional DCT

When actually implemented into an algorithm the two dimensional DCT is usually performed on subblocks of an image with 8x8 to 16x16 pixel block sizes. This is due to the two dimensional transform being very computationally expensive to apply to the entire image. Statistics of an image are generally stationary in 8x8 or 16x16 pixel regions, if larger block sizes are considered, any high frequency coefficients that occur in the transformed block could be due to statistical change in the image as it moves between regions instead of the actual detail of the block [41]. The 2-D DCT is a direct extension of the 1-D case and is given by [42]

$$X(u,v) = \alpha(u)\alpha(v)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(n,m)\cos\left[\frac{\pi(2m+1)u}{2N}\right]\cos\left[\frac{\pi(2n+1)v}{2N}\right] \quad (3.8)$$

As mentioned in Section 3.1 the basis vectors used to represent the original signal block are signal independent. These linear basis functions can be seen in Figure 3.4 for an 8x8 block. The use of a universal set of basis functions is highly valued as it gets around the problem of storing basis functions every time a transform is required, as is the case when using the Karhunen-Loeve transform.

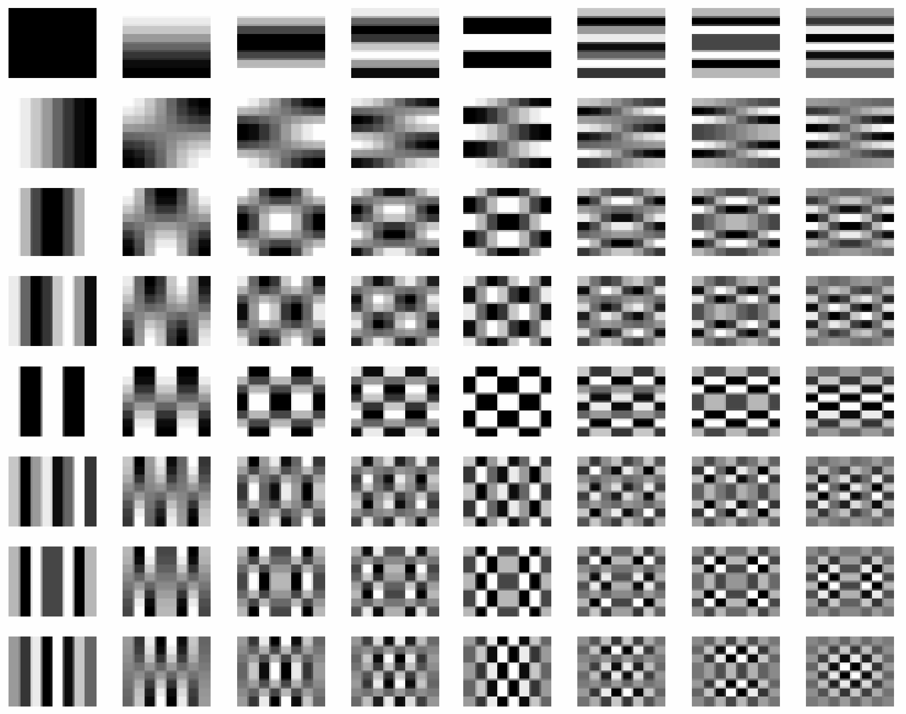


Figure 3.4: Basis functions of an 8x8 DCT.

3.2.3 DCT Coding

The actual DCT image block encoder/decoder can be best described in Figures 3.5 and 3.6 respectively [41]. From these basic diagrams there seems more to the encoding process than simply applying a 2 dimensional DCT to the image subblocks. Firstly, there is the question of quantizing the transform coefficients. This is performed by using a M x M quantization table, which is specified depending on the quality factor required by the user. The lossy aspect of transform coding comes into play at this point of time, as the higher the quality factor required the more bits that are required to the store the image. Conversely, for a lower quality factor fewer bits are required. Quantization is defined as division of each DCT coefficient by its corresponding quantizer step size, followed by rounding to the nearest integer as defined by

$$X_c^Q(n,m) = \left[\frac{X_c(n,m)}{Q(n,m)} \right]_Q \qquad (3.9)$$

with [.] denotes rounding.

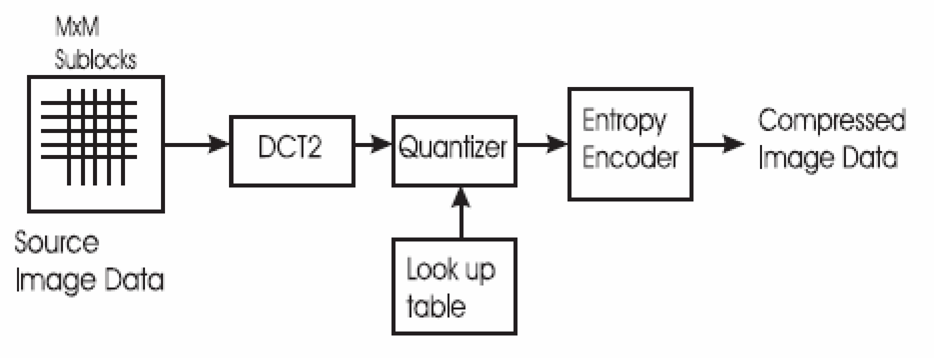


Figure 3.5: DCT-based encoder processing steps.

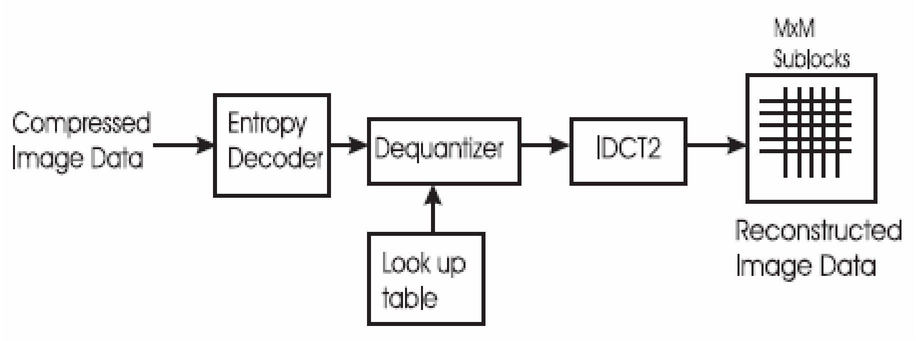
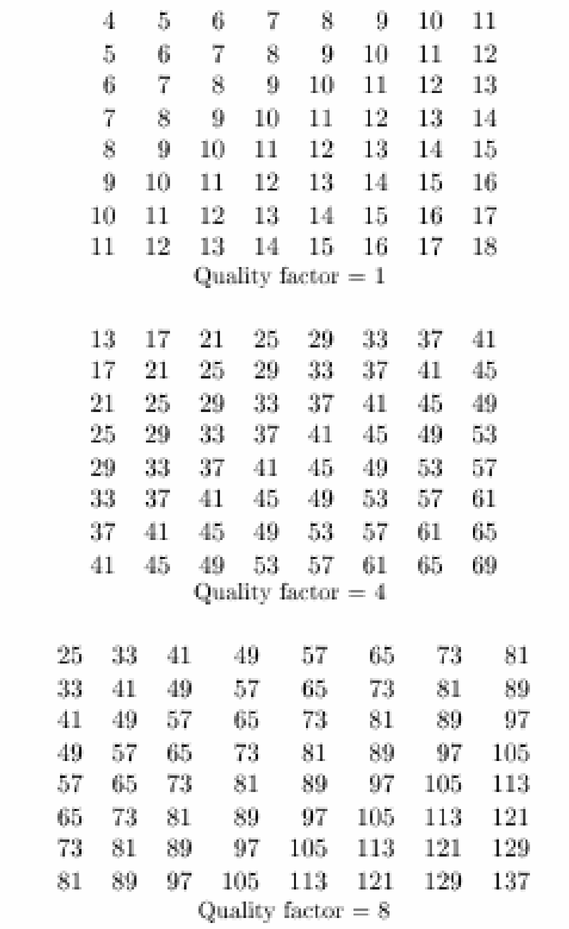


Figure 3.6: DCT-based decoder processing steps.

The output value obtained by applying Equation 3.9 is normalized by the quantizer step size. Dequantization is the inverse function of Equation 3.9, which in this case means simply that the normalization is removed by multiplying by the step size, which returns the result to a representation appropriate for the inverse DCT.

The actual quantization table was created via heuristic approach, as it should ideally be chosen based on the perceptual threshold of each frequency subband. The implementation used in this case was that of a linear table whose quantization intervals increase as the spatial frequency increases. This was done to eliminate any non essential high frequency components which are not as perceptually important as their low frequency counterparts. The quality factor can be simplified down to a single quality factor number ranging typically from 1 (being good) to 25 (being the most coarse quantization available). An example of quantization table for an 8x8 block at different quality factors can be seen in Figure 3.7. By analyzing this table it can be noticed that the quantization intervals become larger as they extend out to the higher frequency regions. As the quality factor is increased, so are the overall quantization intervals. An example of this can be seen in the table for quality factor 8 shown in Figure 3.7.

At the highest frequency component, located in the bottom left corner of the 8x8 block, a value greater than 137 is required for this frequency component not to be quantized to zero.



4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14
8	9	10	11	12	13	14	15
9	10	11	12	13	14	15	16
10	11	12	13	14	15	16	17
11	12	13	14	15	16	17	18

Quality factor = 1

13	17	21	25	29	33	37	41
17	21	25	29	33	37	41	45
21	25	29	33	37	41	45	49
25	29	33	37	41	45	49	53
29	33	37	41	45	49	53	57
33	37	41	45	49	53	57	61
37	41	45	49	53	57	61	65
41	45	49	53	57	61	65	69

Quality factor = 4

25	33	41	49	57	65	73	81
33	41	49	57	65	73	81	89
41	49	57	65	73	81	89	97
49	57	65	73	81	89	97	105
57	65	73	81	89	97	105	113
65	73	81	89	97	105	113	121
73	81	89	97	105	113	121	129
81	89	97	105	113	121	129	137

Quality factor = 8

Figure 3.7: Quantization tables for 8x8 blocks at different quality factors.

Using these quantization tables alone is not enough to take advantage of the redundancies that arise from transform coding. The DC coefficient located in the top left hand corner of a typical transformed 8x8 block can be separately encoded from the AC coefficients. This is due to the DC coefficient being a measure of the average intensity of the 64 image samples and contains most of the energy for the entire block. There is usually a strong correlation between the DC coefficients in adjacent 8x8 blocks. If these quantized DC coefficients are entropy coded based on their statistics alone, a marked improvement in compression can occur when compared to entropy coding the DC coefficients in conjunction with the AC coefficients. This difference in compression can be accounted for in the large difference that occurs in the statistical properties between AC coefficients, which usually have a Laplacian like statistical distribution [50], and DC coefficients, whose statistics vary quite markedly depending on the image. There is a further reason for this separate coding, in that the AC coefficients usually have large runs of zeros due to the coarse quantization they undergo. These runs of zeros can be taken advantage of by run length encoding.

Run length encoding expresses a run of characters of the same value by simply storing the value of the character and the number that occurred instead of writing out each character N times. This form of compression is ideally suited to transform coding, after the AC quantized coefficients are ordered into the zigzag sequence similar to Figure 3.8. This ordering helps to facilitate entropy coding and run length coding by placing low frequency coefficients (which are more likely to be nonzero) before high frequency coefficients. This method of DCT image coding broadly forms the basis for the image coding standard known as JPEG (Joint Photographic Experts Group) [52].

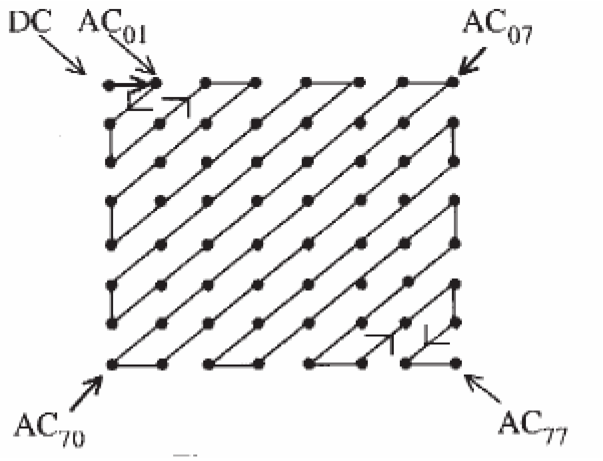


Figure 3.8: Zigzag sequence.

3.2.4 Properties of DCT

In the preceding sections, a mathematical foundation for DCT has been developed. However, the intuitive insight into its image processing application has not been presented. In this section, we outline (with examples) some properties of the DCT which are of particular value to image processing applications.

3.2.4.1 Decorrelation

As discussed previously, the principle advantage of image transformation is the removal of redundancy between neighboring pixels. This leads to uncorrelated transform coefficients which can be encoded independently. Let us consider the example from Figure 3.9 to outline the decorrelation characteristics of the 2-D DCT. Figures 3.9 (c) and (d) show the normalized autocorrelation among pixels in one line of the respective images (a) and (b) before DCT. The normalized autocorrelation of the two images after DCT is shown in Figures 3.9 (e) and (f). Clearly, the amplitude of the autocorrelation after the DCT operation is very small at all lags. Hence, it can be inferred that DCT exhibits excellent decorrelation properties.

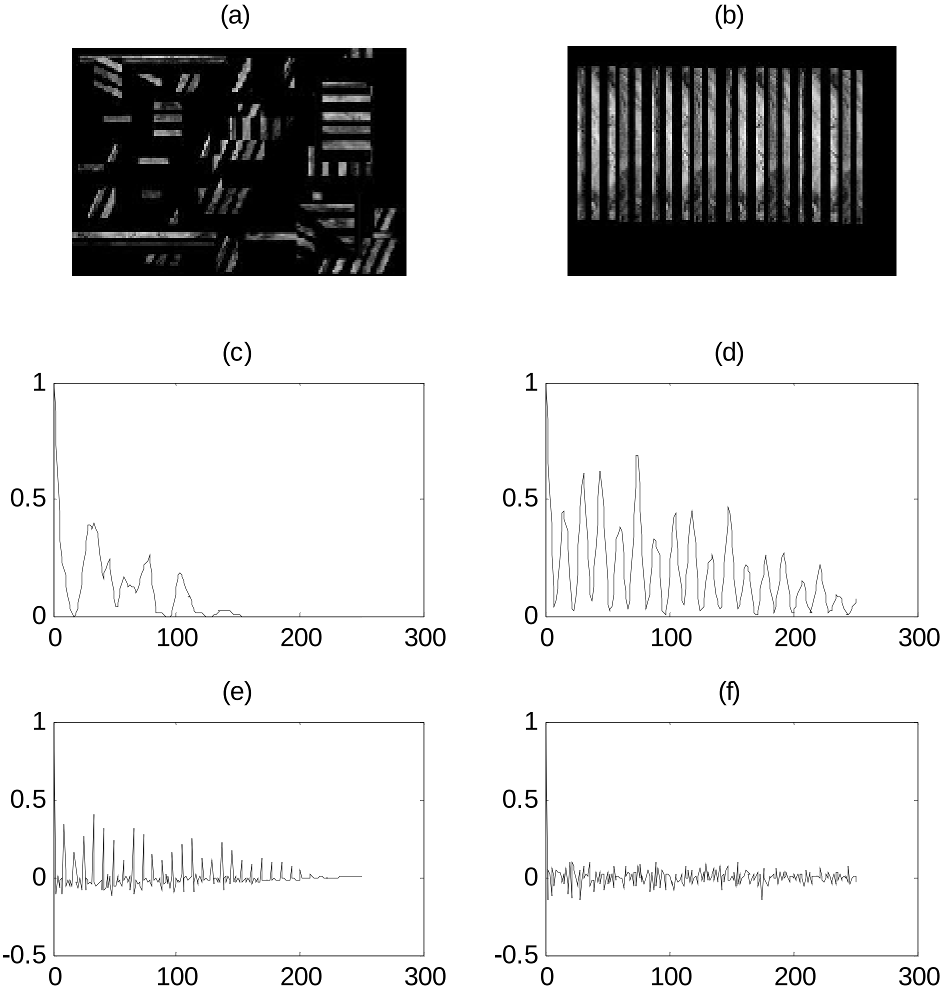


Figure 3.9: (a) First image; (b) Second image; (c) Normalized autocorrelation of uncorrelated image (first image) before DCT; (d) Normalized autocorrelation of correlated image (second image) before DCT; (e) Normalized autocorrelation of uncorrelated image after DCT; (f) Normalized autocorrelation of correlated image after DCT.

3.2.4.2 Energy Compaction

Efficacy of a transformation scheme can be directly gauged by its ability to pack input data into as few coefficients as possible. This allows the quantizer to discard coefficients with relatively small amplitudes without introducing visual distortion in the reconstructed image. DCT exhibits excellent energy compaction for highly correlated images. Let us consider the two example images of Figures 3.9 (a) and (b). In addition to their respective correlation properties discussed in preceding sections, the uncorrelated image has more sharp intensity variations than the correlated image. Therefore, the former has more high frequency content than the latter. Figure 3.10 shows the DCT of both images. Clearly, the uncorrelated image has its energy spread out, whereas the energy of the correlated image is packed into the low frequency region (i.e., top left region).

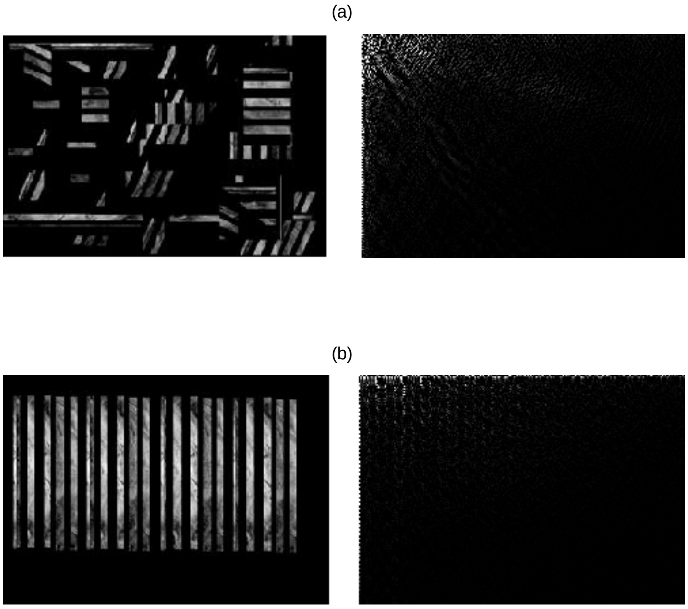


Figure 3.10: (a) uncorrelated image and its DCT; (b) correlated image and its DCT.

One of the disadvantages with DCT-based transform coding is the introduction of block artifacts, as seen in Figure 3.11. This problem is caused by the discontinuities that result from the rectangular windowing of the image data. Various methods of reducing blocking artifacts include the use of overlapping blocks and low-pass filtering boundary pixels [53]. Disadvantages of these methods include an increase in bitrate and blurring, respectively [54]. Malvar and Staelin [54] investigated the lapped orthogonal transform (LOT). The idea of the LOT is to map blocks of n samples, to n basis functions which are l samples in length, such that $l > n$. Therefore, longer blocks of l samples are formed from the smaller blocks which overlap each other by l-n samples [54]. This achieves the overlapping effect but since there are only n transform coefficients to quantize, there will be no increase in bitrate. Furthermore, the LOT basis functions decay toward zero at their boundaries, which leads to a reduction in block artifacts [54].

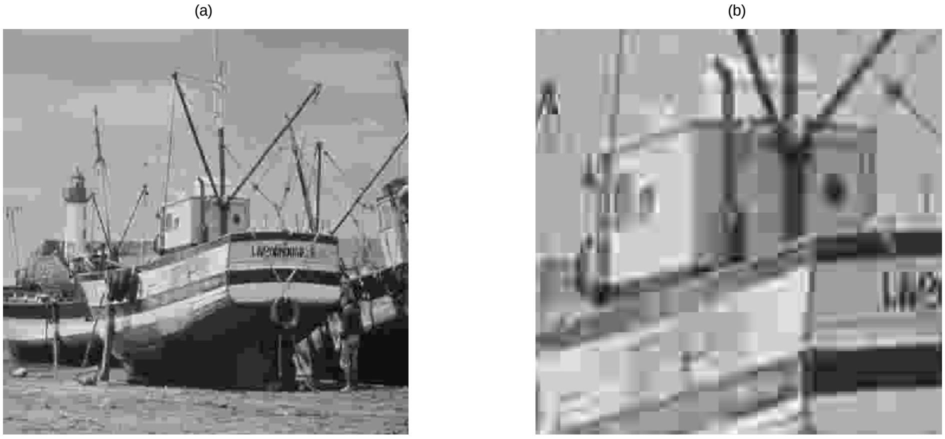


Figure 3.11: Showing the block artifacts of JPEG: (a) The image `boat' coded at low bitrate (quality 8); (b) Close up of coded image, showing the 8 x 8 blocks.

Transform coders which use a single transformation and quantization scheme, assume that images are stationary, i.e. the statistics throughout the image are uniform [55]. Unfortunately, this is not true as images contain edges and textures which have different spectral characteristics than those of smooth regions. Chen and Smith [56] investigated an adaptive scheme, where after each image block was transformed using the DCT, the resulting transformed blocks are classified based on the amount of activity, as measured by the energy level of the AC coefficients. That is, high activity blocks, such as edges or textures, will disperse energy among the high frequency (AC) coefficients, while smooth blocks will tend to concentrate energy in the DC coefficient. Classification is done using an equiprobable partition of the cumulative distribution function of the AC energies, with more bits being assigned to blocks of higher activity or AC coefficient energy and fewer bits given to blocks of lower activity [56]. Therefore, each block to be coded is classified based on activity, and then scalar quantised using the bit allocation of that class.

3.3 Subband coding

Subband coding [57], is based on decomposing a signal into a set of frequency subbands by using a filter called the analysis filter. Separating the signal into different frequency components and treating them separately can yield better coding performance. The outputs of the analysis filters (h_0 and h_1, for a special case of 2 channels) are sub-sampled to maintain critical sampling and the resulting coefficients are transmitted to the receiver. At the decoder end, the received coefficients are interpolated and a set of filters, called the synthesis filters (g_0 and g_1), operate on the received coefficients to reconstruct the signal.

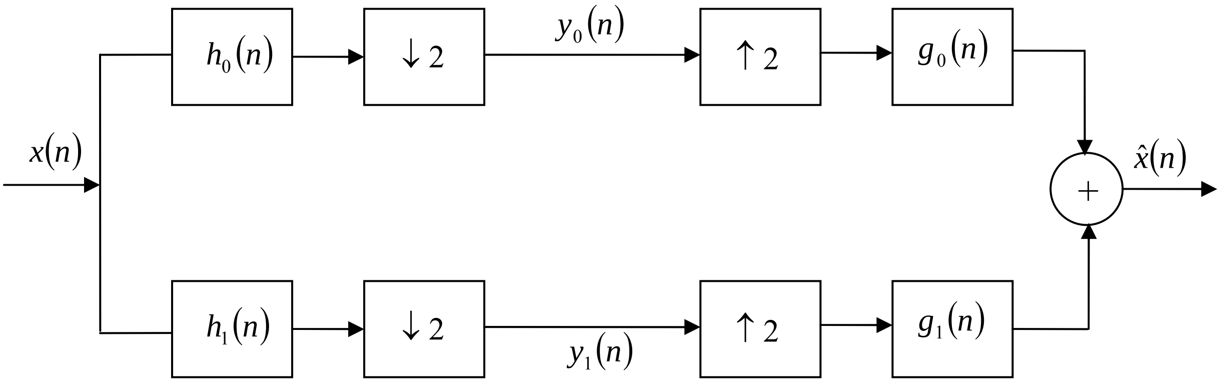


Figure 3.12: A 2-Channel Perfect Reconstruction Filter Bank.

A block diagram representing the above scenario is shown in Figure 3.12. The analysis and synthesis filters are chosen so that aliasing effects due to sub-sampling cancel out. With proper filter selection the above system is called the two-channel perfect reconstruction (PR) filter bank. If there is no quantization of the signals at the encoder and if there are no errors in the channel through which the signal is transmitted, then the reconstructed signal is an exact replica of the transmitted signal. If $X(Z)$ is the representation of the signal $x(n)$ in the z-transform domain, $H_0(z)$, $\mathrm{H}_1(z)$, $G_0(z)$ and $G_1(z)$ are the frequency responses (in the z-transform domain) of the analysis low-pass, high-pass and synthesis low-pass and high-pass filters respectively and $\hat{X}(z)$ corresponds to the reconstructed signal, $\hat{x}(n)$, then we have [58]

$$\hat{X}(z) = \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]X(z) + \frac{1}{2}\left[H_0(-z)G_0(-z) + H_1(-z)G_1(-z)\right]X(-z) \tag{3.10}$$

For perfect reconstruction

$$\hat{X}(z) = X(z) \tag{3.11}$$

which implies that the coefficients of $X(z)$ must add up to 1; or

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2 \tag{3.12}$$

Also, all the aliasing terms which contain the factor $X(-z)$ must cancel,

$$H_0(-z)G_0(-z) + H_1(-z)G_1(-z) = 0 \tag{3.13}$$

In practice it is desirable to deal with finite impulse response (FIR) filters. In order to achieve PR with FIR filters, it is required that [58]

$$H_0(z)H_1(-z) - H_1(z)H_0(-z) = 2z^{-2k-1} \tag{3.14}$$

where k is any integer. Substituting Equation 3.14 into Equations 3.12 and 3.13, we get the definition of the synthesis filters in terms of the analysis filters

$$G_0(z) = z^{2k+1} H_1(z) \tag{3.15}$$

and

$$G_1(z) = -z^{2k+1} H_0(-z) \tag{3.16}$$

From the above equations it is clear that the synthesis filters will be non-causal if the analysis filters are causal. The synthesis filters can be made causal by multiplying by z^{-2k-1}, so that Equation 3.11 becomes

$$\hat{X}(z) = z^{-2k-1} X(z) \tag{3.17}$$

describing perfect reconstruction with delay $2k+1$. Extensive research has been done to solve the Equations 3.12 and 3.13 [59].

The ideal subband filter with a vertical transition band, shown in Figure 3.13, is often called a `brick-wall' filter and can only be approximated using realisable digital filters.

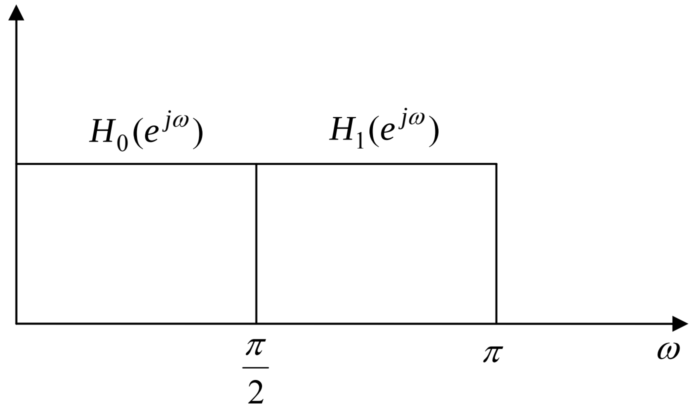


Figure 3.13: Brick-wall filter with ideal filter response.

Esteban and Galand [60] introduced a new type of subband filter that avoided the aliasing problems and these were called quadrature mirror filters (QMFs) whose ideal frequency response is shown Figure 3.14. The high pass QMF is obtained from the low pass QMF, $h(n)$, by the formula, $(-1)^n h(n)$, therefore $H_0(z) = H(z)$ and $H_1(z) = H(-z)$. Rather than attempting to approximate the ideal subband filters, QMFs were designed to cancel out the aliasing effect of the filter transition band [57]. The aliasing distortion can be shown to be equal to:

$$D = \frac{1}{2}\left[G_0(z)H(-z) + G_1(z)H(z)\right]X(-z) \tag{3.18}$$

To set the aliasing distortion to zero, the following condition must be satisfied by the QMFs:

$$G_0(z) = H(z) \tag{3.19}$$

$$G_1(z) = -H(-z) \tag{3.20}$$

The perfect reconstruction property can therefore be expressed as (assuming an even-length linear phase QMF) [61]

$$\left|H(\omega)\right|^2 + \left|H(\omega - \pi)\right|^2 = 1 \tag{3.21}$$

Figure 3.14 shows this perfect reconstruction property for the 8-tap Johnston QMFs.

A popular set of QMFs used in audio subband coding are the Johnston filters [62]. Generally, the more taps there are in the QMF, the better the separation ability and therefore efficiency [44] as it is shown in Figure 3.14. However, too many taps can lead to problems with the accumulation of errors due to coarse quantisation. As opposed to human hearing, because the human visual system is more sensitive to phase changes [63], only linear phase (symmetric) QMFs are used in subband image coding.

Vetterli extended the QMF idea to two or more dimensions, which was necessary for the subband coding of images. The idea was to use separable filters, where a two-dimensional QMF can be separated into two one dimensional filters [38].

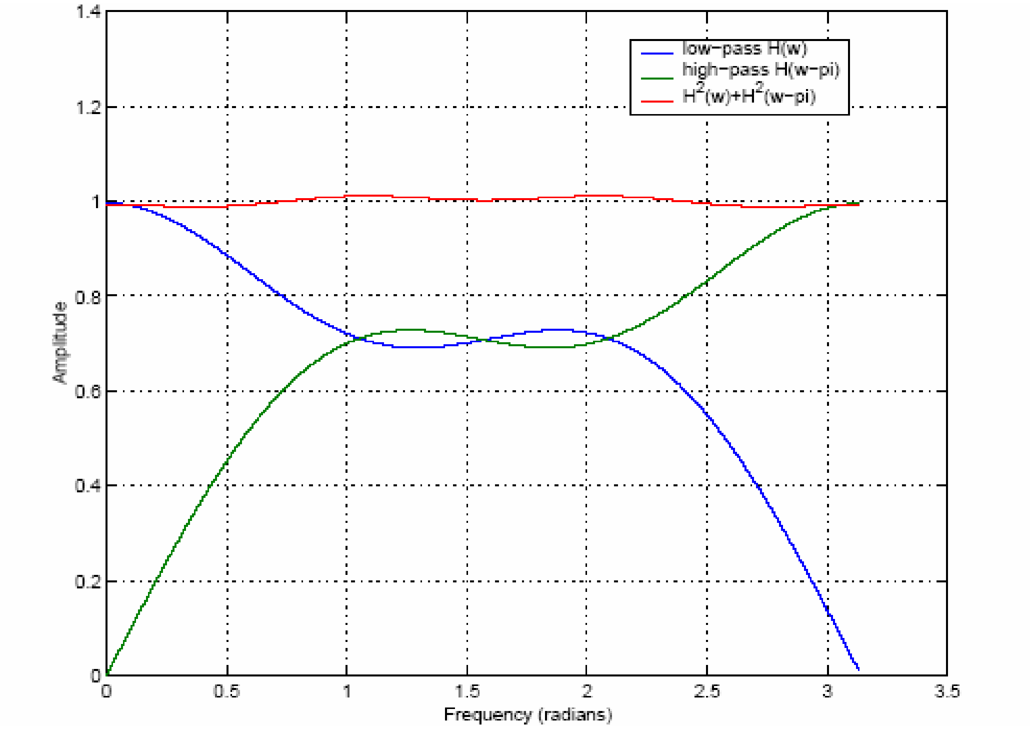
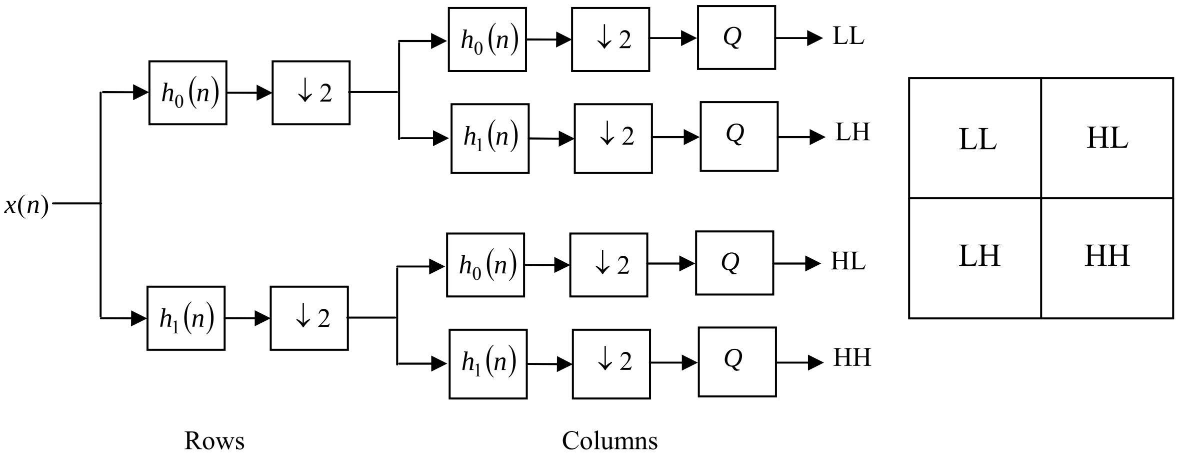


Figure 3.14: Frequency response of the Johnston QMF (8-taps).

$$h(m,n) = h_0(m)h_1(n) \tag{3.22}$$

This simplifies the implementation of two-dimensional subband coding where the row and columns can be filtered separately, similar in operation to the separable transform. Figure 3.15 shows one `level' of a subband decomposition which produces four subbands, labelled *LL* (low-low), *LH* (low-high), *HL* (high-low), and *HH* (high-high).



3.15: Two dimensional subband coding using separable filters.

3.4 Wavelet Transform Coding

With the discovery that continuous time wavelets can be generated from discrete-time filters and vice-versa [64, 65], the design of wavelets and filter banks have become connected. Wavelets can be used to generate filters that can be used for subband coding.

3.4.1 Wavelets and their properties

With the short-time Fourier transform (STFT), a signal is time windowed (for the case of images, the signals span space rather than time) into segments and the frequency spectra are determined for each segment via a discrete Fourier transform [38]. This results in a two-dimensional data representation known as the time-frequency representation. While this representation is adequate for signal analysis, it has inherent disadvantages. To obtain high resolution in the frequency domain (narrowband), the time window has to be made wider, in order to capture longer data lags, but this results in low resolution in the time domain. On the other hand, to obtain high resolution in the time domain, a shorter time window is required but this in turn degrades the frequency resolution to wideband [66]. This trade-off between time and frequency is intrinsic and was first discovered by Gabor [67], popularly known as the Gabor uncertainty principle

$$\Delta\omega\Delta t \geq \frac{1}{2} \tag{3.23}$$

Since the basis functions of the STFT are of fixed size in time and frequency, the time-frequency resolution is fixed as well [38]. Relating this to DCT-based transform image coding, because the basis functions (cosines) have fixed spatial area and frequency bandwidth, smooth and edge regions are represented by transform coefficients at the same spatial-frequency resolution. Consequently, edge information tends to result in energy being dispersed to many transform coefficients. A higher bitrate is therefore required to reconstruct the edge information accurately [66].

Wavelets allow a signal to be analysed at different scales or support widths. The wavelet basis set consists of functions with different support widths to trade-off time and frequency resolution [38]. Wavelets with wide support examine large regions of the signal and hence are suitable for low frequency content or analysing `trends' while those with short support examine small regions of the signal and hence are suitable for high frequency content or analysing `anomalies' [38]. Because wavelets often have compact support, the decomposition of a signal at a time instant consists of only those wavelets that are located in (or, translated to) that region. Other wavelets that are outside the vicinity do not contribute to the reconstruction. This contrasts to the Fourier transform, whose complex exponential basis functions have global support and exist for all time, or in the case of the STFT, exist throughout the analysis window. The property of compact support is of benefit to the image coding problem, as wavelets with short support have excellent spatial resolution and this allows edge information to be represented by sparser wavelet coefficients. Therefore, encoding edge information will not require as high a bitrate as one would need in a DCT-based transform coder.

Wavelets are a set of basis functions that are generated through dilations and translations of a single function, $\psi(t)$, called the mother wavelet [68].

$$\psi_{a,b}(t) = |a|^{\frac{-1}{2}} \psi\left(\frac{t-b}{a}\right) \tag{3.24}$$

Where a specifies the dilation factor (scale) and b the amount of translation. High frequency resolution (or, small scale) wavelets which have a narrow width, correspond to $a < 1$ while low frequency (or, large scale) wavelets which have a wider width, correspond to $a > 1$ [68]. The mother wavelet also satisfies the following property

$$\int_{-\infty}^{\infty} \psi(x) dx = 0 \tag{3.25}$$

which implies the function is oscillatory [68].

The continuous wavelet transform (CWT), $c(a,b)$, of a signal, $x(t)$, is defined as

$$c(a,b) = \int x(t) \psi_{a,b}^{*}(t) dt \tag{3.26}$$

where a and b are the scale and translation of the wavelet, respectively. As opposed to the time-frequency representation of the STFT, the CWT produces a time-scale representation [69].

It should be noted that the scale, a, and translation, b, in Equation (3.26) are continuous variables. In order to limit the number of wavelet basis functions and to avoid the generation of an awful lot of data, the scale and translations are discretised, with $c(a,b)$ becoming a set of wavelet coefficients. The resulting wavelet is termed a discrete wavelet [69]. In order for the wavelet transform to be non-redundant and easily computed, the basis functions need to be orthogonal. A dyadic discretisation of the scale and translation, $a = 2^m$ and $b = 2^m n$, is popular and results in the following functions [68]

$$\psi_{m,n}(t) = 2^{\frac{-m}{2}} \psi\left(2^{-m} t - n\right) \tag{3.27}$$

which form an orthogonal basis in $L^2(R)$, the space of square integrable functions. A wavelet series decomposition can then be performed on a continuous signal $x(t)$:

$$x(t) = \sum_{m,n} c_{m,n}(x) \psi_{m,n}(t) \tag{3.28}$$

where

$$c_{m,n}(x) = < \psi_{m,n}(t), x(t) > \tag{3.29}$$

3.4.2 Multiresolution Analysis and the Discrete Wavelet Transform

Mallat [64] introduced the concept of multiresolution analysis (MRA) using wavelets, which is similar to the Laplacian pyramid idea proposed by Burt and Adelson [70]. That is, a lower resolution approximation of an image is found, effectively via a low-pass and downsampling operation. However, the loss of information (residual), as a result of going to the coarser approximation, is represented more efficiently than in the Laplacian pyramidal coder and does not lead to an increase in data. This is the basic operation of the discrete wavelet transform (DWT).

In multiresolution analysis, one really has two functions: the mother wavelet ψ and its dilated and translated versions [68]

$$\phi_{m,n}(t) = 2^{\frac{-m}{2}} \phi\left(2^{-m} t - n\right) \tag{3.30}$$

For fixed *m*, the $\phi_{m,n}$ are orthonormal. Let V_m denote the space spanned by the $\phi_{m,n}$. These spaces then describe successive approximation spaces,

$$...V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2}... \tag{3.31}$$

each with resolution 2^m. For each *m*, the wavelets $\psi_{m,n}$ span a space W_m which is exactly the orthogonal complement in V_{m-1} of V_m. Thus the coefficients

$$c_{m,n}(x) =< \psi_{m,n}(t), x(t) >= \int \psi_{m,n}(t) x(t) dt \tag{3.32}$$

describe the information lost when going from an approximation of x with resolution 2^{m-1}, to the coarser approximation with resolution, 2^m. All this is translated into the following algorithm [68]

$$\begin{aligned} c_{m,n}(x) &= \sum_k h_1(2n-k) a_{m-1,k}(x) \\ a_{m,n}(x) &= \sum_k h_0(2n-k) a_{m-1,k}(x) \end{aligned} \tag{3.33}$$

$$\begin{aligned} h_1(n) &= (-1)^n h_0(1-n) \\ h_0(n) &= \sqrt{2} \int \phi(t-n) \phi(2t) dt \end{aligned} \tag{3.34}$$

where h_0 and h_1 are low-pass and high-pass filters respectively. If x is given in sampled form, then one can take these samples as the highest order resolution coefficients $a_{0,n}$ and describe a subband coding algorithm using the low-pass filter h_0 and the high-pass filter h_1. Because of their association with orthonormal wavelet bases, these filters give exact reconstruction

$$a_{m-1,l}(x) = \sum_n \left[h_0(2n-l)a_{m,n}(x) + h_1(2n-l)c_{m,n}(x) \right] \tag{3.35}$$

These equations form the basis of the multiscale image decomposition filter structure depicted in Figure 3.16.

The DWT described in the previous section is for one dimensional (1-D) signals. Images are 2-D and are analyzed using a separable 2-D wavelet transform. A 2-D separable transform is equivalent to two 1-D transforms in series. It is implemented as a 1-D row transform followed by a 1-D column transform on the data obtained from the row transform. Figure 3.17 shows the filter bank structure for computation of a 2-D DWT and IDWT.

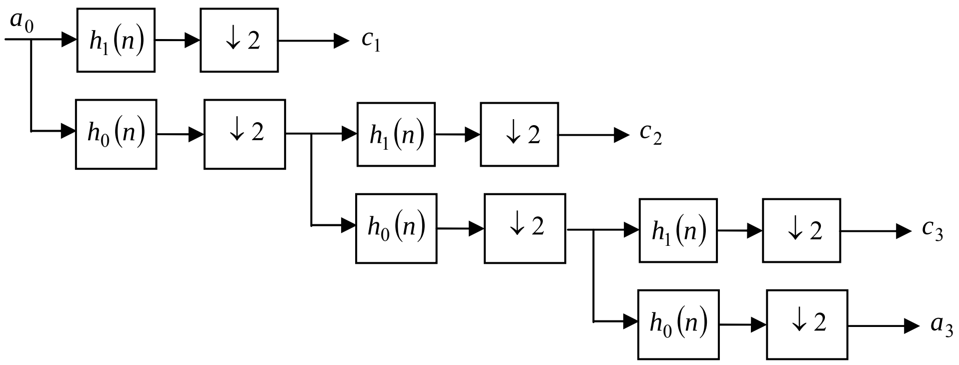


Figure 3.16: Decomposition filter bank for wavelet transforms.

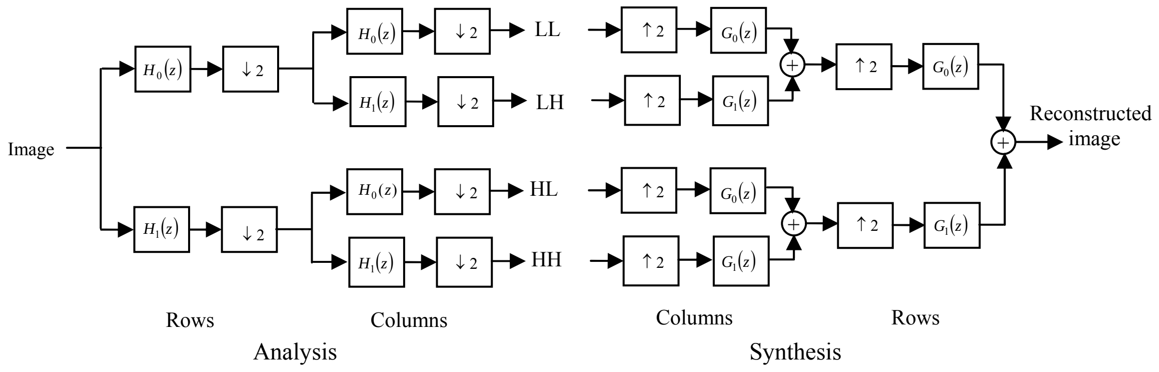


Figure 3.17: One level filter bank for computation of 2-D DWT.

The resulting subband structure for a 1-level decomposition is shown in Figure 3.18 (In the notation LL_i, LH_i etc. the first letter refers to the horizontal filtering (L:lowpass, H: highpass) at level i and the second letter to the vertical filtering of the coefficients in that subband.).

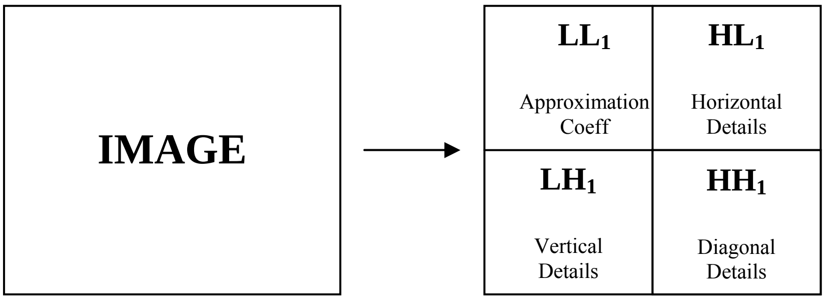


Figure 3.18 The 1-level wavelet decomposition with subband notation.

As we know, most images are smooth. It is reasonable to use smooth mother wavelet for image analysis. On the other hand, it is also desirable that the mother wavelet is symmetric so that the corresponding wavelet transform can be implemented using mirror boundary conditions that reduce boundary artifacts. Unfortunately, except for the Harr wavelet (trivial example), no wavelets are both orthogonal and symmetric [71]. To achieve the symmetric property, the orthogonality requirement could be relaxed by using a biorthonogal basis.

Biorthogonal wavelet filters define an extra set of basis functions, bringing the total to four: $\phi(t),\tilde{\phi}(t),\psi(t),\tilde{\psi}(t)$. The `tilde' functions become the synthesis basis functions while the other two form the analysis basis functions. The reconstruction becomes

$$a_{m-1,l}(x)=\sum\left[g_0(2n-l)a_{m,n}(x)+g_1(2n-l)c_{m,n}(x)\right] \tag{3.36}$$

where the reconstruction filters g_0, g_1 may differ from the analysis filters h_0, h_1. The relations between them are given by [71]

$$\begin{aligned} g_1(n) &= (-1)^n h_0(1-n) \\ h_1(n) &= (-1)^n g_0(1-n) \end{aligned} \tag{3.37}$$

The most commonly used wavelet filters in image coding and compression are the biorthogonal, so called “9/7” filters (named after the number of filter taps in the low and high frequency filters) [72].

Table 3.1 shows the filter coefficients for the biorthogonal 9/7-tap filter. The corresponding scaling and wavelet functions are plotted in Figure 3.18.

n	1	±1	±2	±3	±4
h_n	0.602949	0.602949	-0.078223	-0.016864	0.026749
$\tilde{h}_n$	0.557543	0.295636	-0.028772	-0.045636	0

Table 3.1: Coefficients of the spline variant 9/7-tap wavelet filter.

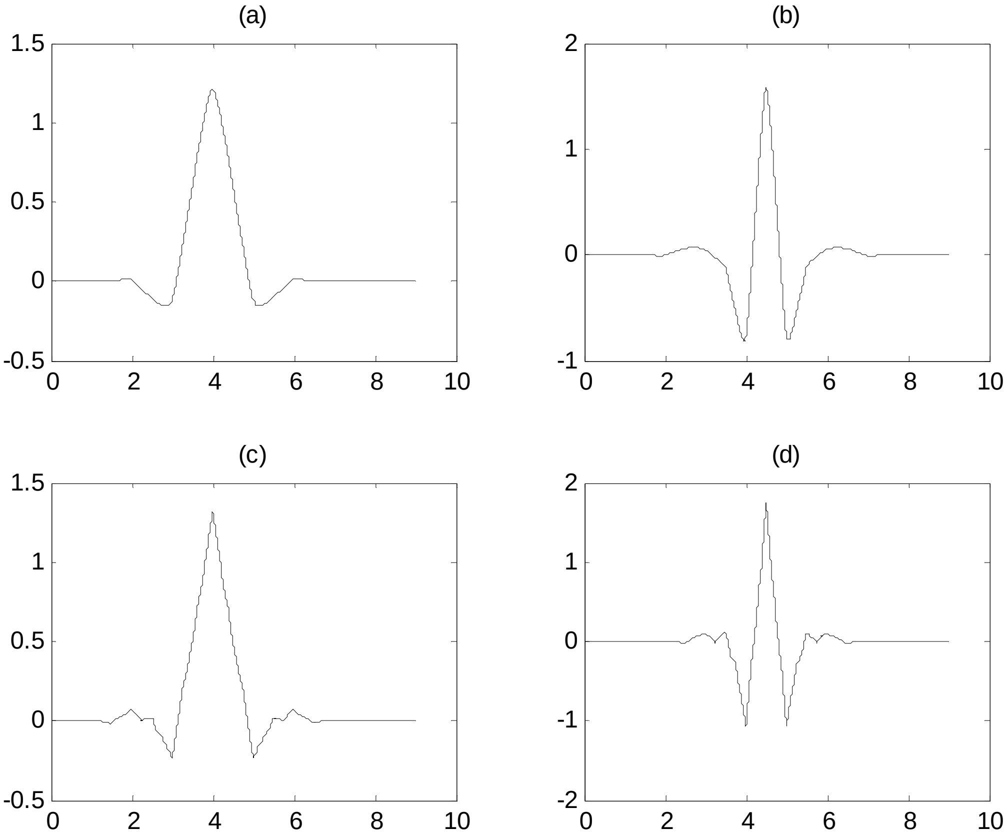
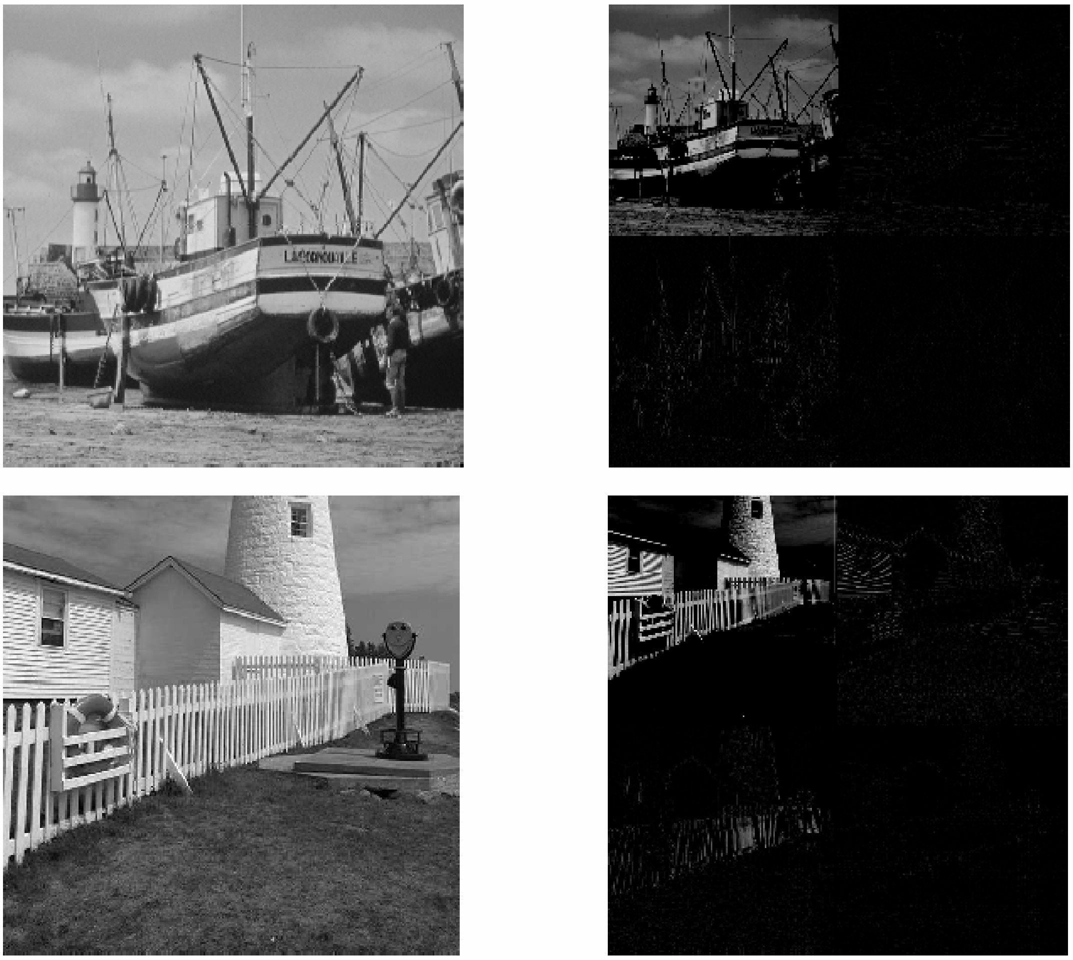


Figure 3.19 (a) analysis scaling function ϕ ; (b) analysis wavelet function ψ ; (c) reconstruction scaling function $\tilde{\phi}$; (d) reconstruction wavelet function $\tilde{\psi}$.

Figure 3.20 shows the original greyscale test images `boat', `lighthouse', and `Sandiego' as well as their wavelet transform (using the 9/7-tap wavelet filter). It can be observed that most of the energy of the image has been compacted into the LL subband, where the low frequency information or `trends' of the image have been captured by the scaling functions. The coefficients in the other subbands are very sparse except for some edge information which have been captured by the local wavelet functions. In fact, wavelets act as `singularity detectors' and it is because of the sparseness of the wavelet coefficients that makes the DWT a more efficient transform than the DCT on edge data [73]. Also, since this wavelet is biorthogonal rather than orthonormal, energy has not been conserved and the LL subband contains more energy.



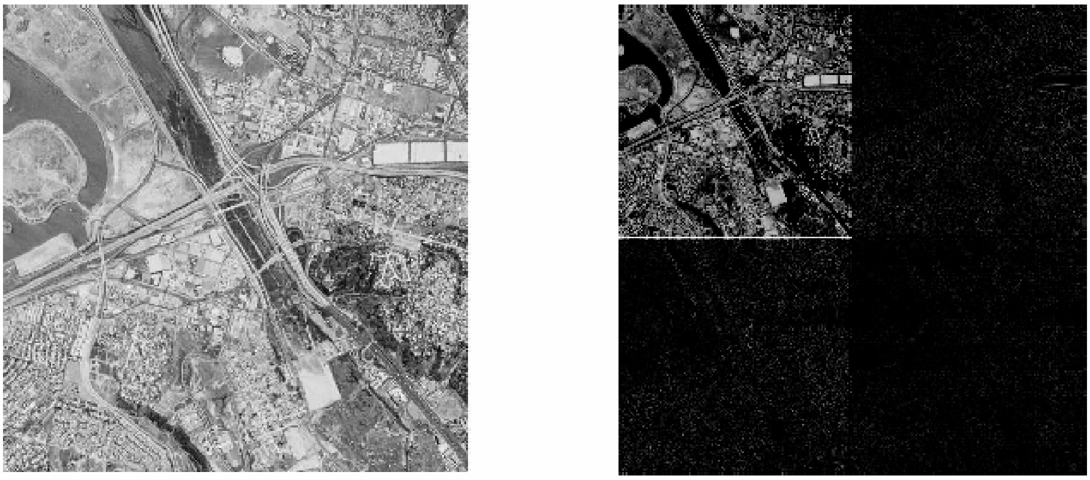


Figure 3.20: Discrete wavelet transform of greyscale test images: `Boat', `Lighthouse', `Sandiego' (using 9/7-tap biorthogonal wavelet filters).

CHAPTER 4

MULTIPLE DESCRIPTION CODING

With the rapid development of Internet applications, a large number of images are sent over the Internet. The most common way to communicate an image over the internet is to use a progressive encoding system and to transmit the coded image as a sequence of packets over a TCP connection. Suppose N packets are used to send an image. When the first packet arrives, a coarse image is got. When more and more packets arrive, the quality of this image is improved. This scheme works well if all packets are received one by one. But if a packet is lost, the whole reconstruction stops until that packet is resent and finally received. For a TCP-based network, the delay for resending a packet is often much longer than the time between the arrivals of packets. If one packet is lost, all packets after it become useless for the reconstruction of the image until the retransmitted lost packet is received.

If packet loss is unavoidable and which packet would be lost can be known in advance, there is no way of making the packet useful only if the packets before it are received. A better idea is to make all packets equally important so that a reconstruction can be made from an arbitrary combination of packets. That is exactly the idea behind multiple description (MD) coding. In MD coding, the signal is encoded into more than one description and each description is packed into one packet. Each packet can have equal importance. When all packets are received, a signal with high fidelity is reconstructed. When one or more packets are lost, the quality of the reconstructed signal will degrade gracefully. Some technique such as path diversity may be employed to make sure that at least some packets are received. Path diversity means that different packets are sent through different paths so that if some paths are blocked due to various reasons, the packets in other paths can still be received.

This chapter addresses the problem of multiple description (MD) coding, which, originally formulated as a source coding scheme that represents a source stream with multiple coded streams, can be cast as a joint source-channel coding method for a channel whose end-to-end performance includes uncorrected erasures [24, 74, 75]. This channel is encountered in a packet communication system that has effective error detection but does not have retransmission of incorrect or lost packets. In our work, each description is a single network packet, so we will use the terms "description" and "packet" interchangeably throughout the thesis.

4.1 Introduction

At the September 1979 Shannon Theory Workshop, the following question was posed by Gersho, Ozarow, Witsenhausen, Wolf, Wyner, and Ziv [11]. If an information source is described by two separate descriptions, what are the concurrent limitations on qualities of these descriptions taken separately and jointly? Known at that time in Bell Laboratories as the channel splitting problem, this came to be known as the multiple description (MD) problem in the information theory community. The primary theoretical results in this area were provided in the 1980s by the aforementioned researchers along with Ahlswede, Berger, Cover, El Gamal, and Zhang.

Figure 4.1 depicts the basic concept for the case of two descriptions, where an encoder is given a sequence of source symbols $\{X_k\}_{k=1}^{N}$ to communicate to three receivers over two independent noiseless (or error-corrected) channels. Description 1 is coded with rate R_1 and Description 2 with rate R_2. The side decoders (Decoder 1 and Decoder 2) receive each one description and the respective distortions are D_1 and D_2. The central decoder (Decoder 0) however receives both the descriptions and the corresponding distortion is D_0. For a quadratique distortion measure and real-valued signals, the distortions are given by

$$D_i = \frac{1}{N}\sum_{k=1}^{N} E\left[\left(X_k - \hat{X}_{i,k}\right)^2\right], \qquad i \in \{0,1,2\} \tag{4.1}$$

where $\hat{X}_{i,k}$ denotes the reconstruction sequence produced by decoder i.

The problem is formulated as follows. Given the rates of the two descriptions, R_1 and R_2, minimize D_0 subject to the constraints [76]

$$\begin{array}{lcl} D_1 & \leq & D_{1,\max} \\ D_2 & \leq & D_{2,\max} \end{array} \tag{4.2}$$

where $D_{1,\max}$ and $D_{2,\max}$ are given constants. If $D_1 = D_2$ and $R_1 = R_2$, the two descriptions are said to be balanced.

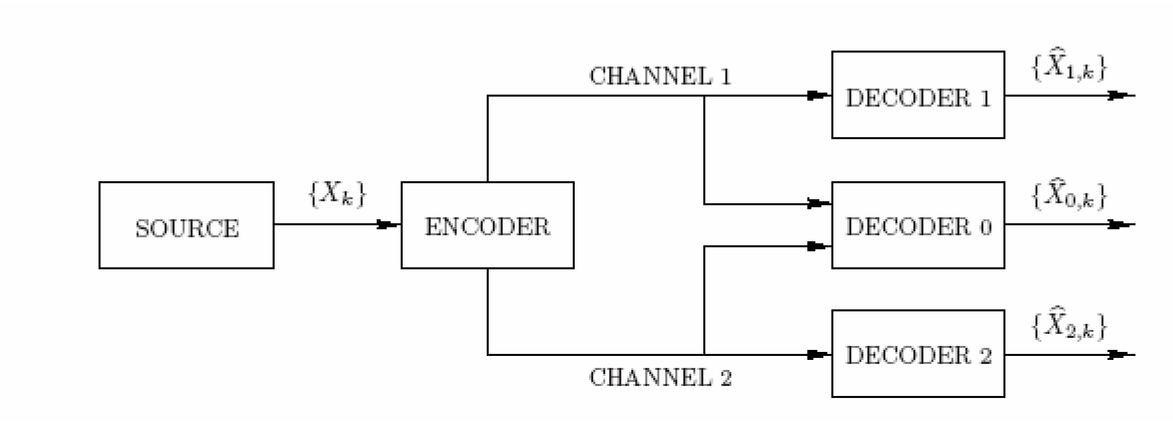


Figure 4.1: The concept of multiple description coding.

In other words, the central theoretical problem is to determine the set of achievable values (in the usual Shannon sense) for the quintuple$\left(R_1, R_2, D_0, D_1, D_2\right)$. Decoder 1 receives R_1 bits and hence cannot have distortion less than $D(R_1)$, where $D(.)$ is the distortion-rate function of the source. Making similar arguments for the other two decoders gives the following bounds on the achievable region:

$$D_0 \geq D\left(R_1 + R_2\right) \tag{4.3}$$

$$D_1 \geq D\left(R_1\right) \tag{4.4}$$

$$D_2 \geq D\left(R_2\right) \tag{4.5}$$

Achieving equalities in Equation (4.3) would imply that an optimal rate $R_1 + R_2$ description can be partitioned into optimal rate R_1 and rate R_2 descriptions. Unfortunately, this is usually not true because optimal individual descriptions at rates R_1 and R_2 are similar to each other and hence redundant when combined. Making descriptions individually good and yet sufficiently different is the fundamental tradeoff in this problem [12].

The MD problem can be generalized to more than two channels and more than three receivers. The natural extension is to M channels and $2^M - 1$ receivers, one receiver for each nonempty subset of channels. This generalization was considered by Witsenhausen [77] for the restricted case where the source has finite entropy rate and lossless communication is required when any $k < M$ of the channels are lost. Normalizing the source rate to one and assuming equal usage of each channel, each channel must accommodate a rate of $1 \big/ \left(M - k\right)$. (The rate cannot be lowered because the sum of the rates of the received channels must be at least one.) This bound is achieved by using truncated Reed–Solomon codes. The situation with three channels and seven decoders was studied by Zhang and Berger [78].

4.2 Historical Notes

The multiple description approach was originally introduced through research done at AT&T Bell labs in the 1970s to increase the reliability of the phone system through channel splitting. Some of this work was either not archived or not publicly available and much of the details have been lost [12]. One of the first real applications was suggested by Jayant and Christensen [79, 80]. Here the authors suggest a method for segmenting audio into even and odd samples (Figure 4.2) in an attempt to improve the reliability of digital audio communication.

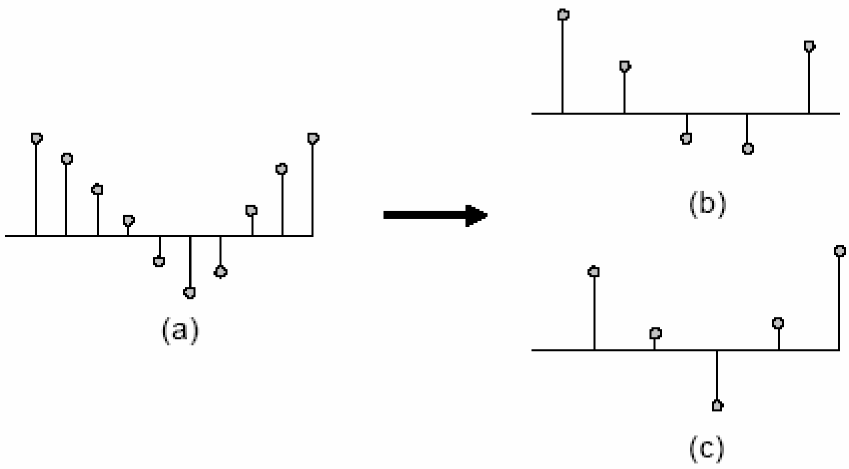


Figure 4.2: The Multiple description coding of speech using even-odd sample segmentation. Each sub-sampled audio stream is encoded independently and transmitted over the network. The temporary loss of either stream can be concealed by up-sampling the correctly received stream by interpolating the missing values [79].

Around the same time, the problem was introduced into the information theory community by Wyner, Witsenhausen, Wolf, and Ziv [77, 81] who presented preliminary results on MD coding for a binary source and Hamming distortion. This problem became very interesting from a theoretical point of view and much work has been done to analyze the problem in depth. The main focus in the information theory community has been on characterizing the multiple description region, defined as the set of all achievable quintuples$\left(R_1, R_2, D_1, D_2, D_0\right)$, under various assumptions about the statistical properties of the source. El-Gamal and Cover, for an arbitrary memoryless source and a bounded distortion measure, have done extensive work mapping out achievable rate-distortion region of pairs $\left[R_1, R_2\right]$(the rates of each description), for a given distortion vector $\left[D_0, D_1, D_2\right]$(the distortions when both descriptions arrive and when either one of them but not the other arrive) using multiple description codes for channel splitting [11]. Ozarow then showed that, if the source is Gaussian and the distortion measure is squared error, the achievable rate region of El-Gamal and Cover is, in fact, the exact MD rate/distortion region for the source [82]. Subsequently, Ahlswede [83] showed that the El Gamal–Cover region is tight in the “no excess rate sum” case (where there is equality:$D_0 = D\left(R_1 + R_2\right)$ in Equation 4.3), and Zhang and Berger [84] showed that this region is not tight when there is excess rate. The complementary situation, where Equations 4.4 and 4.5 hold with equality, is called the “no excess marginal rate” case and was also studied by Zhang and Berger [84]. It should be emphasized that, to this date, Ozarow’s result for the Gaussian source and squared error metric is the only one presenting a complete characterization of a MD rate/distortion region.

For non-Gaussian sources, no technique for precisely determining the achievable rate-distortion region is known. In [85], Zamir developed an outer bound and inner bound for MD region for a general memoryless real source with squared error distortion. These results are an extension of Shannon bounds on rate distortion function of a real source by the rate distortion function of the Gaussian source with the same variance/entropy [86].

Venkataramani, Kramer and Goyal have found bounds on the achievable performance region for MD coding with more than two descriptions [87].

An important special case of the MD problem was presented in [88] and is known as the problem successive refinement of information or multiresolution (MR). The successive refinement problem can also be described by Figure 4.1, but the interest is only in characterizing achievable$\left(R_1, R_2, D_1, D_0\right)$. In other words, no attempt is made to estimate the source from Channel 2 alone; or, Channel 1 is always present. Successive refinement was first studied by Koshelev [89, 90, 91]. The conditions for perfect successive refinement, where Equations 4.4 and 4.5 hold with equality, are described in [88]. The result follows from the tightness of the achievable region established by El Gamal and Cover [11] for the no excess rate sum case [83].

4.3 Theoretical Bounds for a Memoryless Gaussian Source

The achievable rate-distortion region is completely known only for a memoryless Gaussian source with MSE distortion [12]. This result, obtained by Ozarow [82] is summarized as follows.

Let $X_1, X_2, ..., X_n$ be a sequence of independent identically distributed (i.i.d.) unit variance Gaussian random variables. The achievable set of rates and MSE distortions is the union of points satisfying

$$D_i \geq 2^{-2R_i}, \quad i = 1,2 \tag{4.6}$$

$$D_0 \geq 2^{-2(R_1+R_2)}.\gamma\left(D_1, D_2, R_1, R_2\right) \tag{4.7}$$

where

$$\gamma = \frac{1}{1-\left(\sqrt{\left(1-D_1\right)\left(1-D_2\right)} - \sqrt{D_1 D_2 - 2^{-2\left(R_1+R_2\right)}}\right)^2} \quad \text{for} \quad D_1 + D_2 < 1 + 2^{-2(R_1+R_2)} \quad \text{and}$$

$\gamma = 1$ otherwise.

The bounds in Equation 4.6 are simply the side-channel rate-distortion bounds, a repeat of Equations 4.4 and 4.5. In the final inequality 4.7, the central distortion must exceed the rate-distortion bound by the factor γ.

The above equations are interpreted in three situations [12]:

- The side descriptions are very good individually: $D_1 = 2^{-2R_1}$ and $D_2 = 2^{-2R_2}$. Then,

$$D_0 \geq D_1 D_2 \frac{1}{1-(1-D_1)(1-D_2)} = \frac{D_1 D_2}{D_1 + D_2 - D_1 D_2}. \tag{4.8}$$

  which leads to $D_0 \geq \min(D_1, D_2)/2$.

- The central description is as good as possible: $D_0 = 2^{-2(R_1+R_2)}$, then,

$$D_1 + D_2 \geq 1 + 2^{-2(R_1+R_2)} \tag{4.9}$$

- Intermediate between the above two extreme cases: The situation is analyzed for the balanced case. Under the assumptions $R_1 = R_2 >> 1$ so $D_1 = D_2 << 1$, $\frac{1}{\gamma} = 1 - ((1-D_1) - \sqrt{D_1^2 - 2^{-4R_1}})^2 \approx 4D_1$, $D_0 \geq 2^{-4R_1}(4D_1)^{-1}$. Then,

$$D_0 D_1 \geq \frac{1}{4} 2^{-4R_1} \tag{4.10}$$

4.4 Practical Coder Designs for Multiple Descriptions

One of the first practical coder designs for multiple descriptions appears in the context of speech coding. As previously mentioned, in 1981 Jayant and Christensen [79, 80], consider MD coding of DPCM speech for combating speech coding degradation due to packet losses. Information bits corresponding to even and odd samples are placed in separate packets. If only even (odd) sample packets are lost, data contained in odd (even) packet is used to estimate the missing samples using the nearest neighbour interpolation.

4.4.1 Multiple description scalar quantization

One of the first practical MDC methods, called multiple description scalar quantization (MDSQ), was proposed by Vaishampayan in [92]. The objective of this MD coding system is to minimize the average distortion when both channels work, subject to constrains on the average side distortion when only one channel works. In this method, the encoder generates the descriptions through an Index Assignment procedure that maps the quantized transformed coefficients into a pair of indices $\left(i\,,j\,\right)$, where the first and second component are sent over its respective channels. When all the information arrives correctly to the receiver, we obtain perfect reconstruction quality in the recovery image. When there is a channel failure, an estimation procedure is applied in order to recover the lost description from the available one.

The separation of an input image into multiple descriptions is a very important issue in MD coding. There are many separating algorithms, such as polyphase transform, scalar quantization, correlating transform and DCT separation.

Figure 4.3 shows the basic high level architecture of the MDSQ system for two path connection between the source and destination. In the encoder, the system decomposes the input image X with a decorrelating transform and then applies a uniform scalar quantizer to the transformed coefficients, resulting in the quantized fieldX_q. Two descriptions of this field are generated by mapping X_q to a pair of index streams i and$j$, using the index assignment (IA) component of a MDSQ.

Basically, the IA component performs the splitting of the quantized field into two complementary and correlated descriptions. IA is an injection$I : N \rightarrow N\text{x}N$. When the scalar quantizer maps the source to a finite number of values$N$, the map I can be thought as a matrix of size $N\times N$, in which only N locations are occupied. Figure 4.4 shows an example of

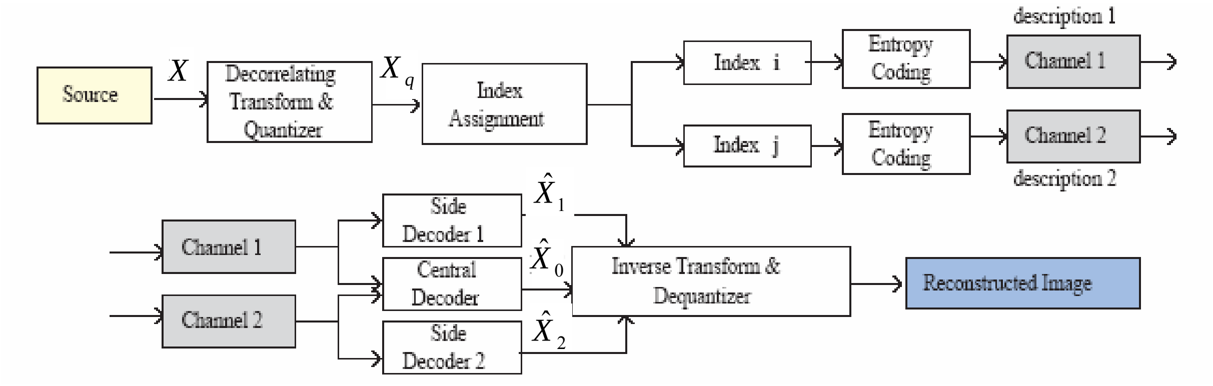
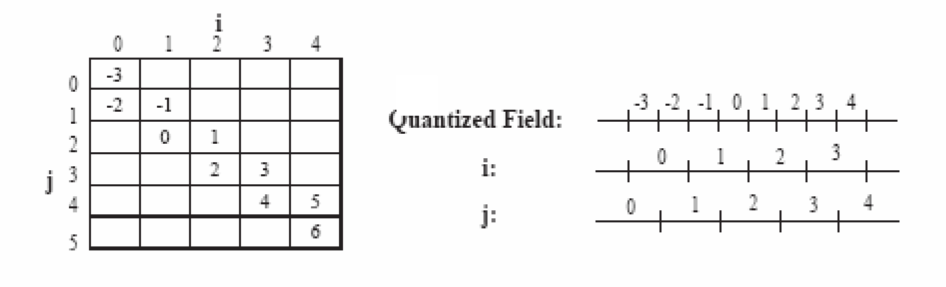


Figure 4.3: MDSQ architecture.

IA and Figure 4.5(a) depicts simple splitting procedure example for generating the descriptions. The pair of indices *i* and *j* corresponds to column and row indices of cells in the IA. The correlation between the pair of indices *i* and *j* is inversely proportional to the number of diagonals used in the matrix.



j \ i	0	1	2	3	4
0	-3				
1	-2	-1			
2		0	1		
3			2	3	
4				4	5
5					6

Figure 4.4: Example of Index Assignment Matrix [93].

When all the descriptions are received, the central decoder applies the inverse IA procedure, as it is illustrated in Figure 4.5(b). If only one description is available, estimation is applied with the help of the IA matrix. Assuming that indices *i* are available (columns in the IA matrix), we estimate the indices *j* by a mean operation. Figure 4.6 illustrates this estimation activity.

We can see that the resultant estimated output differs slightly from the initial input (Figure 4.5(a)). In MDSQ, the reconstructed image quality depends on how IA matrix is designed. Central decoder maps both descriptions to a reconstructed value$\hat{X}_0$, side decoder 1 maps description 1 (*i* indices) to $\hat{X}_1$ and side decoder 2 maps description 2 (*j* indices) to $\hat{X}_2$.

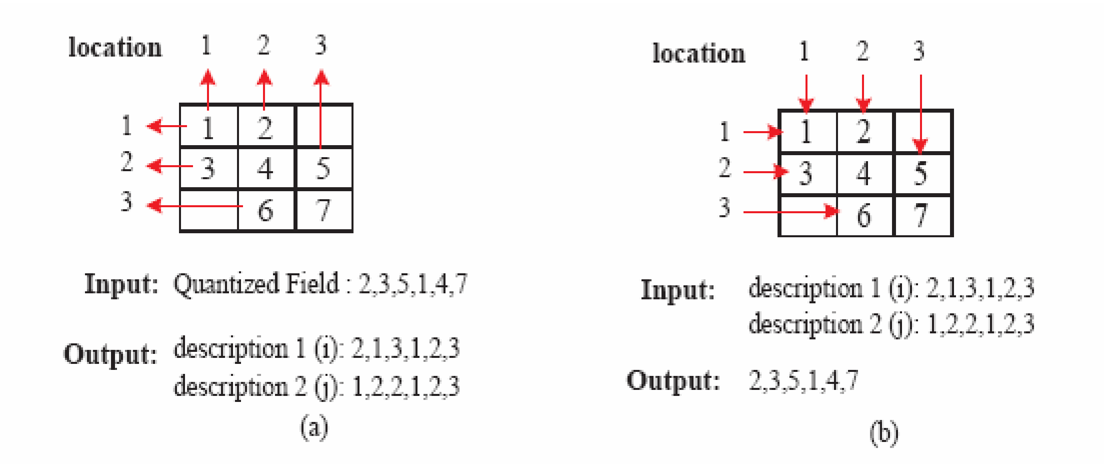

location 1 2 3
1 2
3 4 5
6 7
1
2
3
Input: Quantized Field : 2,3,5,1,4,7
Output: description 1 (i): 2,1,3,1,2,3
description 2 (j): 1,2,2,1,2,3
(a)
location 1 2 3
1 2
3 4 5
6 7
1
2
3
Input: description 1 (i): 2,1,3,1,2,3
description 2 (j): 1,2,2,1,2,3
Output: 2,3,5,1,4,7
(b)

Figure 4.5: Index Assignment Procedure (a) Generation of descriptions from quantized field and (b) Recovery of quantized field from descriptions.

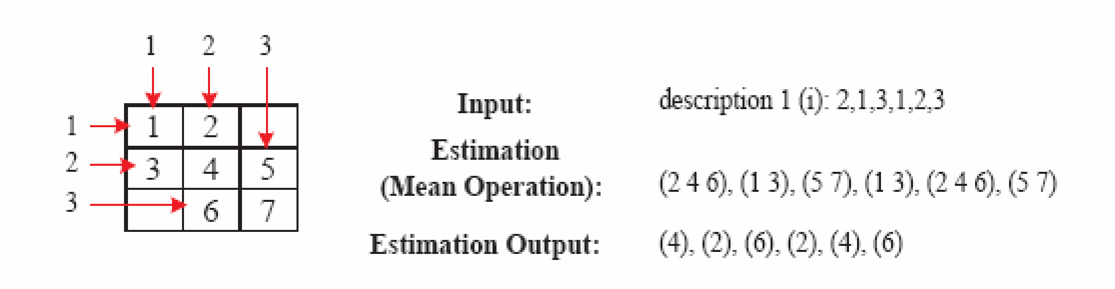

1 2 3
1 2
3 4 5
6 7
1
2
3
Input: description 1 (i): 2,1,3,1,2,3
Estimation (Mean Operation): (2 4 6), (1 3), (5 7), (1 3), (2 4 6), (5 7)
Estimation Output: (4), (2), (6), (2), (4), (6)

Figure 4.6: Index Assignment Estimation Procedure in case of one description lost.

Average distortion of central decoder, side decoder 1 and side decoder 2 are

$$
\begin{aligned}
D_0 &= E\left[\left(X - \hat{X}_0\right)^2\right] \\
D_1 &= E\left[\left(X - \hat{X}_1\right)^2\right] \\
D_2 &= E\left[\left(X - \hat{X}_2\right)^2\right]
\end{aligned}
\tag{4.11}
$$

respectively.

The same concept may be extended to more than two descriptions by using higher dimensional mappings [94].

Vaishampayan and Domaszewicz in [95] extended the work in [92] to entropy constrained quantizers. They also used variable length codes (VLCs) instead of fixed length codes. With VLCs better performances are achieved, however, VLCs are very sensitive to errors (due to synchronization problems). In [96] the authors analyse the dependencies between the variables involved in the MDSQ coding chain and design an estimation strategy making use of part of the global model of dependencies each time. By analysing the MDC system they evidence the most appropriate form of redundancy one should introduce in the context of VLC compressed streams in order to fight against de-synchronization when impaired by channel noise.

In [97] Vaishampayan and Batllo present an asymptotic analysis of MDSQ present in [92]. Specifically, expressions are derived for the average side and central distortions and for entropy when the number of quantization levels is large. In this work they compare the distortion product $D_0 D_1$ of the optimum level-constrained quantizer for a unit variance Gaussian source with the one on the converse theorem. From the converse theorem, it is shown that at high rates, for the case of balanced descriptions ($R_1 = R_2 = R$) and Gaussian sources, the distortion product $D_0 D_1$ of the entropy-constrained MD scalar quantizer (MDSQ) takes the form $\frac{1}{4}\left(\frac{\pi e}{6}\right)^2 2^{-4R}$. At the same time, the MD rate distortion bound (when put in distortion product form) becomes $\frac{1}{4}2^{-4R}$. This is an important result because it shows that for the MDSQ both the side and the central distortion attain the optimal exponential rate of decay ($D_0 \sim 2^{-2R}$,$D_1 \sim 2^{-2R}$). The only suboptimality of MDSQ at high rates is due to the use of a scalar quantizer which partitions the space into cubic regions instead of an ideal vector quantizer that would optimally partition the space into spheres. Various constructions of MD vector quantizers have been proposed [98, 99, 100] in which the MD lattice quantizers do effectively close the gap between the performance of the entropy constrained MDSQ and the MD rate-distortion bound.

4.4.2 Pairwise correlating transform

A rather different approach pioneered by Wang et al. [101, 102] and then extended by Goyal and Kovacevic [12, 13, 103] consists of applying a suitable block-wise transform to the input vector before coding to obtain the MD property. This approach is usually called MD transform coding. The basic idea is to decorrelate the vector components and then to introduce again correlation between coefficients, but in a known and controlled manner, so that erased coefficients can be statistically estimated from those received.

Let X_1 and X_2 be independent zero-mean Gaussian random variables with variances $\sigma_1^2 > \sigma_2^2$. For conventional (single-description) source coding, there would be no advantage to using a linear transform prior to quantization. Assuming high-rate entropy-coded uniform quantization, the distortion at *R* bits per sample is given by [12]

$$D_0 = \frac{\pi e}{6}\sigma_1\sigma_2 2^{-2R} \tag{4.12}$$

This is the best single-description performance that can be obtained with scalar quantization. Now suppose that the quantized versions of X_1 and X_2 are sent on channels 1 and 2, respectively, in an MD system. Since X_1 and X_2 are independent, side decoder 1 cannot estimate X_2, aside from using its mean. Thus,

$$D_1 = \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R} + \sigma_2^2 \tag{4.13}$$

and similarly

$$D_2 = \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R} + \sigma_1^2 \tag{4.14}$$

Assume for the moment that each channel is equally likely to fail. Then, instead of dealing with D_1 and D_2 separately, the average distortion will be used when one channel is lost

$$\bar{D}_1 = \frac{1}{2}(D_1 + D_2) = \frac{1}{2}\left(\sigma_1^2 + \sigma_2^2\right) + \frac{\pi e}{12}\sigma_1\sigma_2 2^{-2R} \tag{4.15}$$

$\bar{D}_1$ could be reduced if side decoder i had some information about $X_j, i \neq j$. This can be accomplished by transmitting not X_i's, but correlated transform coefficients. The simplest possibility, as proposed in [102], is to transmit quantized versions of Y_1 and Y_2 given by

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{4.16}$$

Since $\begin{bmatrix} Y_1 & Y_2 \end{bmatrix}^t$ (t stands for transposition) is obtained with an orthonormal transformation and the MSE distortion is used, the distortion in approximating the X_i 's equals the distortion in approximating the Y_i 's. The variances of Y_1 and Y_2 are both $\left(\sigma_1^2 + \sigma_2^2\right)/2$, so the central decoder performance is

$$D_0 = \frac{\pi e}{6}\left(\frac{\sigma_1^2 + \sigma_2^2}{2}\right)2^{-2R} \tag{4.17}$$

which is worse than the performance without the transform by a constant factor of

$$\gamma = \frac{\left(\sigma_1^2 + \sigma_2^2\right)/2}{\sigma_1\sigma_2} \tag{4.18}$$

Now consider the situation at side decoder 1. The distortion is approximately equal to the quantization error plus the distortion in estimating Y_2 fromY_1 . Since Y_2 and Y_1 and are jointly Gaussian, $Y_2 | Y_1 = y_1$ is Gaussian and $E[Y_2 | Y_1 = y_1]$ is a linear function of y_1. Specifically, $Y_2 | Y_1 = y_1$ has mean $(\sigma_1^2 + \sigma_2^2)^{-1} (\sigma_1^2 - \sigma_2^2) y_1$ and variance $2(\sigma_1^2 + \sigma_2^2)^{-1} \sigma_1^2 \sigma_2^2$. Thus

$$\bar{D}_1 \approx \frac{2\sigma_1^2\sigma_2^2}{(\sigma_1^2 + \sigma_2^2)} + \frac{\pi e}{12} \sigma_1 \sigma_2 2^{-2R} \tag{4.19}$$

Comparing (4.15) and (4.19), the constant term has been reduced by a factor of γ^2. By using other orthogonal transforms, intermediate tradeoffs can be obtained. In addition, the use of nonorthogonal transforms allows yet more operating points, including more extreme tradeoffs. In other words the limitations of the pairwise correlating transform method led to the work reported in [101].

4.5 Multiple Description Transform Coding

We propose to use the standard transform coding framework to realize the objective of MDTC. In conventional transform coding, the transform is used to decorrelate the input variables. Here we use a transform to introduce a controlled amount of correlation among the transformed coefficients. In other words, a block of N independent, zero-mean variables with different variances is mapped to a block of N statistically correlated transform coefficients. The transform coefficients are distributed to different packets (distributions) so in the case of packet loss, the lost coefficients can be estimated from the received coefficients.

The forward transform with quantization stepsize Δ of a source vector x, whose components are assumed to be independent, zero-mean and Gaussian, is implemented as follows [12, 13]:

1. $x = [x_1 \quad x_2 \quad . \quad . \quad x_n]^t$ (t stands for transposition) is uniformly quantized : $x_q = [x]_\Delta$ with $[.]$ denotes rounding to the nearest multiple of Δ.
2. The quantized vector $x_q = [x_{q1} \quad x_{q2} \quad . \quad . \quad x_{qn}]^t$ is transformed with a discrete transform $\hat{T}$:

$$y = \hat{T}(x_q) \tag{4.20}$$

3. The components of y are independently entropy coded.

Where $\hat{T}$ is a discrete version of a continuous transform T. The derivation of $\hat{T}$ from T is by first factoring T into a product of upper and lower triangular matrices with unit diagonals $T = T_1 T_2 ... T_k$. The discrete version of the transform is then given by [12]:

$$\hat{T}(x) = \left[T_1\left[T_2 ... \left[T_k x_q\right]_\Delta\right]_\Delta\right]_\Delta \tag{4.21}$$

The coding structure presented here is the Goyal and Kovacevic generalization [12] of the method proposed by Orchard, Wang, Vaishampayan, and Reibman [101] where they considered coding of two variables with the transform

$$T = \begin{bmatrix} 1 & \beta \\ -(2\beta)^{-1} & 1/2 \end{bmatrix} \tag{4.22}$$

approximated by

$$\hat{T}(x) = \left[\begin{bmatrix} 1 & 0 \\ -(2\beta)^{-1} & 1 \end{bmatrix}\left[\begin{bmatrix} 1 & \beta \\ 0 & 1 \end{bmatrix}[x]_\Delta\right]_\Delta\right]_\Delta \tag{4.23}$$

The analysis that follows is based on a high-rate (or fine-quantization, small quantization step Δ) assumption. In particular, the following assumptions or approximations are used:

- The scalar entropy of $\hat{T}\left([x]_\Delta\right)$ is the same as that of $[Tx]_\Delta$.
- The correlation structure of y is unaffected by the quantization; i.e.,

$$E\left[yy^t\right] = E\left[\hat{T}(x)\hat{T}(x)^t\right] = E\left[Tx(Tx)^t\right]$$

- When one or more components of y are lost, the distortion dominated by the effect of the erasure, so quantization can be ignored.

When all the components of $y = [y_1 \ \ y_2 ... y_n]^t$ are received, the reconstruction is obtained from the inverse transform. The distortion is precisely the quantization error from step 1. If some components of y are lost, they are estimated from the received components using the statistical correlation introduced by the transform $\hat{T}$. The estimate $\hat{x}$ is then generated by inverting the transform as before.

Denote the variances of the components of x by $\sigma_1^2, \sigma_2^2, ..., \sigma_n^2$ and denote the correlation matrix of x by $R_x = diag\left(\sigma_1^2, \sigma_2^2, ..., \sigma_n^2\right)$. Under fine quantization approximations, assume the correlation matrix of y is $R_y = E\left[yy^t\right] = E\left[Tx(Tx)^t\right] = TR_xT^t$. Consider $k > 0$ components of y are erased, the reconstruction procedure is as follow [12, 13]:

By renumbering the variables if necessary, assume that $\tilde{y}_r = [y_1, y_2, \ldots y_{n-k}]^t$ are received and $\tilde{y}_{nr} = [y_{n-k+1}, \ldots y_n]^t$ are lost. The vector y could be partitioned in "received" and "not received" components as $y = [\tilde{y}_r, \quad \tilde{y}_{nr}]^t$. The minimum mean square error (MSE) estimate of x given $\tilde{y}_r$ is $E[x/\tilde{y}_r]$, which has a simple closed form because x is a jointly Gaussian vector. Using the linearity of the expectation operator we have:

$$\begin{aligned} \hat{x} &= E[x/\tilde{y}_r] = E\left[T^{-1}Tx/\tilde{y}_{\mathrm{r}}\right] = T^{-1}E[Tx/\tilde{y}_r] \\ &= T^{-1}E\left[\begin{bmatrix} \tilde{y}_r \\ \tilde{y}_{nr} \end{bmatrix} \Big/ \tilde{y}_r\right] = T^{-1}\begin{bmatrix} \tilde{y}_r \\ E[\tilde{y}_{nr}/\tilde{y}_r] \end{bmatrix} \end{aligned} \tag{4.24}$$

If the correlation matrix of y is partitioned in a way compatible with the partition of y as

$$R_y = TR_xT^t = \begin{bmatrix} R_1 & B \\ B^t & R_2 \end{bmatrix} \tag{4.25}$$

with R_x and R_y denote respectively the correlation matrices of the vectors x and y.

Then it can be shown that $\tilde{y}_{nr}/\tilde{y}_r$ is Gaussian with mean $B^t R_1^{-1} \tilde{y}_r$ and correlation matrix $A = R_2 - B^t R_1^{-1} B$. Thus $E[\tilde{y}_{nr}/\tilde{y}_r] = B^t R_1^{-1} \tilde{y}_r$ and the reconstruction is

$$\hat{x} = T^{-1}\begin{bmatrix} \tilde{y}_r \\ B^t R_1^{-1} \tilde{y}_r \end{bmatrix} \tag{4.26}$$

4.5.1 Transform optimization

The choice of the transform T determines the performance of the system. This section develops the relationships between the transform, rates, and distortions necessary to design T. Estimating the rate is straightforward. Since the quantization is assumed to fine, y_i is approximately the same as $\left[T(x)_i\right]_\Delta$, i.e., a uniformly quantized Gaussian random variable. If y_i is treated as a Gaussian random variable with power $\sigma_{y_i}^2 = (R_y)_{ii}$ quantized with bin width Δ, the entropy of the quantized coefficient is approximately [104, Ch. 9]

$$H(y_i) \approx \frac{1}{2}\log_2 2\pi e\sigma_{y_i}^2 + k_\Delta \tag{4.27}$$

where $k_\Delta = \frac{1}{2}\log_2 2\pi e - \log_2 \Delta$, all logarithms are base-two. Notice that k_Δ depends only on Δ. Thus, the rate per component is estimated as

$$R = \frac{1}{n}\sum_{i=1}^{n} H\left(y_i\right) = k_\Delta + \frac{1}{2n}\log_2 \prod_{i=1}^{n} \sigma_{y_i}^2 \tag{4.28}$$

The minimum rate occurs when $\prod_{i=1}^{n} \sigma_{y_1}^2 = \prod_{i=1}^{n} \sigma_i^2$ and at this rate the components of y are uncorrelated. Interestingly $T = I$ is not the only transform which achieves the minimum rate. In fact, an arbitrary split of the total rate among the different components of y is possible. This justifies the use of a total rate constraint in the following analyses. However, the case where the rates sent across each channel are equal is particularly considered.

Concerning the distortion, first consider the average distortion due only to quantization. Since the quantization noise is approximately uniform, this distortion is $\Delta^2 / 12$ for each component. Thus the distortion per component when no components are erased is given by

$$\frac{\Delta^2}{12} \tag{4.29}$$

and is independent of T.

For the case when $k > 0$ components are lost, the distortion computation was practically achieved in the development of Equation 4.26. Let $\eta = \tilde{y}_{nr} - E\left[\tilde{y}_{nr} / \tilde{y}_r\right]$, which is Gaussian with zero mean and correlation matrix $A = R_2 - B^t R_1^{-1} B$. η is the error in predicting $\tilde{y}_{nr}$ from $\tilde{y}_r$ and hence is the error caused by the lost coefficients. However, because of the use of a nonorthogonal transform, we must return to the original coordinates using T^{-1} to compute the distortion. Substituting $\tilde{y}_{nr} - \eta$ for $E\left[\tilde{y}_{nr} / \tilde{y}_r\right]$ in Equation 4.24 gives

$\hat{x} = T^{-1}\begin{bmatrix} \tilde{y}_r \\ \tilde{y}_{nr} + \eta \end{bmatrix} = x + T^{-1}\begin{bmatrix} 0 \\ \eta \end{bmatrix}$, so $\left\|x - \hat{x}\right\|^2 = \left\|T^{-1}\begin{bmatrix} 0 \\ \eta \end{bmatrix}\right\|^2 = \eta^t U^t U \eta$ where U is the last l columns of T^{-1}. Finally,

$$E\left[\left\|x - \hat{x}\right\|^2\right] = \sum_{i=1}^{l}\sum_{j=1}^{l}\left(U^t U\right)_{ij} A_{ij} \tag{4.30}$$

The distortion with l erasures is denoted by D_l. To determine D_l, Equation (4.30) is averaged over the $\binom{n}{l}$ possible erasures of l components, weighted by their probabilities if necessary. The final distortion is a weighted sum of the distortions incurred with different numbers of channels available

$$\bar{D} = \sum_{l=0}^{n} \alpha_l D_l \tag{4.31}$$

For the case each channel has an outage probability of p and the channel outages are independent, the overall expected MSE is $\bar{D}$ with $\alpha_l = \binom{n}{l} p^l (1-p)^{n-l}$. The goal is to numerically determine transforms that minimize $\bar{D}$ for a given rate R.

4.5.2 Sending two variables

Here we will apply the analysis of the previous section to find the best transforms for sending $n=2$ variables (packets or descriptions). The case where the two descriptions are of equal importance, i.e. they have the same probability to be lost is considered. Suppose the probabilities of the combinations of packet states are given by the following Table

		Packet 1	
		lost	received
Packet 2	lost	$1-p_0-2p$	p
	received	p	p_0

Table 4.1: Probabilities of system states in optimal transform determination for MDTC for the case of two descriptions.

Let $T = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, normalized so that $\det T = 1$. Then $T^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ and

$$R_y = TR_xT^t = \begin{bmatrix} a^2\sigma_1^2 + b^2\sigma_2^2 & ac\sigma_1^2 + bd\sigma_2^2 \\ ac\sigma_1^2 + bd\sigma_2^2 & c^2\sigma_1^2 + d^2\sigma_2^2 \end{bmatrix} = \begin{bmatrix} R_1 & B \\ B^t & R_2 \end{bmatrix} \tag{4.32}$$

Using Equation 4.28, the rate per component is given by

$$R = k_\Delta + \frac{1}{4}\log_2 \left(R_y\right)_{11}\left(R_y\right)_{22} = k_\Delta + \frac{1}{4}\log_2\left(a^2\sigma_1^2 + b^2\sigma_2^2\right)\left(c^2\sigma_1^2 + d^2\sigma_2^2\right) \tag{4.33}$$

Minimizing Equation 4.33 over transforms with determinant one gives a minimum possible rate of $R^* = k_\Delta + \frac{1}{2}\log_2 \sigma_1\sigma_2$ [13]. $\rho = R - R^*$ is referred to as the redundancy which is the additional bitrate required for an MDTC coder to potentially reduce the distortion in the case of erasures

$$\rho = \frac{1}{4}\log_2 \frac{\left(a^2\sigma_1^2 + b^2\sigma_2^2\right)\left(c^2\sigma_1^2 + d^2\sigma_2^2\right)}{\sigma_1^2\sigma_2^2} \tag{4.34}$$

In order to evaluate the overall average distortion, a weighted average of the distortions for each of the four possible cases of the two descriptions is formed. If both descriptions are received, the distortion due to quantization only is $D_{1,1} = \Delta^2/12$. If neither description is received, the distortion is $D_{0,0} = \left(\sigma_1^2 + \sigma_2^2\right)/2$. The remaining two situations where one description is received and the other is lost require the application of the results of the previous section. Let $D_{1,0}$, the MSE distortion when y_1 (description1) is received but y_2 (description2) is lost. Substituting in Equation 4.30 with $\left(U^t U\right)_{11} = \begin{bmatrix} -b & a \end{bmatrix} \begin{bmatrix} -b \\ a \end{bmatrix} = a^2 + b^2$ $A_{11} = \left(\left(R_y\right)_{22} - \frac{\left(R_y\right)_{12}^2}{\left(R_y\right)_{11}} \right)$, and $\det T = ad - bc = 1$ we get

$$D_{1,0} = \frac{1}{2} E\left[\|x - \hat{x}\|^2 / y_1 received\ and\ y_2 lost \right] = \frac{1}{2}\left(a^2 + b^2\right) \frac{\sigma_1^2 \sigma_2^2}{a^2 \sigma_1^2 + b^2 \sigma_2^2} \tag{4.35}$$

Similarly, the distortion when $description1$ is lost but $description2$ is received is

$$D_{0,1} = \frac{1}{2}\left(c^2 + d^2\right) \frac{\sigma_1^2 \sigma_2^2}{c^2 \sigma_1^2 + d^2 \sigma_2^2} \tag{4.36}$$

The overall average distortion is

$$\begin{aligned} \bar{D} &= p_0 D_{1,1} + p\left(D_{1,0} + D_{0,1}\right) + \left(1 - p_0 - 2p\right) D_{0,0} \\ &= \left[p_0 \frac{\Delta^2}{12} + \left(1 - p_0 - 2p\right) \frac{\left(\sigma_1^2 + \sigma_2^2\right)}{2} \right] + p \underbrace{\left(D_{1,0} + D_{0,1}\right)}_{\bar{D}'} \end{aligned} \tag{4.37}$$

Where the first bracketed term is independent of T. Thus the optimization problem is to minimize $\bar{D}'$ for a given redundancy ρ. After eliminating d through $d = \left(1 + bc\right)/a$ and solving Equation 4.34 with the assumption $\sigma_1 > \sigma_2$, the optimal transform will satisfy [12]

$$a = \frac{1}{2c} \frac{\sigma_2}{\sigma_1} \left(\sqrt{2^{4\rho} - 1} + \sqrt{2^{4\rho} - 1 - 4bc\left(bc + 1\right)} \right) \tag{4.38}$$

When this value of a is used, $\bar{D}'$ depends only on the product bc, not on the individual values of b and c. The optimal value of bc is $(bc)_{optimal} = -1/2$ since we are considering the situation of equal packet lost probabilities. Thus, the optimal set of transforms is described by

$$\begin{array}{lllllll} \mathrm{a} & = & 0(\mathrm{arbitrary}) & & \mathrm{c} & = & -1/2\mathrm{b} \\ \mathrm{b} & = & \pm\left(2^{2\rho} - \sqrt{2^{4\rho} - 1}\right)\sigma_1 \mathrm{a}/\sigma_2 & & \mathrm{d} & = & 1/2\mathrm{a} \end{array} \tag{4.39}$$

and using a transform from this set gives

$$D_1 = D_{1,0} = D_{0,1} = \frac{1}{2}\sigma_2^2 + \frac{\sigma_1^2 - \sigma_2^2}{4.2^{2\rho}\left(2^{2\rho} + \sqrt{2^{4\rho} - 1}\right)} \tag{4.40}$$

In [22] it is suggested to use transforms of the form $\begin{bmatrix} 1 & b \\ -1/(2b) & 1/2 \end{bmatrix}$ which, in fact, lie in the optimal set of transforms described by Equation 4.39. The transforms of [22] do not give descriptions with equal rate (or, equivalently, power). Using Equation 4.27, the entropies (rates) of two transformed quantized (with quantization step Δ) components $\begin{bmatrix} y_1 & y_2 \end{bmatrix}^t = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x_1 & x_2 \end{bmatrix}^t$ are $R_1 = H(y_1) \approx \frac{1}{2}\log_2\left(a^2\sigma_1^2 + b^2\sigma_2^2\right) + k_\Delta$, $R_2 = H(y_2) \approx \frac{1}{2}\log_2\left(c^2\sigma_1^2 + d^2\sigma_2^2\right) + k_\Delta$. Getting balanced rates, i.e. $R_1 = R_2$ is equivalent to requiring $|a| = |c|$ and $|b| = |d|$, and yields

$$a = \pm\sqrt{\frac{1}{2}\frac{\sigma_2}{\sigma_1}\left(2^{2\rho} + \sqrt{2^{4\rho} - 1}\right)} \tag{4.41}$$

$$b = \pm\frac{1}{2a} = \pm\sqrt{\frac{1}{2}\frac{\sigma_1}{\sigma_2}\left(2^{2\rho} - \sqrt{2^{4\rho} - 1}\right)} \tag{4.42}$$

These balanced-rate transforms will be used in the applications sections. The notation

$$T_\alpha = \begin{bmatrix} \alpha & 1/2\alpha \\ -\alpha & 1/2\alpha \end{bmatrix} \tag{4.43}$$

will be used (where α is equal to a) . When there is no lost, the reconstruction uses

$$T_\alpha^{-1} = \begin{bmatrix} 1/2\alpha & -1/2\alpha \\ \alpha & \alpha \end{bmatrix} \tag{4.44}$$

where α is a parameter determined from the redundancy ρ to be introduced by the transform and the variances of the two components.

4.5.3 Sending four variables and more

For sending any even number of variables over two channels, Orchard et al. [101] have suggested the following: form pairs of variables add correlation within each pair and send one variable from each pair across each channel. A necessary condition for optimality is that all the pairs are operating at the same distortion-redundancy slope. IfT_α is used to transform variables with variancesσ_1^2 and σ_2^2 andT_β is used to transform variables with variancesσ_3^2 and σ_4^2, then the equal-slope condition implies

$$\beta^4 = \frac{\gamma\left(16\alpha^8\sigma_1^4 - \sigma_2^4\right) + \sqrt{\gamma^2\left(16\alpha^8\sigma_1^4 - \sigma_2^4\right)^2 + 64\alpha^8\sigma_3^4\sigma_4^4}}{32\alpha^4\sigma_3^4} \tag{4.45}$$

where

$$\gamma = \frac{\sigma_3^2\sigma_4^2\left(\sigma_3^2 - \sigma_4^2\right)}{\sigma_1^2\sigma_2^2\left(\sigma_1^2 - \sigma_2^2\right)} \tag{4.46}$$

4.5.4 Cascade structure

The transform given by Equation 4.43 is used to build larger transforms as in Figure 4.7 which illustrates the case of sending four components (descriptions). This cascade structure proposed in [12, 13] will be used in the application section for transmitting still images using four packets (descriptions).

Using this cascade structure is equivalent to using a transform of the form

$$\begin{aligned} T &= \begin{bmatrix} T_\gamma & 0 \\ 0 & T_\gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_\alpha & 0 \\ 0 & T_\beta \end{bmatrix} \\ &= \begin{bmatrix} \alpha\gamma & \gamma/2\alpha & \beta/2\gamma & 1/4\beta\gamma \\ -\alpha\gamma & -\gamma/2\alpha & \beta/2\gamma & 1/4\beta\gamma \\ -\alpha\gamma & \gamma/2\alpha & -\beta/2\gamma & 1/4\beta\gamma \\ \alpha\gamma & -\gamma/2\alpha & -\beta/2\gamma & 1/4\beta\gamma \end{bmatrix} \end{aligned} \tag{4.47}$$

The coder implementation is completed by designing the transform$T$ through numerical optimization of the parameters α, β andγ given the total redundancy and the variances of the four components (descriptions).

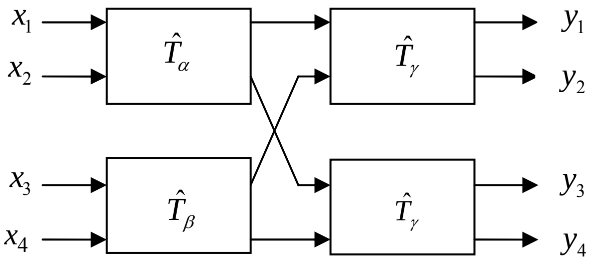


Figure 4.7: Cascade structure for MDTC coding of ıour variables

4.6 Application to Image Coding

As already mentioned, the most common way to communicate an image over the Internet is to use a progressive encoding system and to transmit the coded image as a sequence of packets over a TCP connection. When there are no packet losses, the receiver can reconstruct the image as the packets arrive; but when there is a packet loss, there is a large period of latency while the transmitter determines that the packet must be retransmitted and then retransmits the packet. The latency is due to the fact that the application at the receiving end uses the packets only after they have been put in the proper sequence. The problem is more acute if there are stringent delay requirements, for example, for fast browsing or for streaming video. In this case retransmission is not just undesirable but impossible. To combat this latency problem, it is desirable to have a communication system that is robust to arbitrarily placed packet erasures and that can reconstruct an image progressively from packets received in any order. The MDTC method described earlier seems suitable for this task.

4.6.1 DCT-based MDTC image coding

In this section, the transmission of still images using the concept of MDC coding is considered. First, we have deeply studied the Multiple Description Transform Coding (MDTC) scheme suggested by Goyal et al. in [12, 13] and its application to image coding for the case of four descriptions, then we have proposed a different way in forming the descriptions. It is shown that our employed technique leads to an improvement in performance in terms of rate/distortion [105]. In addition, using the proposed scheme, the DC coefficients (which are very important in image coding/transmission) obtained after the application of DCT need not be communicated reliably by some other means as suggested by Goyal et al.

A typical MDTC DCT-based image coder is illustrated in Figure 4.8 where the dashed block highlights where our contribution resides.

The MDTC method is designed to operate on source vectors with uncorrelated components. Such condition is obtained by forming vectors from DCT components. The coding process is implemented as follow:

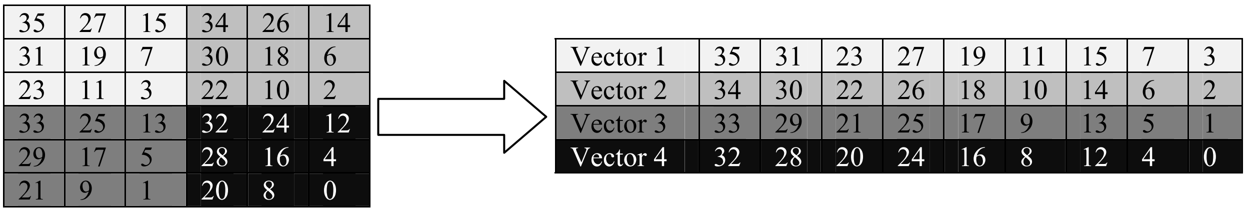
1. The source image is transformed by an 8x8 DCT transformation.
2. The DCT coefficients are uniformly quantized.
3. The quantized DCT coefficients are split into 4 vectors (descriptions).

4. Correlating transform is applied to the 4 vectors.
5. Entropy coding is applied to each vector.

In step 3, we have applied two techniques to create the four descriptions, and for each technique we've considered two cases concerning the transmission of the DC coefficients:

- **Technique1/case1**: used in [12, 13], where vectors are formed from quantized DCT coefficients separated to the maximum in frequency and space (The spatial separation is maximized, *i.e.*, for 512x512 images, the samples that are grouped together are spaced by 256 pixels horizontally and/or vertically) with the DC coefficients assumed to be communicated reliably by some other means. This method is referred to as $MDTC\ /Tec1_{DC}$.
- **Technique1/case2:** as in technique1/case1 but here the DC coefficients are assumed to be transmitted along with the four packets of data. This method is referred to as $MDTC\ /Tec1_{\overline{DC}}$.

The following example illustrates the procedure used in this technique to form the four vectors:

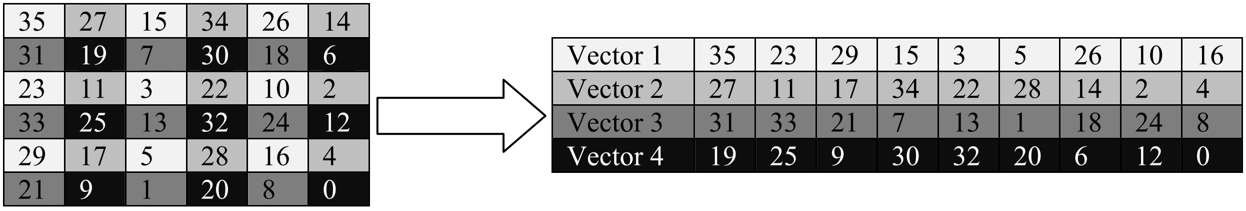


35	27	15	34	26	14
31	19	7	30	18	6
23	11	3	22	10	2
33	25	13	32	24	12
29	17	5	28	16	4
21	9	1	20	8	0

Vector 1	35	31	23	27	19	11	15	7	3
Vector 2	34	30	22	26	18	10	14	6	2
Vector 3	33	29	21	25	17	9	13	5	1
Vector 4	32	28	20	24	16	8	12	4	0

Figure 4.9: Technique 1: description forming from 2D-DCT matrix with block size of 6.

- **Technique2/case1:** quantized DCT coefficients at (odd row, odd column) are assigned to description 1; those at (odd row, even column) are assigned to description 2; those at (even row, odd column) are assigned to description 3; and those at (even row, even column) are assigned to description 4. The DC coefficients are assumed to be communicated reliably by some other means. This technique is referred to as $MDTC\ /Tec2_{DC}$.
- **Technique2/case2:** akin to technique2/case1, but the DC coefficients are transmitted along with the four packets of data. This technique is referred to as $MDTC\ /Tec2_{\overline{DC}}$. Figure 4.10 illustrates the procedure used in this technique to form the four vectors.



35	27	15	34	26	14
31	19	7	30	18	6
23	11	3	22	10	2
33	25	13	32	24	12
29	17	5	28	16	4
21	9	1	20	8	0

Vector 1	35	23	29	15	3	5	26	10	16
Vector 2	27	11	17	34	22	28	14	2	4
Vector 3	31	33	21	7	13	1	18	24	8
Vector 4	19	25	9	30	32	20	6	12	0

Figure 4.10: Technique 2: description forming from 2D-DCT matrix with block size of 6.

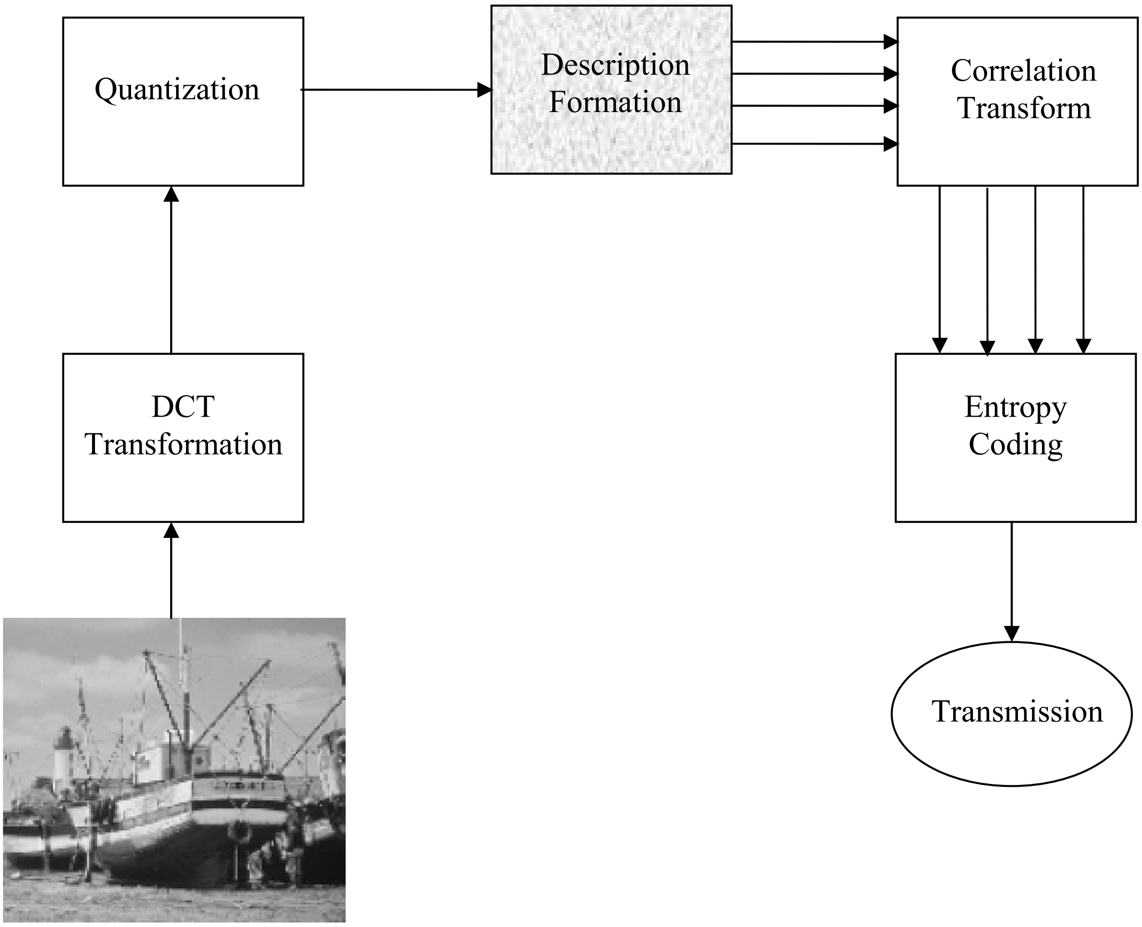


Figure 4.8: Four descriptions MDTC DCT-based image coder.

The performance is evaluated for 9 ***512x512*** grayscale images that can be grouped into three image types: three low frequency (LF) (Lena, Boat, Goldhill), three medium frequency (MF) (Lighthouse, Nitf7, House), three high frequency (HF) (Satellite, Mandrill, Sandiego) images. The frequency type groups are based on the percentage of total image energy (96% - 100% LF, 92% - 96% MF and $\leq$92% HF) in the LL subband obtained after decomposition using the 1-level *biorthogonal* $B_{9/7}$ wavelet transform [106]. The distribution of energy for the nine images is given in Table 4.2.

		Subbands			
Image type	Image	LL	LH	HL	HH
Low frequency	Lena	99.2500	0.4674	0.1863	0.0938
	Goldhill	98.6664	0.6946	0.5106	0.1284
	Boat	98.6835	0.9904	0.2452	0.0809
Medium frequency	Lighthouse	95.6545	2.6774	1.5854	0.0827
	House	94.4307	2.9444	2.4701	0.1548
	Nitf7	93.2428	3.4490	2.6661	0.6421
High frequency	Baboon	84.2130	3.5517	8.7502	3.4852
	Satellite	89.7875	5.5814	3.9542	0.6769
	Sandiego	90.1828	5.0479	3.9718	0.7975

Table 4.2: Percentage energy distribution after 1-level of decomposition using the $B_{9/7}$ wavelet transform.

Redundancy of *0.1 bit/sample* is evenly allocated to the four descriptions, and the bit rate is estimated by sample scalar entropies.

Simulation results for the nine test images using the four coders ($MDTC\,/Tec1_{DC}$, $MDTC\,/Tec1_{\overline{DC}}$, $MDTC\,/Tec2_{DC}$, $MDTC\,/Tec2_{\overline{DC}}$) are given in Figures 4.11-4.19. In all figures, the average **PSNR** is reported as a function of the bit rate for the case of one packet dropped. For comparison, the situation of no packet lost (using the two techniques) is also reported in all the figures. From these curves it can be noticed that:

- In the first situation (i.e. the DC coefficients are assumed to be communicated reliably by some other means), the technique$MDTC\,/Tec2_{DC}$ mostly performs better than $MDTC\,/Tec1_{DC}$ technique for low and medium frequency test images (Figures 4.11 (a) – 4.16 (a)), and both technique performances are practically similar for the high frequency test images (Figures 4.17 (a) – 4.19 (a)). The average performance gain amounts to nearly 2.66 dB, 3.16 dB and 0.12 dB for respectively low frequency, medium frequency and high frequency test images.
- In the second situation (i.e. the DC coefficients are transmitted along with the four packets of data.), our proposed method $MDTC\,/Tec2_{\overline{DC}}$ outperforms $MDTC\,/Tec1_{\overline{DC}}$ coder for mainly all the test images. The average performance gain amounts to nearly 7.78 dB, 6.82 dB and 2.83 dB for respectively low frequency, medium frequency and high frequency test images.

To illustrate further the performances brought by our proposed technique for erasure channels, we have measured the PSNR of the restored images in the cases of 1, 2 and 3 packets lost at a bit rate of 2 bits/sample. The corresponding results are reported in Tables 4.3-4.8. We clearly see from the Tables that using the suggested method results in an improvement in the reconstructed image quality when compared to Goyal et al. method.

For a qualitative comparison, the subjective qualities of the nine test images for different levels of reconstruction with the four coders are depicted in Figures 4.20–4.55 where Figures (a), (b), (c) and (d) illustrate respectively the situations of 0, 1, 2, and 3 packets lost. It is worth noticing that our proposed method is more robust than the other method especially when the DC coefficients are not transmitted reliably with some other means. Moreover, in the case of one packet lost (Figures (b)), which is the most probable to happen, our proposed method shows a considerable improvement in the image reconstruction quality with respect to Goyal et al. technique.

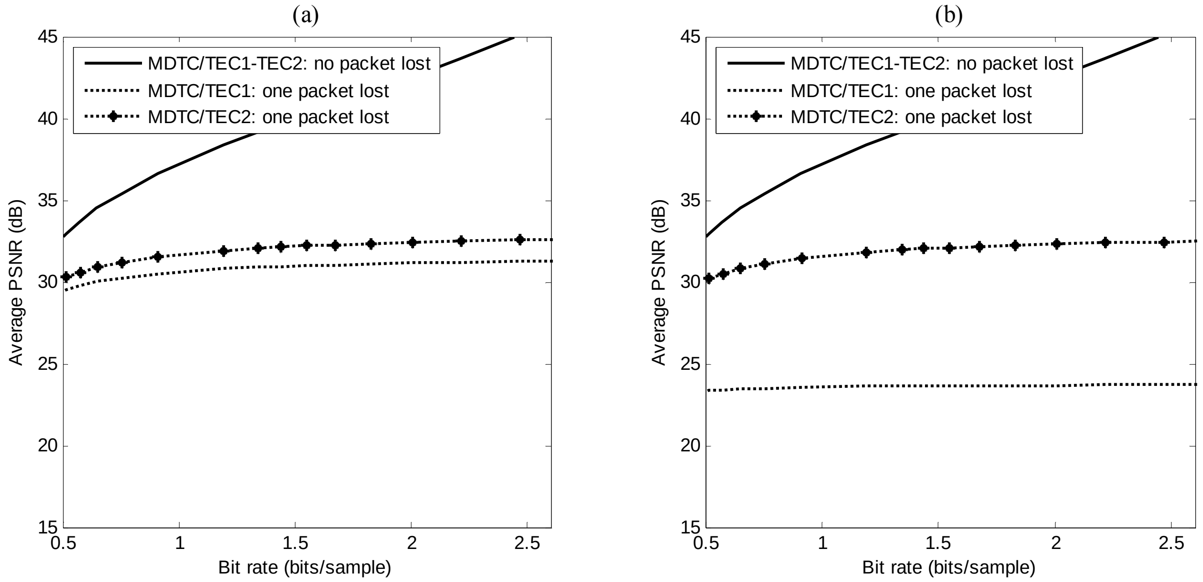


Figure 4.11: Average PSNR versus bits per sample for Lena image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

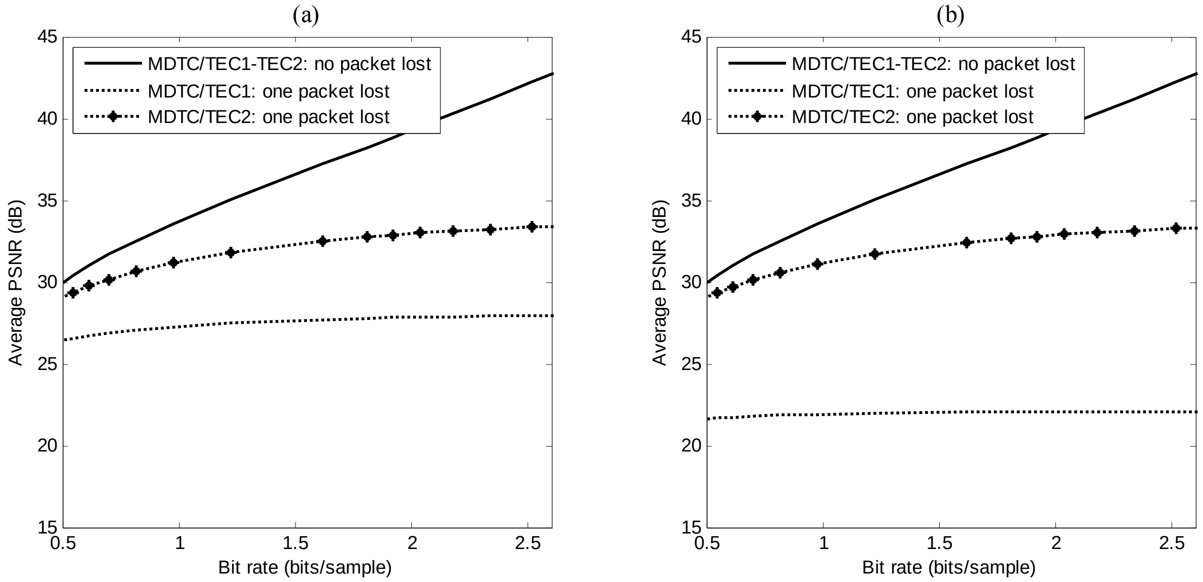


Figure 4.12: Average PSNR versus bits per sample for Goldhill image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

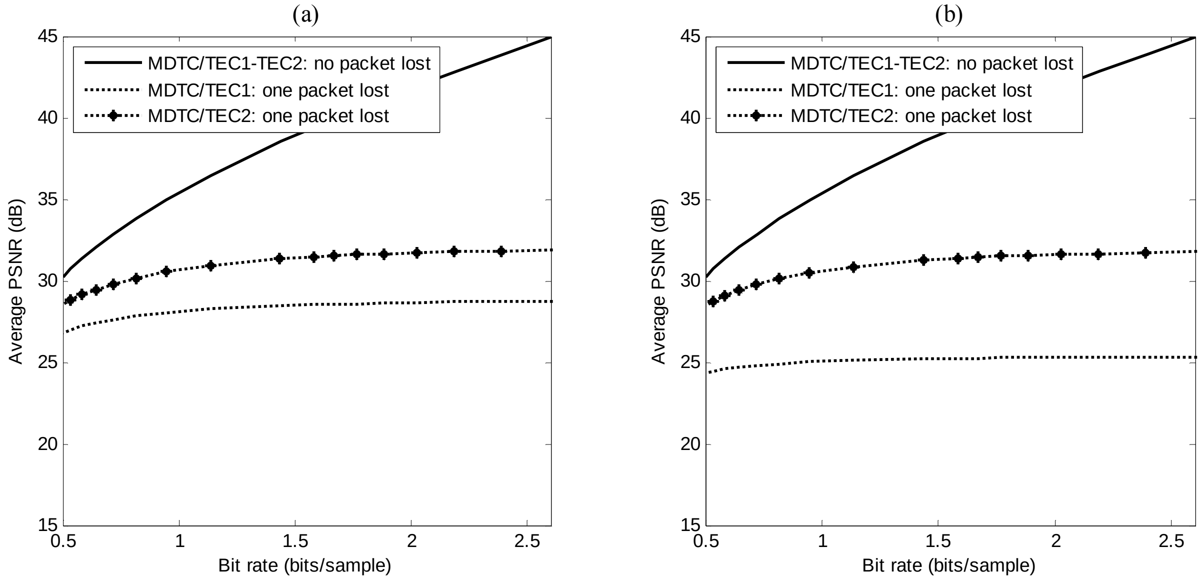


Figure 4.13: Average PSNR versus bits per sample for Boat image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

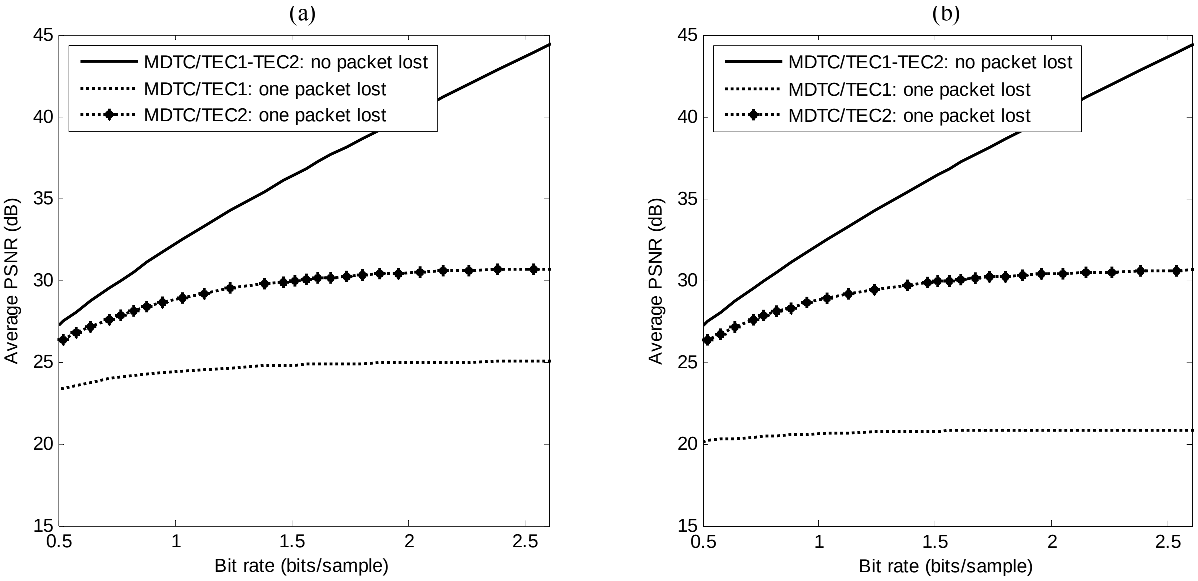


Figure 4.14: Average PSNR versus bits per sample for Lighthouse image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

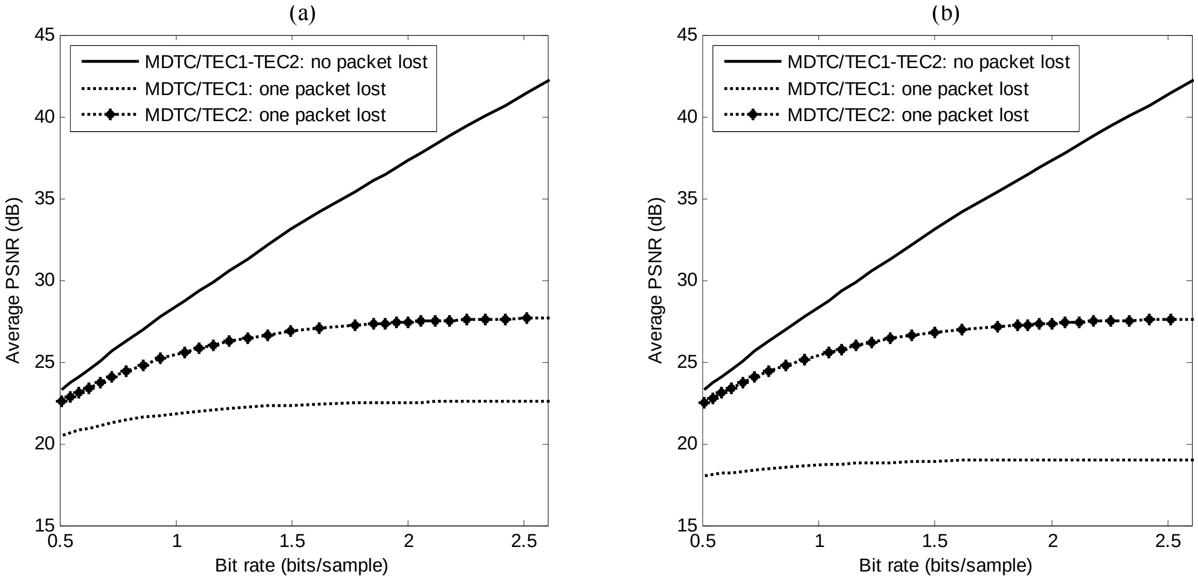


Figure 4.15: Average PSNR versus bits per sample for House image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

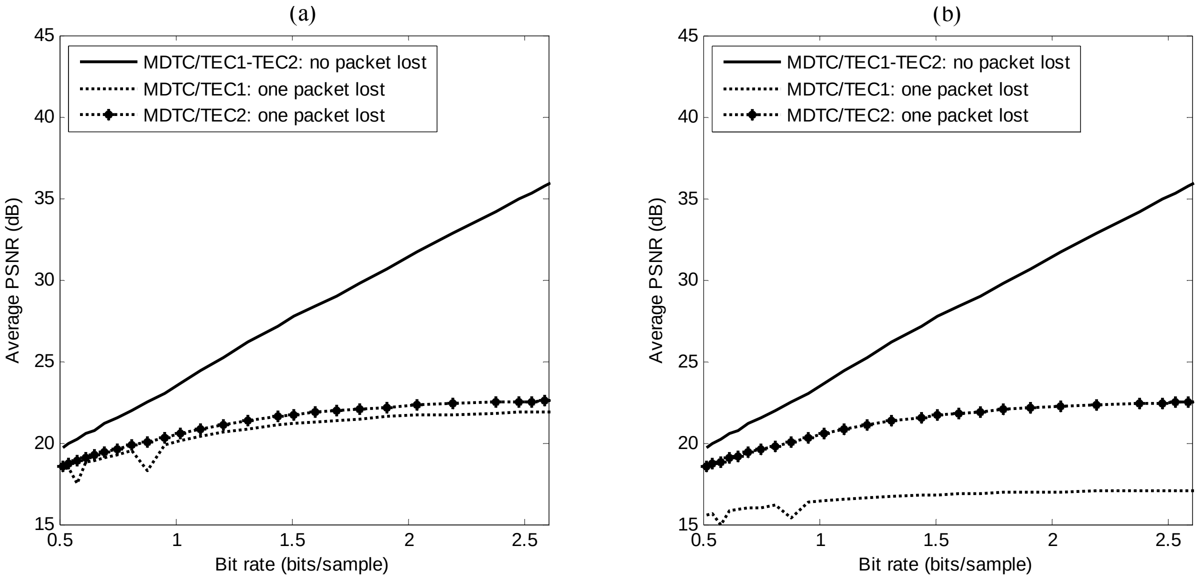


Figure 4.16: Average PSNR versus bits per sample for Nitf7 image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

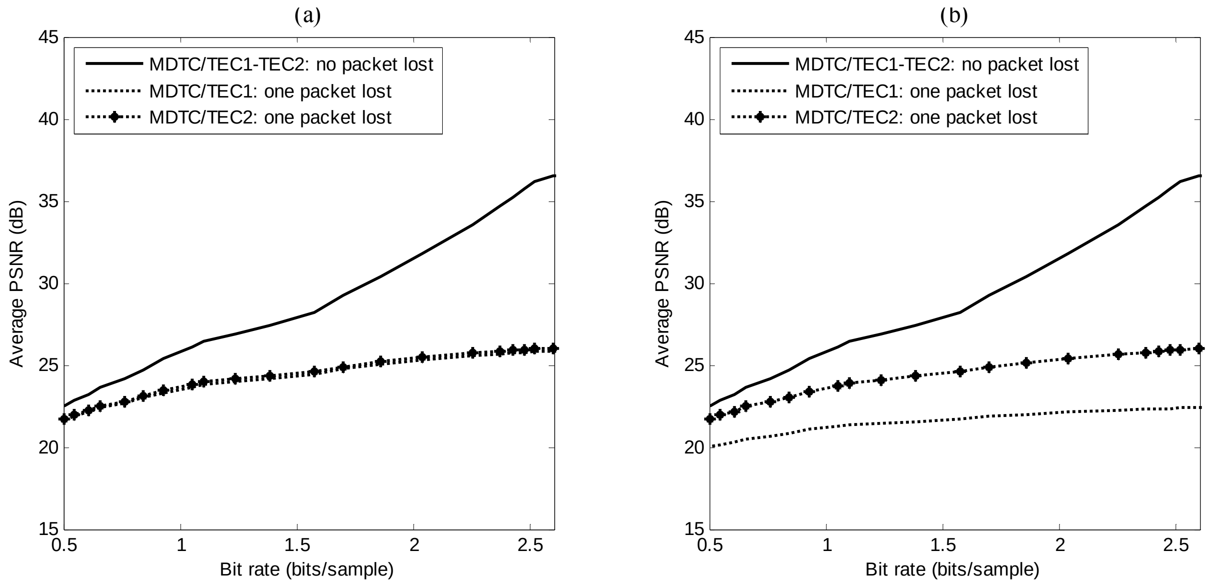


Figure 4.17: Average PSNR versus bits per sample for Baboon image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

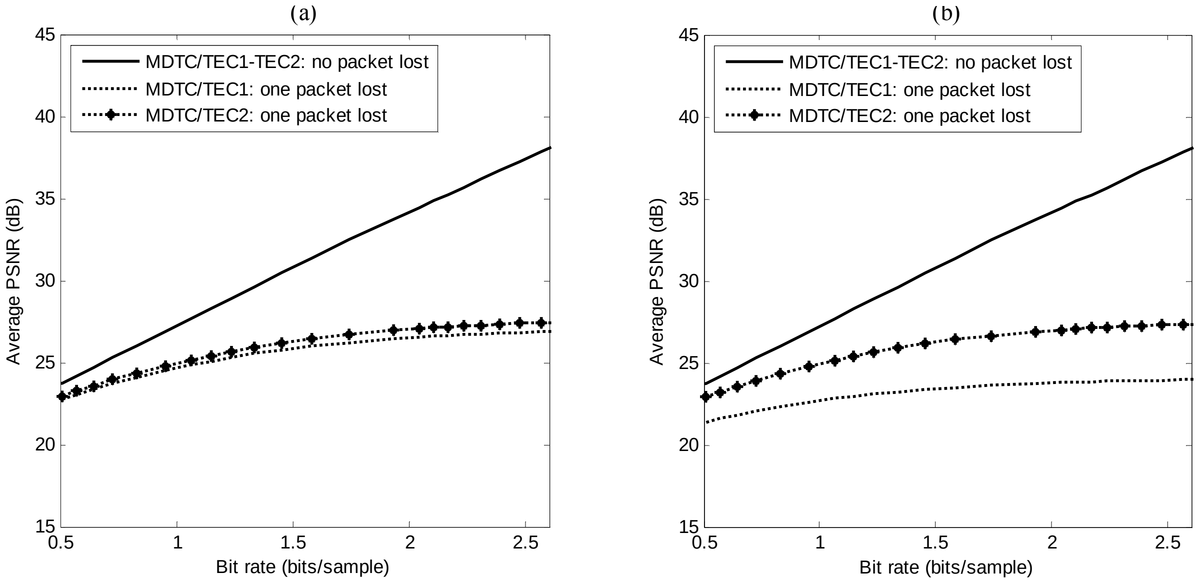


Figure 4.18: Average PSNR versus bits per sample for satellite image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

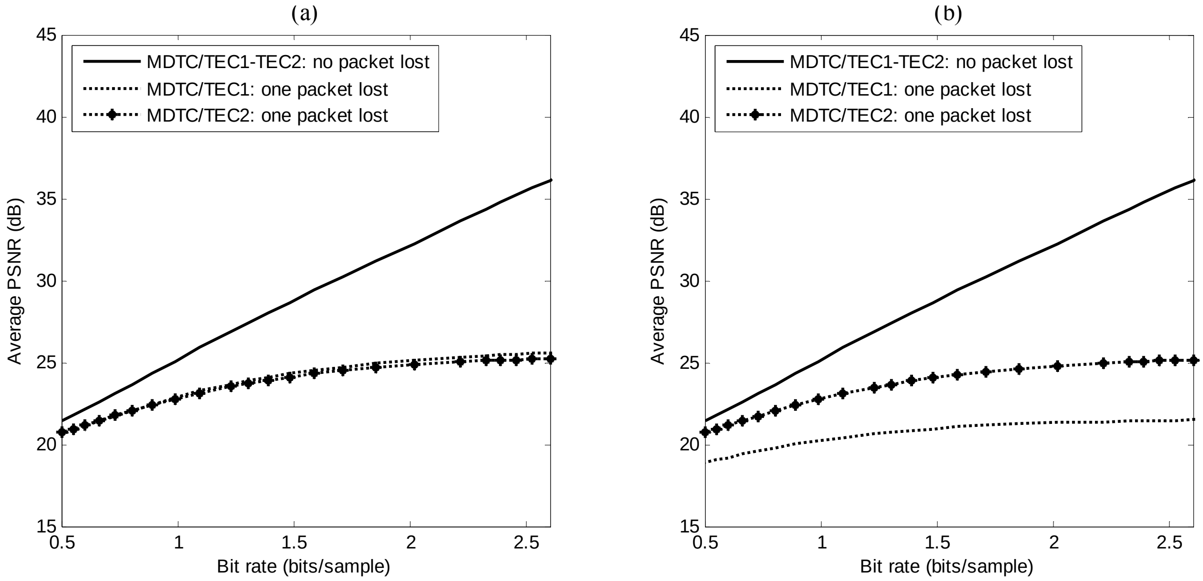


Figure 4.19: Average PSNR versus bits per sample for Sandiego image. (a) DC coefficients reliably communicated with some other means, (b) DC coefficients transmitted along with the four descriptions.

No. of packets dropped	Lena image		Goldhill image		Boat image	
	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$
	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	42.50	42.50	39.43	39.43	41.94	41.94
1	31.16	32.42	27.83	32.98	28.65	31.70
2	26.55	26.69	25.38	26.18	25.22	24.48
3	25.04	23.68	24.71	23.82	23.59	22.39

Table 4.3: PSNR as a function of the number of packets lost with DC coefficients being communicated reliably by some other means for Lena, Goldhill and boat images.

No. of packets dropped	Lena image		Goldhill image		Boat image	
	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$
	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	42.50	42.50	39.43	39.43	41.94	41.94
1	23.67	32.30	22.07	32.91	25.29	31.60
2	18.00	24.04	17.22	23.25	15.94	21.52
3	15.74	20.89	16.12	21.28	14.81	19.71

Table 4.4: PSNR as a function of the number of packets lost with DC coefficients being transmitted along with the four data streams for Lena, Goldhill and boat images.

No. of packets dropped	Lighthouse image		House image		Nitf7 image	
	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$
	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	39.93	39.93	37.34	37.34	31.42	31.42
1	24.95	30.45	22.52	27.42	21.64	22.27
2	21.22	22.67	19.93	20.05	18.00	16.49
3	20.65	19.95	17.79	17.56	16.13	14.81

Table 4.5: PSNR as a function of the number of packets lost with DC coefficients being communicated reliably by some other means for Lighthouse, House and Nitf7 images.

No. of packets dropped	Lighthouse image		House image		Nitf7 image	
	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$
	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	39.93	39.93	37.34	37.34	31.42	31.42
1	20.83	30.39	18.99	27.35	17.00	22.21
2	16.55	20.36	15.86	17.29	12.40	14.82
3	14.30	17.95	13.41	15.17	10.44	13.09

Table 4.6 PSNR as a function of the number of packets lost with DC coefficients being transmitted along with the four data streams for Lighthouse, House and Nitf7 images.

No. of packets dropped	Baboon image		Satellite image		Sandiego image	
	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$	$MDTC\ /Tec1_{DC}$	$MDTC\ /Tec2_{DC}$
	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	31.42	31.42	34.12	34.12	32.21	32.21
1	25.26	25.42	26.50	27.02	25.12	24.84
2	22.13	21.13	22.78	22.47	21.63	20.07
3	20.38	19.41	20.96	20.40	19.52	18.41

Table 4.7 PSNR as a function of the number of packets lost with DC coefficients being communicated reliably by some other means for Baboon, Satellite and Sandiego images.

No. of packets dropped	Baboon image		Satellite image		Sandiego image	
	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$	$MDTC\ /Tec1_{\overline{DC}}$	$MDTC\ /Tec2_{\overline{DC}}$
	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	31.42	31.42	34.12	34.12	32.21	32.21
1	22.11	25.38	23.76	26.96	21.31	24.78
2	18.63	19.68	20.36	20.52	18.64	18.70
3	16.86	18.05	18.15	18.82	16.47	17.24

Table 4.8 PSNR as a function of the number of packets lost with DC coefficients being transmitted along with the four data streams for Baboon, Satellite and Sandiego images.

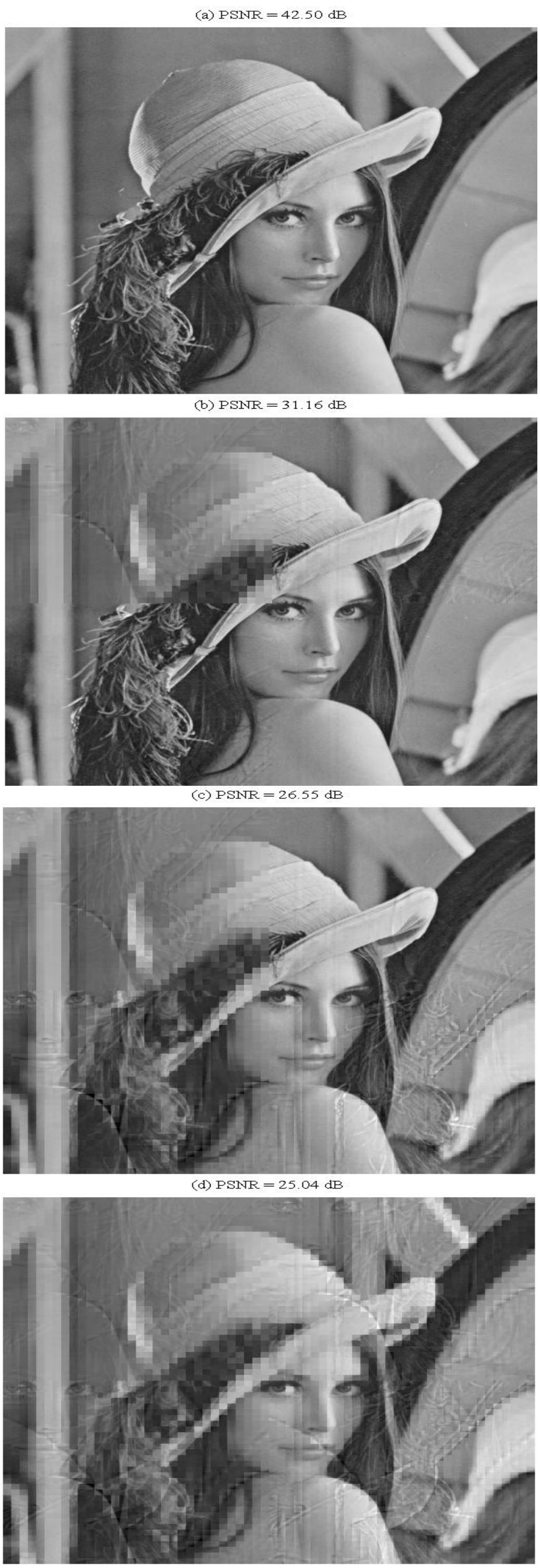


Figure 4.20: Lena image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

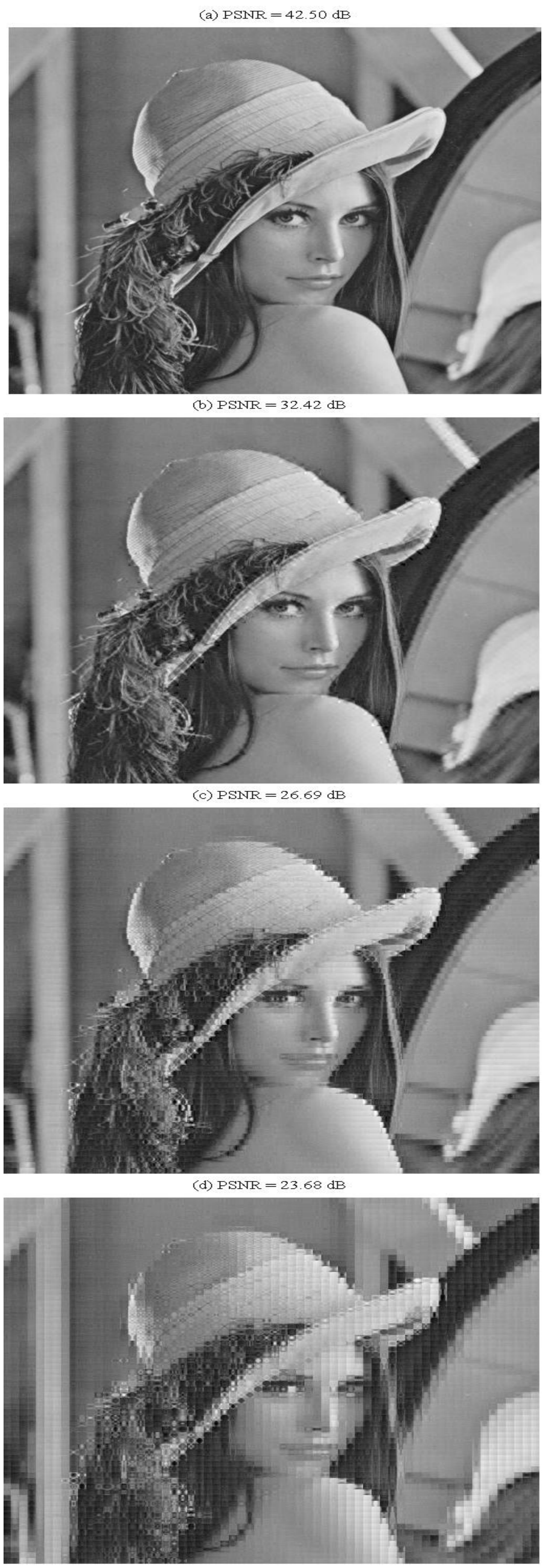
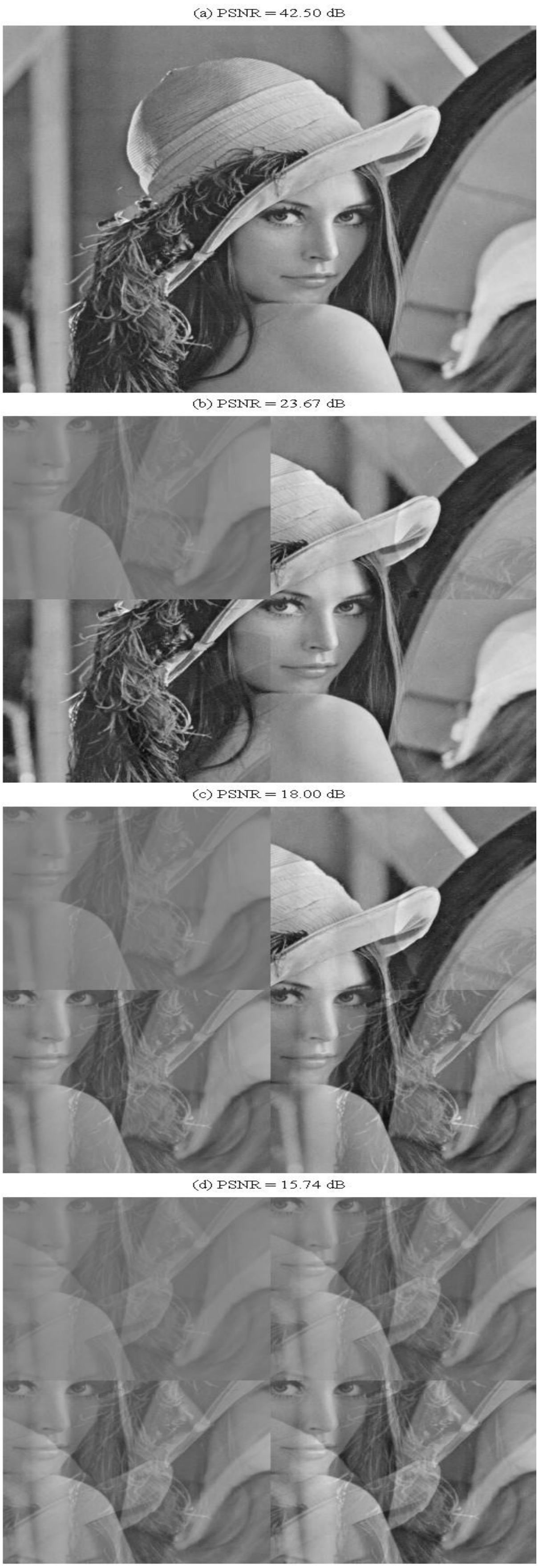


Figure 4.21: Lena image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



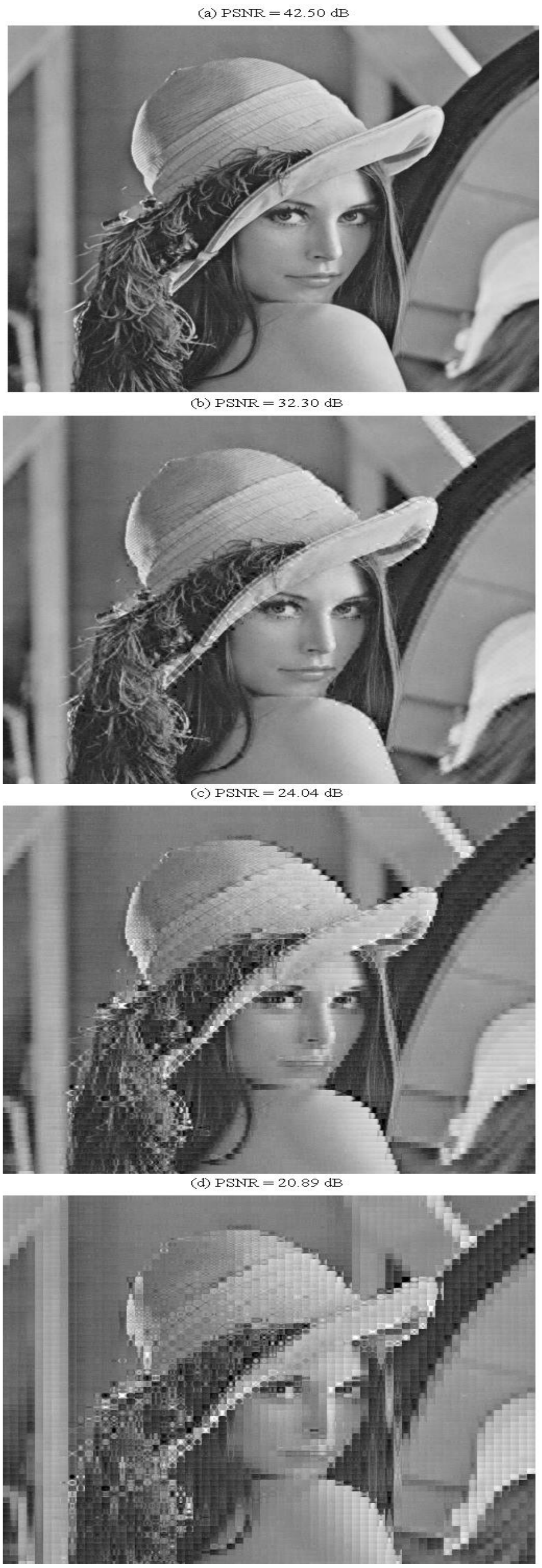
(a) PSNR = 42.50 dB

(b) PSNR = 23.67 dB

(c) PSNR = 18.00 dB

(d) PSNR = 15.74 dB

Figure 4.22: Lena image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



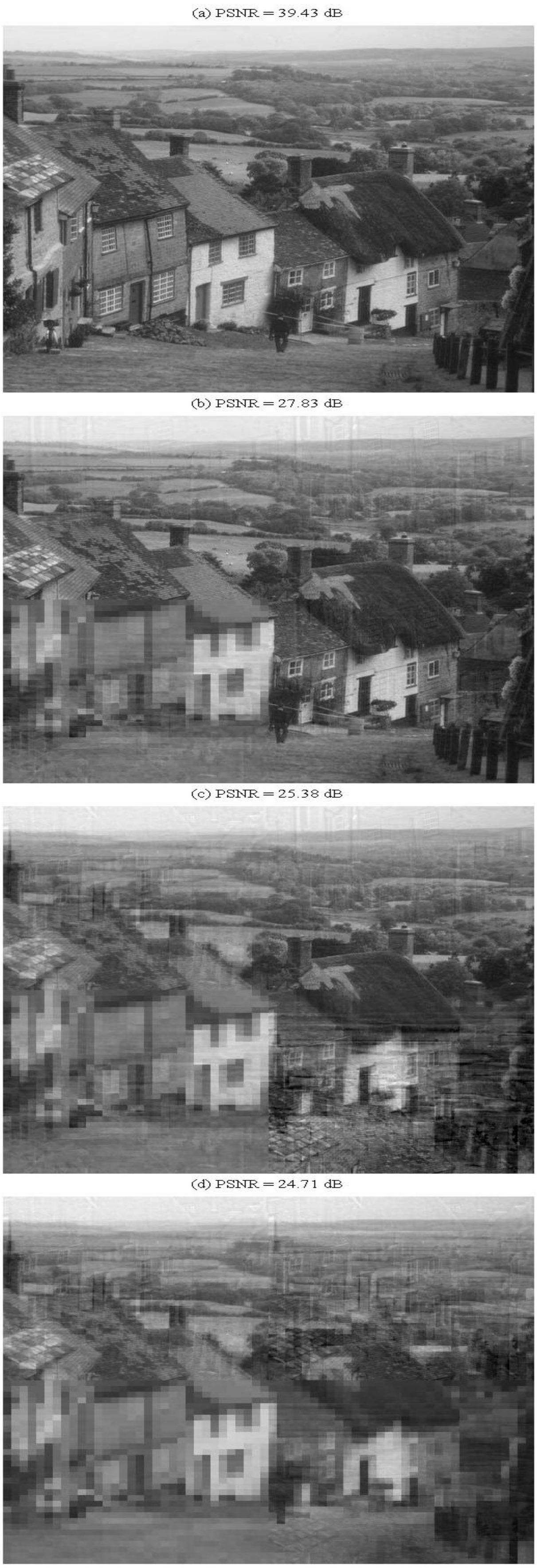
(a) PSNR = 42.50 dB

(b) PSNR = 32.30 dB

(c) PSNR = 24.04 dB

(d) PSNR = 20.89 dB

Figure 4.23: Lena image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



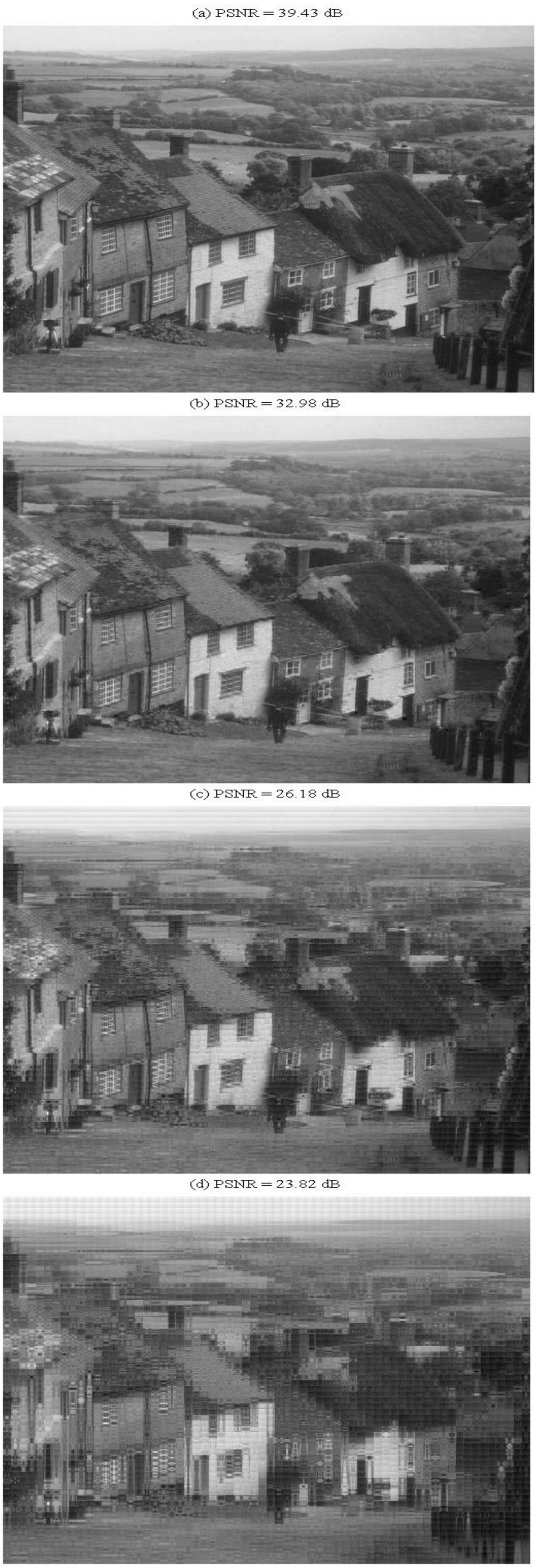
(a) PSNR = 39.43 dB

(b) PSNR = 27.83 dB

(c) PSNR = 25.38 dB

(d) PSNR = 24.71 dB

Figure 4.24: Goldhill image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



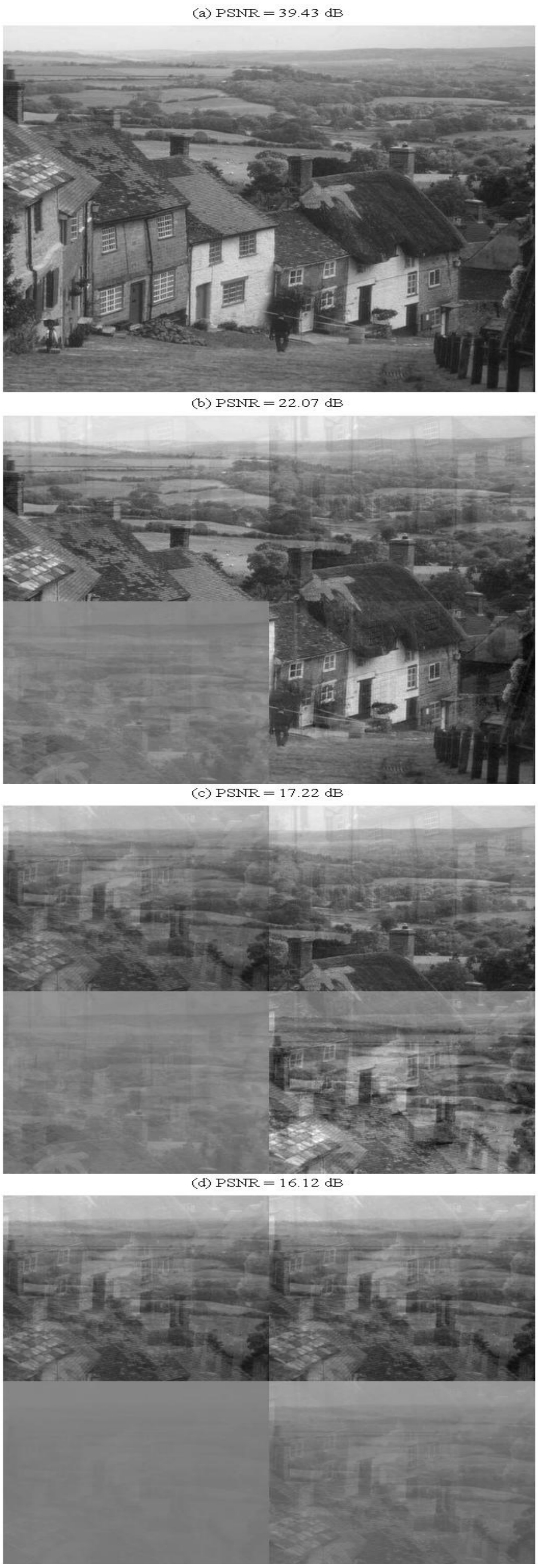
(a) PSNR = 39.43 dB

(b) PSNR = 32.98 dB

(c) PSNR = 26.18 dB

(d) PSNR = 23.82 dB

Figure 4.25: Goldhill image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



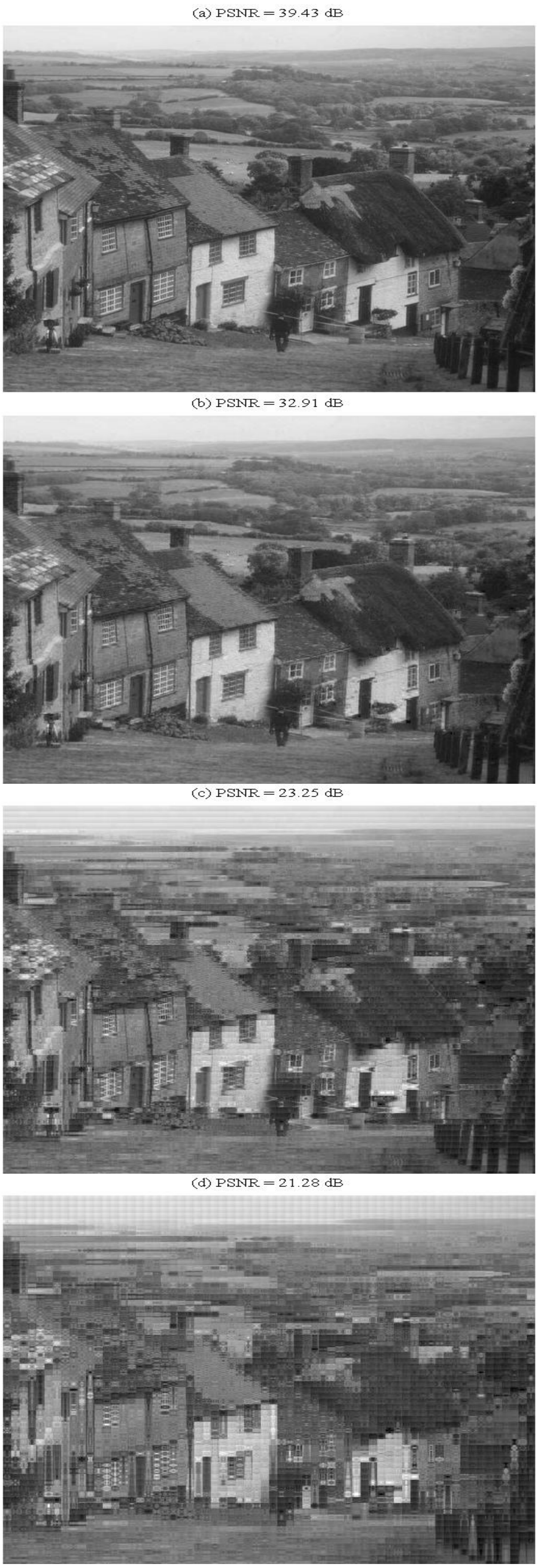
(a) PSNR = 39.43 dB

(b) PSNR = 22.07 dB

(c) PSNR = 17.22 dB

(d) PSNR = 16.12 dB

Figure 4.26: Goldhill image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



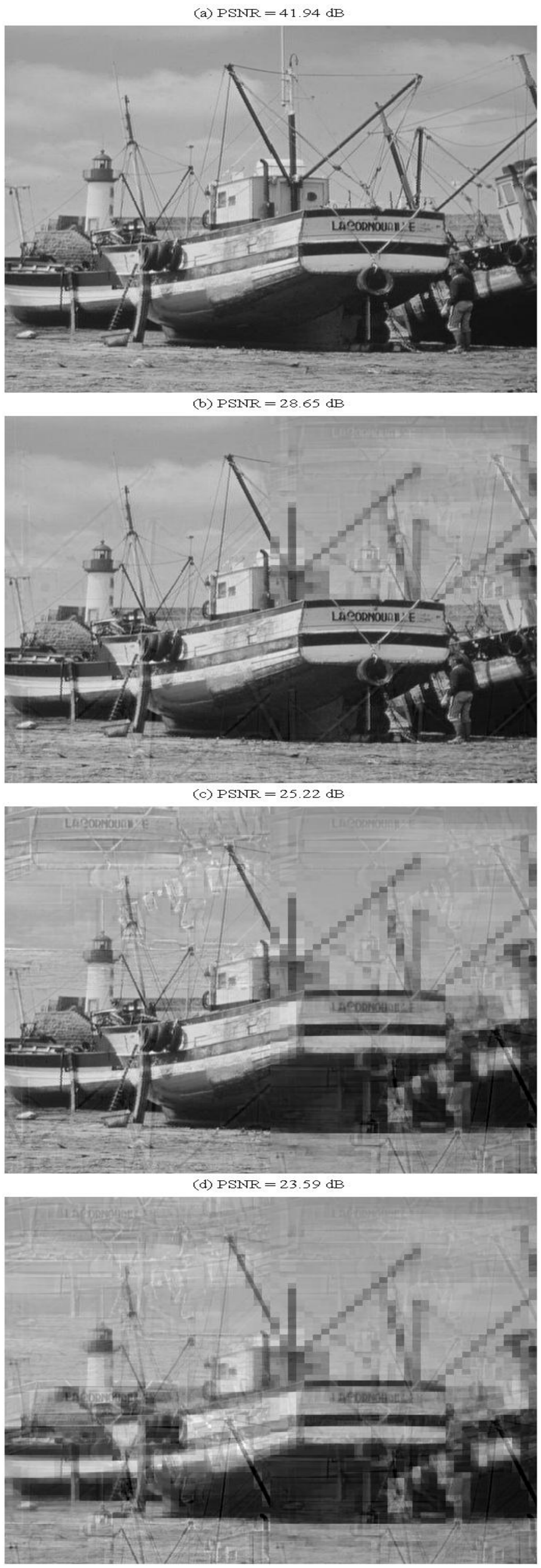
(a) PSNR = 39.43 dB

(b) PSNR = 32.91 dB

(c) PSNR = 23.25 dB

(d) PSNR = 21.28 dB

Figure 4.27: Goldhill image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



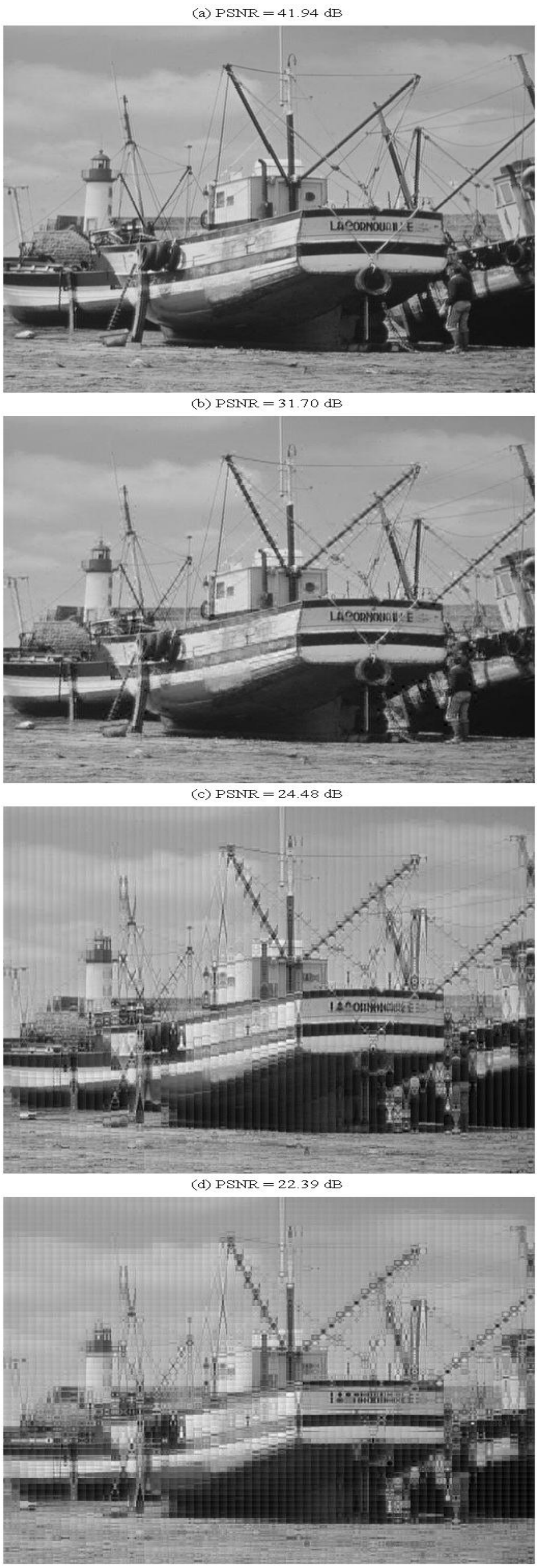
(a) PSNR = 41.94 dB

(b) PSNR = 28.65 dB

(c) PSNR = 25.22 dB

(d) PSNR = 23.59 dB

Figure 4.28: Boat image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



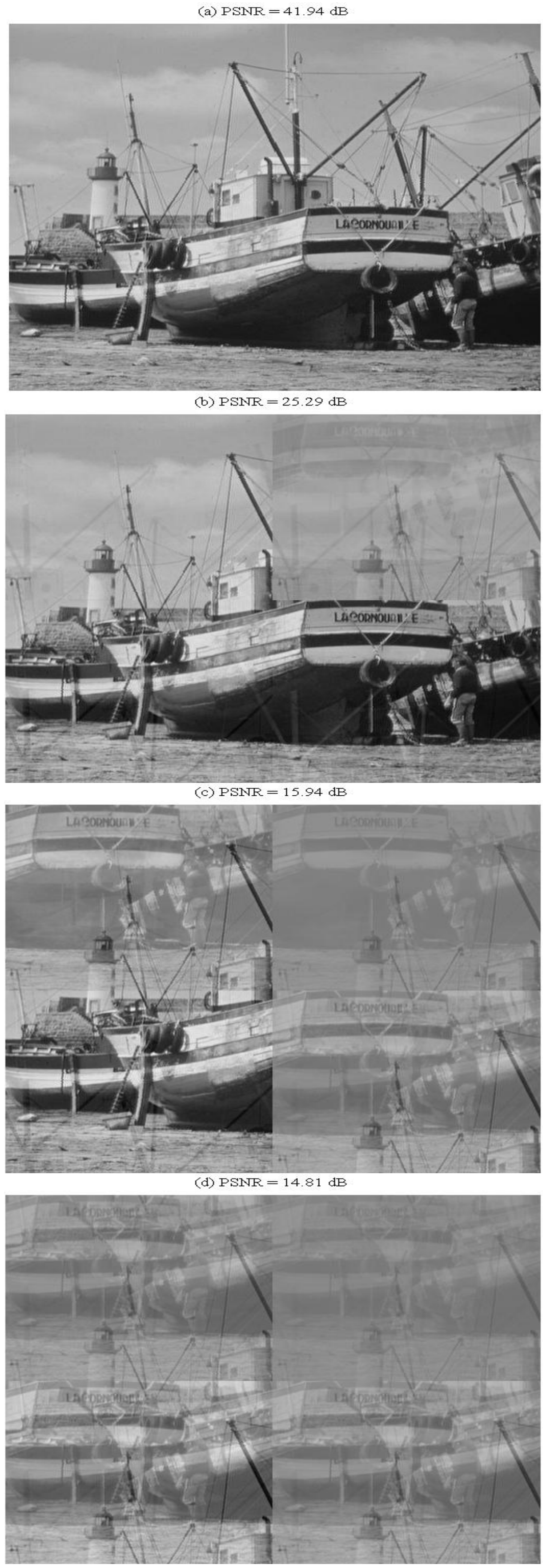
(a) PSNR = 41.94 dB

(b) PSNR = 31.70 dB

(c) PSNR = 24.48 dB

(d) PSNR = 22.39 dB

Figure 4.29: Boat image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



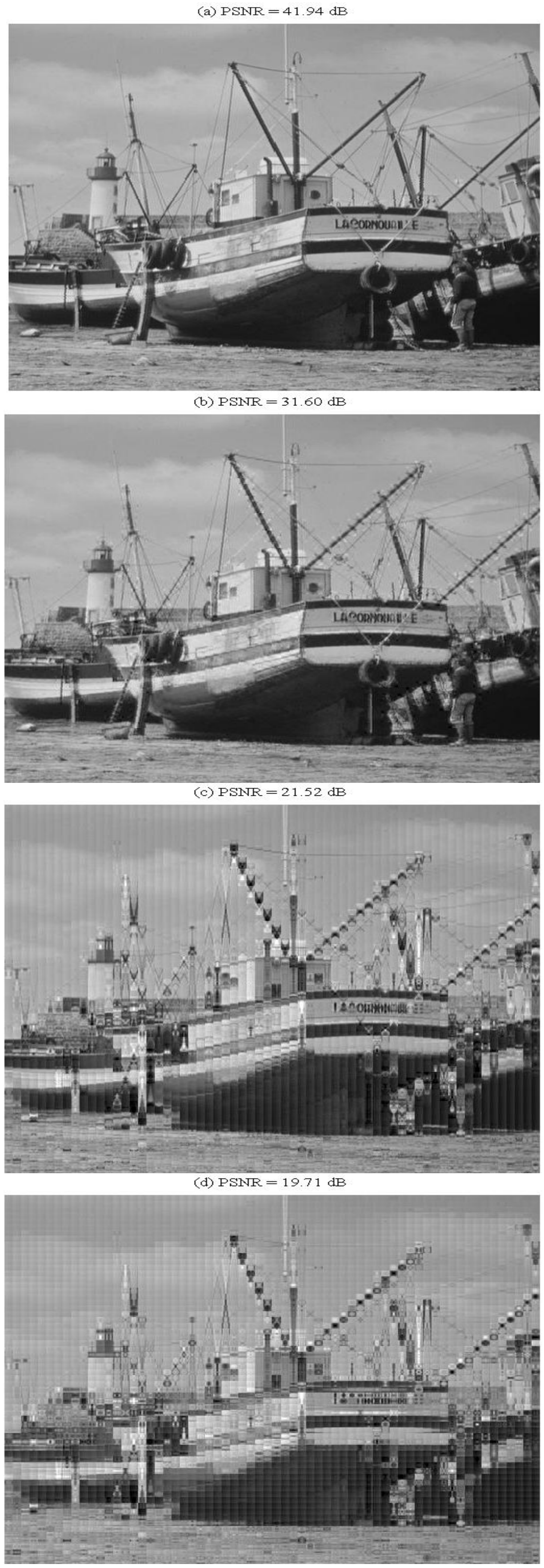
(a) PSNR = 41.94 dB

(b) PSNR = 25.29 dB

(c) PSNR = 15.94 dB

(d) PSNR = 14.81 dB

Figure 4.30: Boat image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



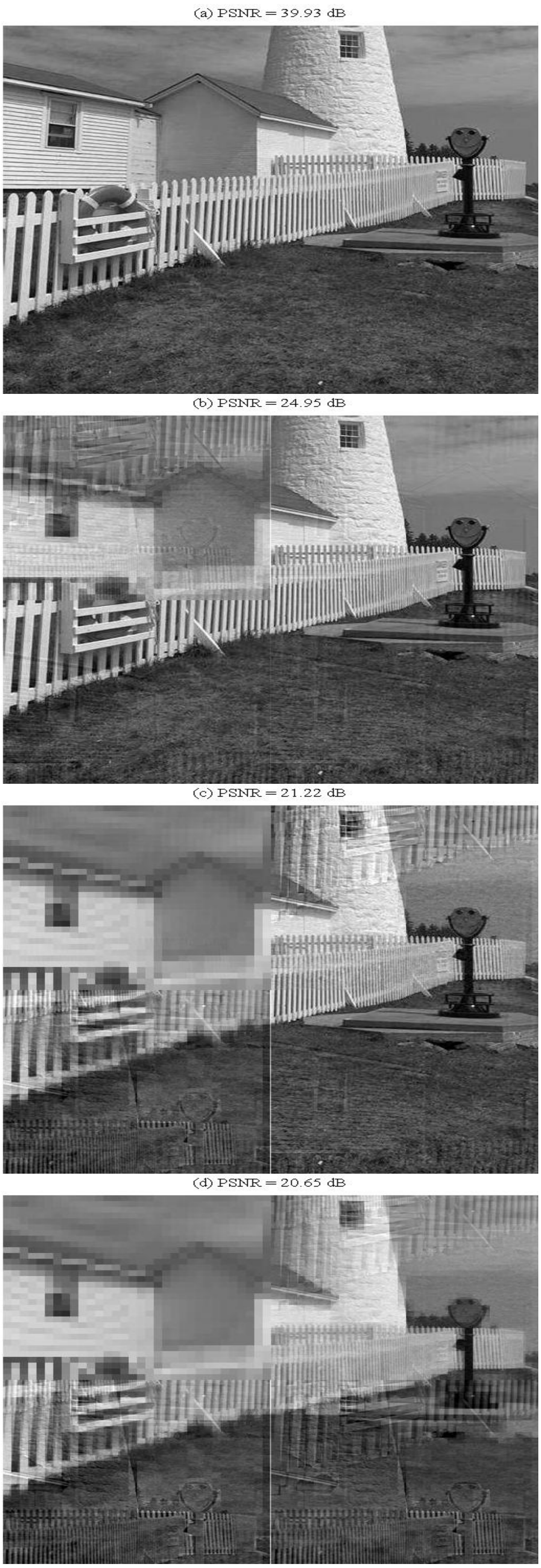
(a) PSNR = 41.94 dB

(b) PSNR = 31.60 dB

(c) PSNR = 21.52 dB

(d) PSNR = 19.71 dB

Figure 4.31: Boat image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



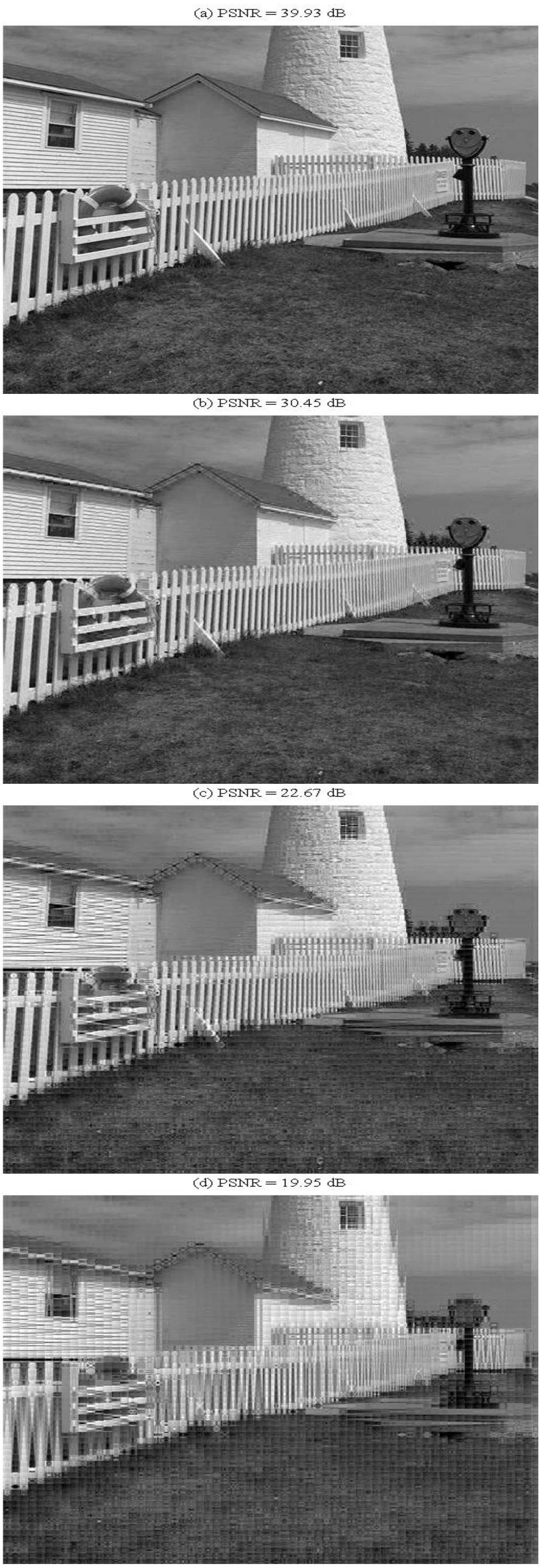
(a) PSNR = 39.93 dB

(b) PSNR = 24.95 dB

(c) PSNR = 21.22 dB

(d) PSNR = 20.65 dB

Figure 4.32: Lighthouse image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



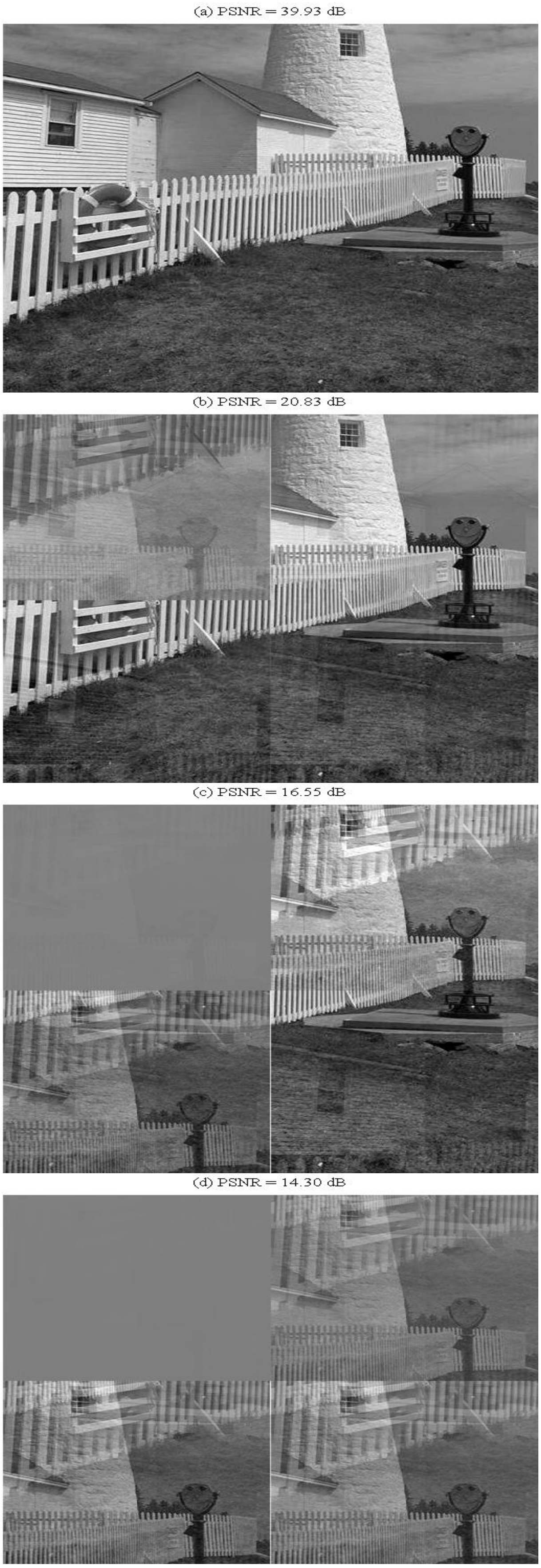
(a) PSNR = 39.93 dB

(b) PSNR = 30.45 dB

(c) PSNR = 22.67 dB

(d) PSNR = 19.95 dB

Figure 4.33: Lighthouse image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



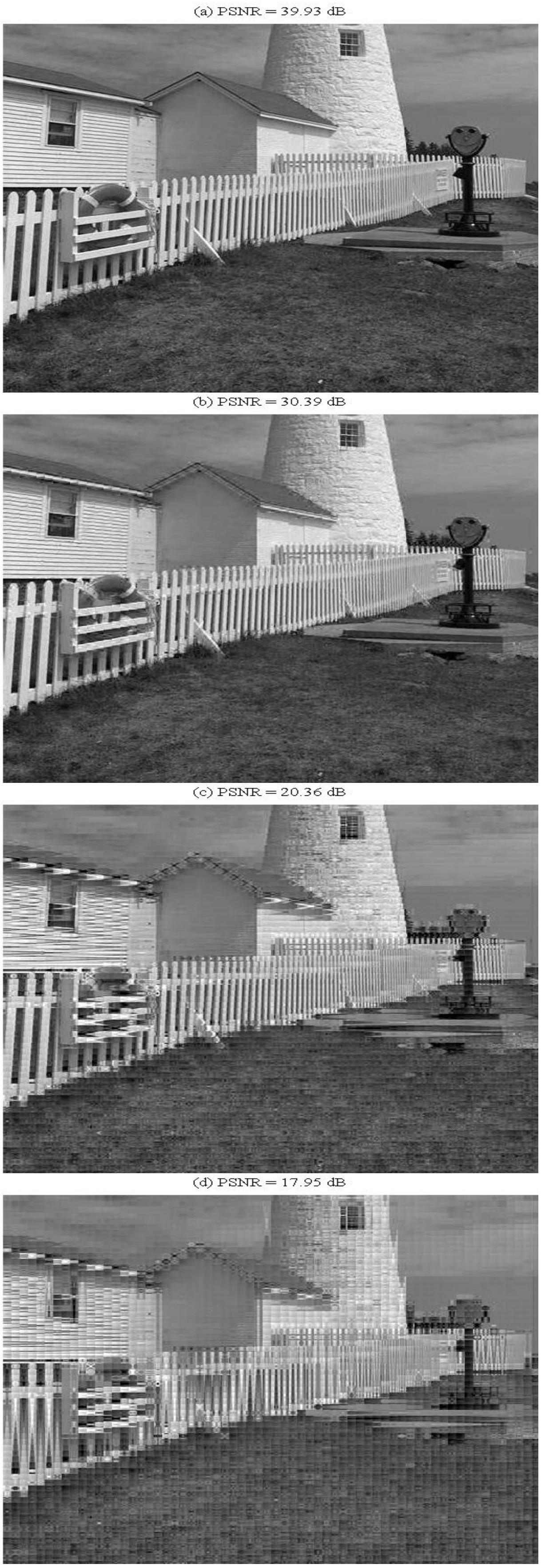
(a) PSNR = 39.93 dB

(b) PSNR = 20.83 dB

(c) PSNR = 16.55 dB

(d) PSNR = 14.30 dB

Figure 4.34: Lighthouse image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



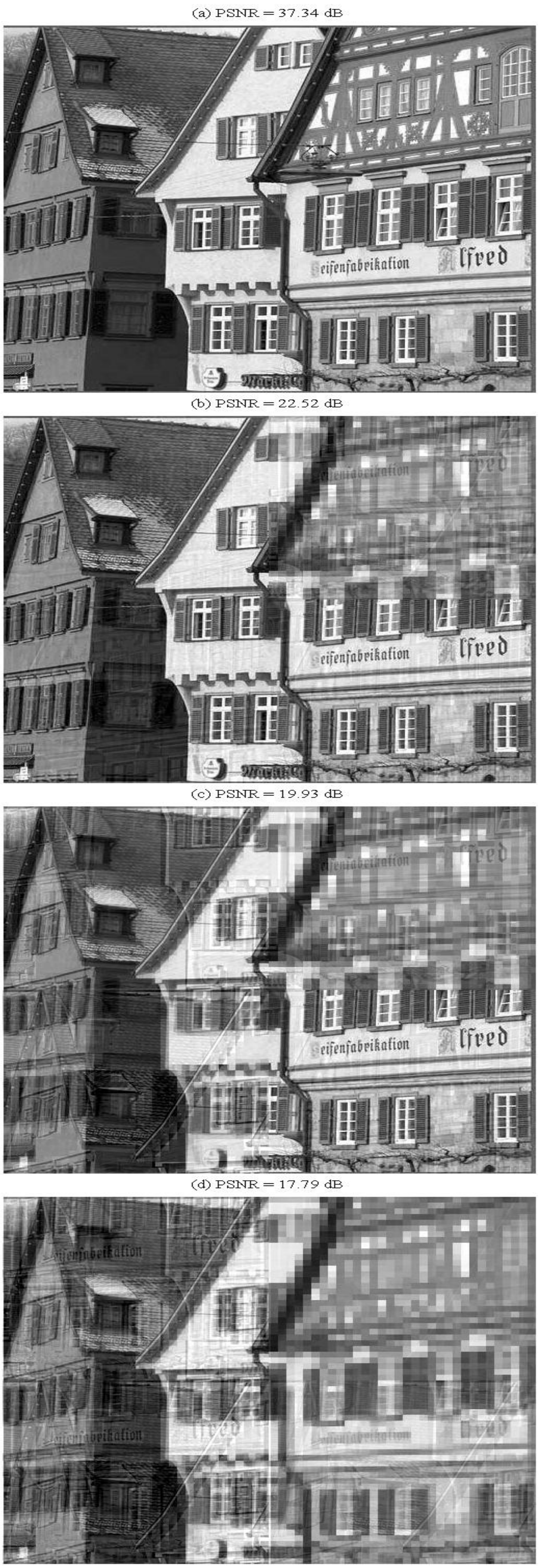
(a) PSNR = 39.93 dB

(b) PSNR = 30.39 dB

(c) PSNR = 20.36 dB

(d) PSNR = 17.95 dB

Figure 4.35: Lighthouse image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



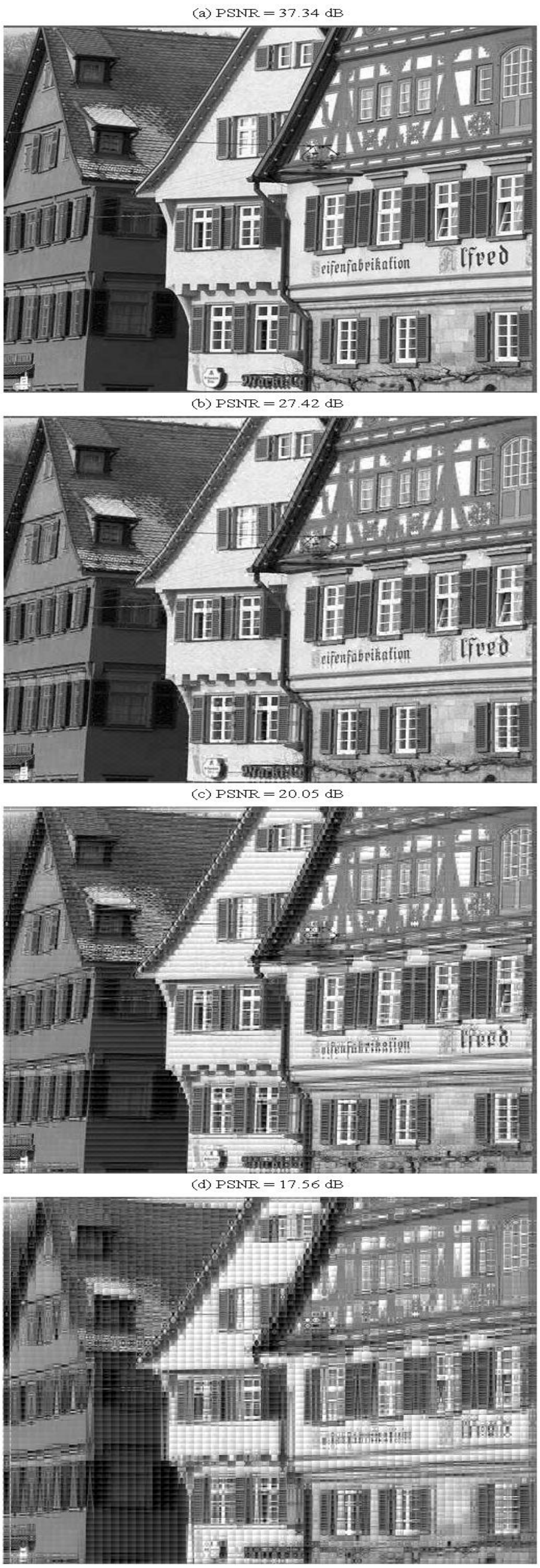
(a) PSNR = 37.34 dB

(b) PSNR = 22.52 dB

(c) PSNR = 19.93 dB

(d) PSNR = 17.79 dB

Figure 4.36: House image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



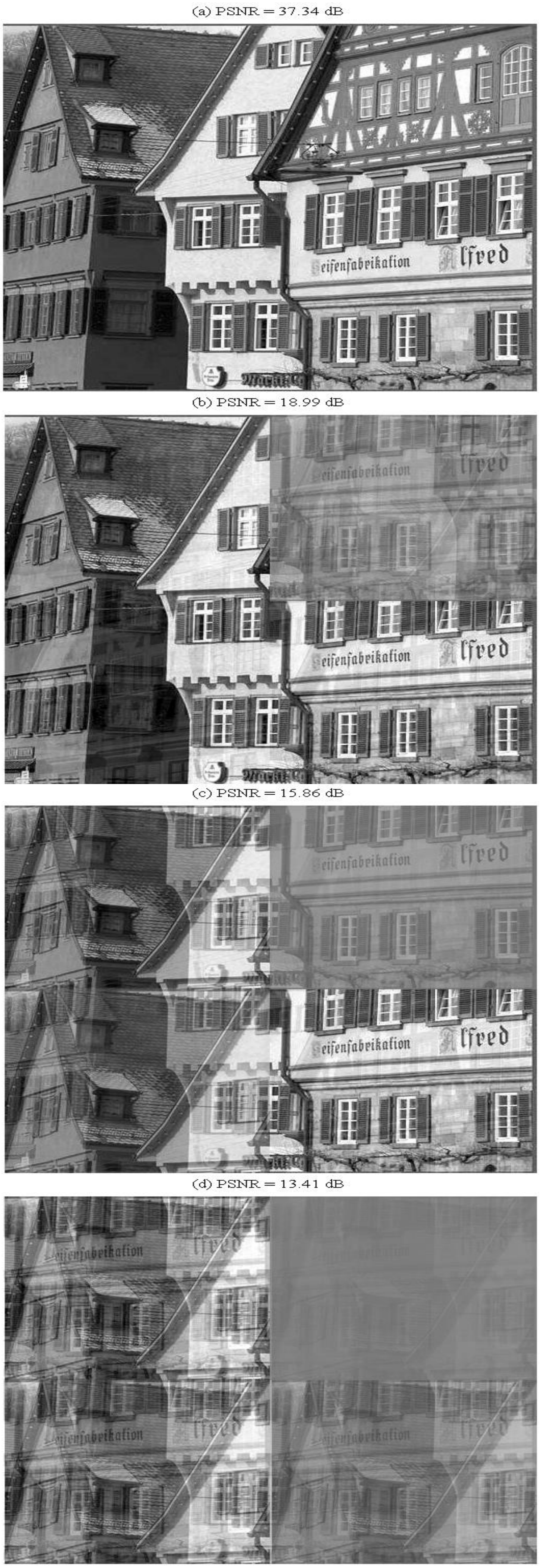
(a) PSNR = 37.34 dB

(b) PSNR = 27.42 dB

(c) PSNR = 20.05 dB

(d) PSNR = 17.56 dB

Figure 4.37: House image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



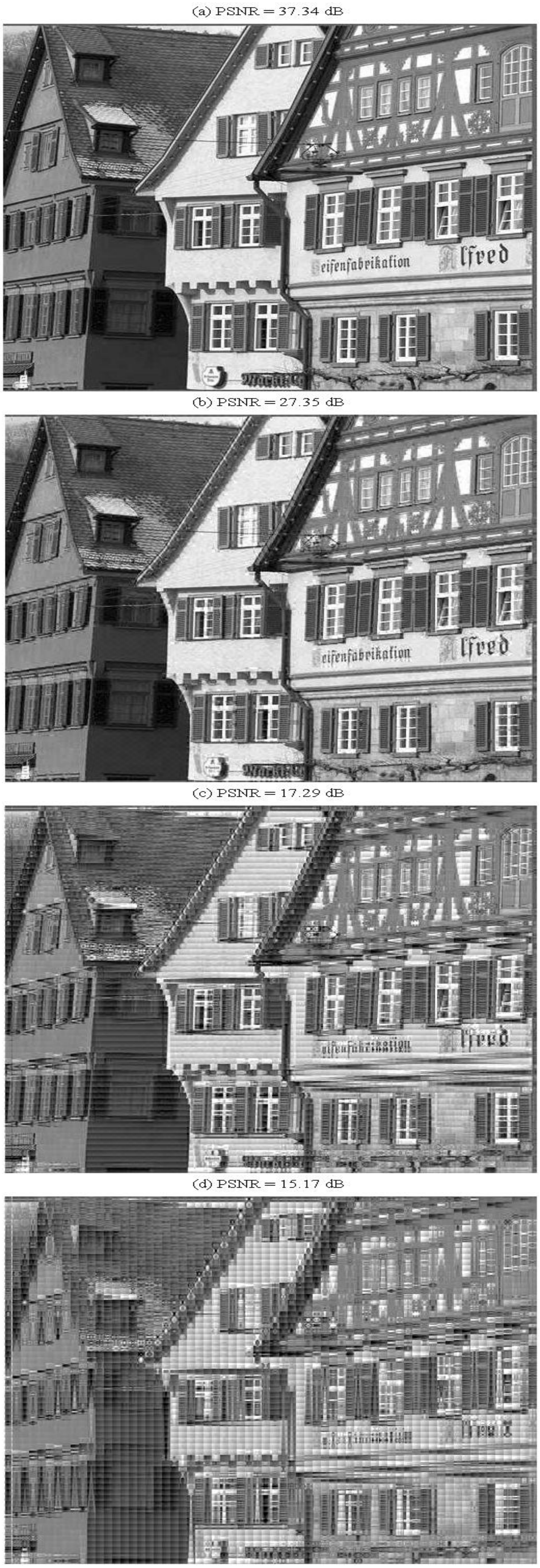
(a) PSNR = 37.34 dB

(b) PSNR = 18.99 dB

(c) PSNR = 15.86 dB

(d) PSNR = 13.41 dB

Figure 4.38: House image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



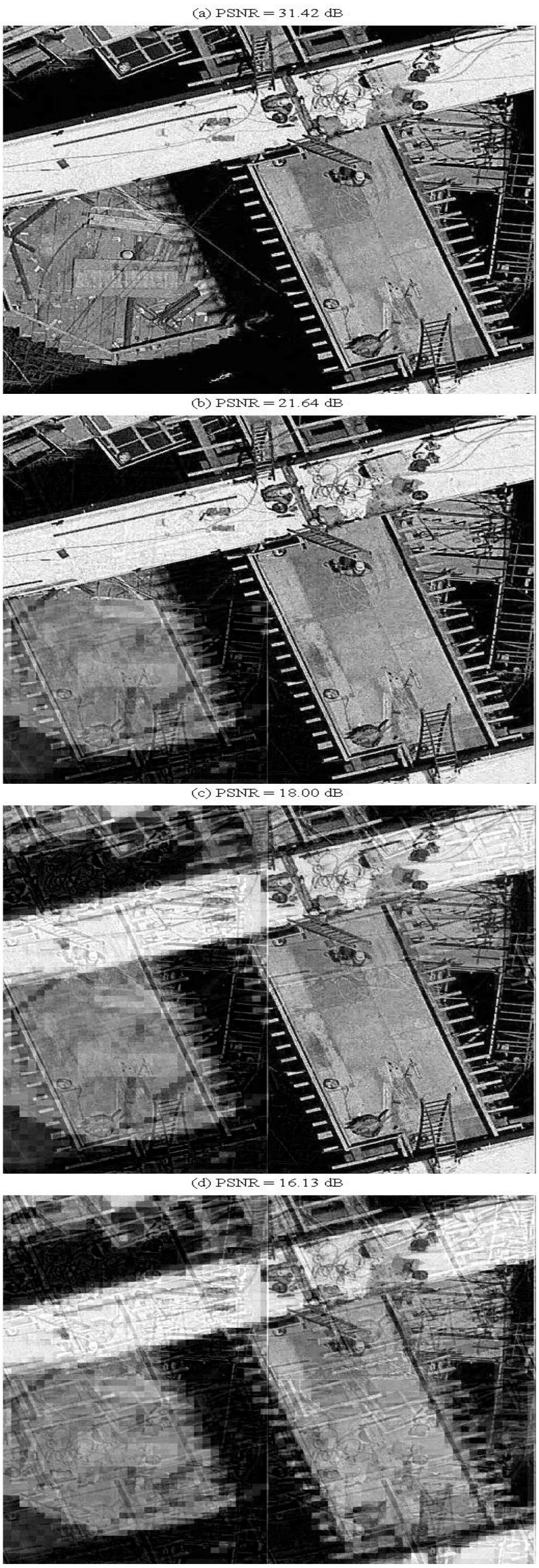
(a) PSNR = 37.34 dB

(b) PSNR = 27.35 dB

(c) PSNR = 17.29 dB

(d) PSNR = 15.17 dB

Figure 4.39: House image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



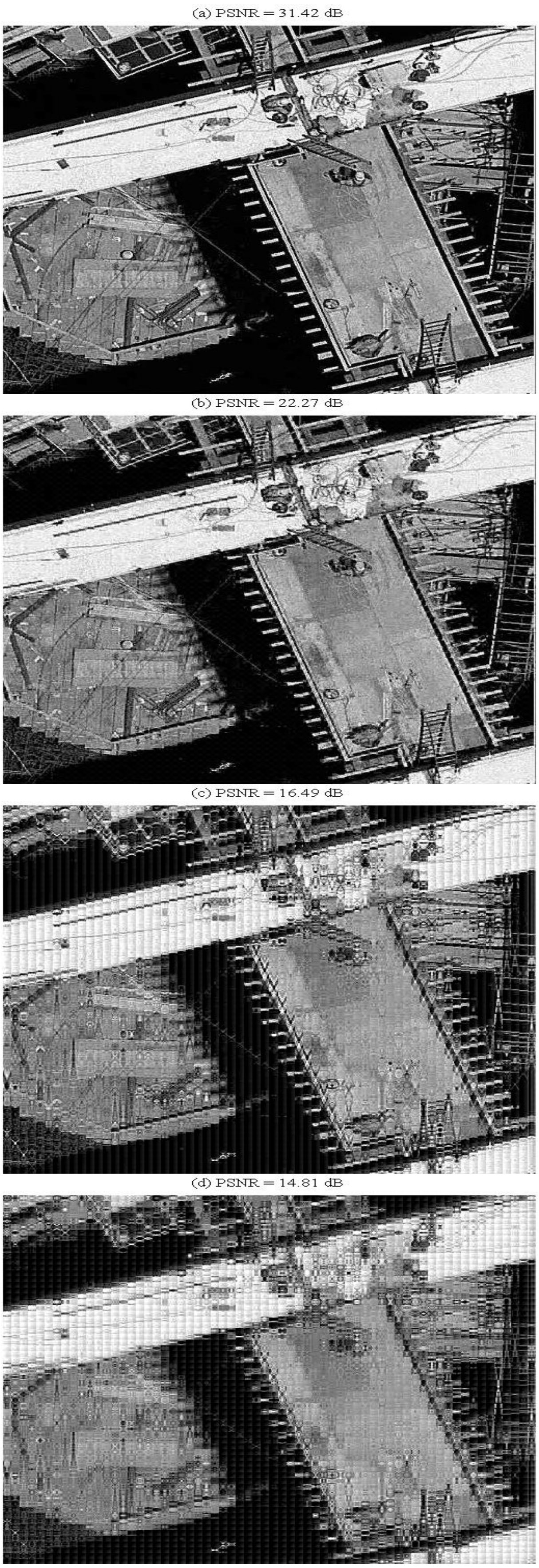
(a) PSNR = 31.42 dB

(b) PSNR = 21.64 dB

(c) PSNR = 18.00 dB

(d) PSNR = 16.13 dB

Figure 4.40: Nitf7 image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



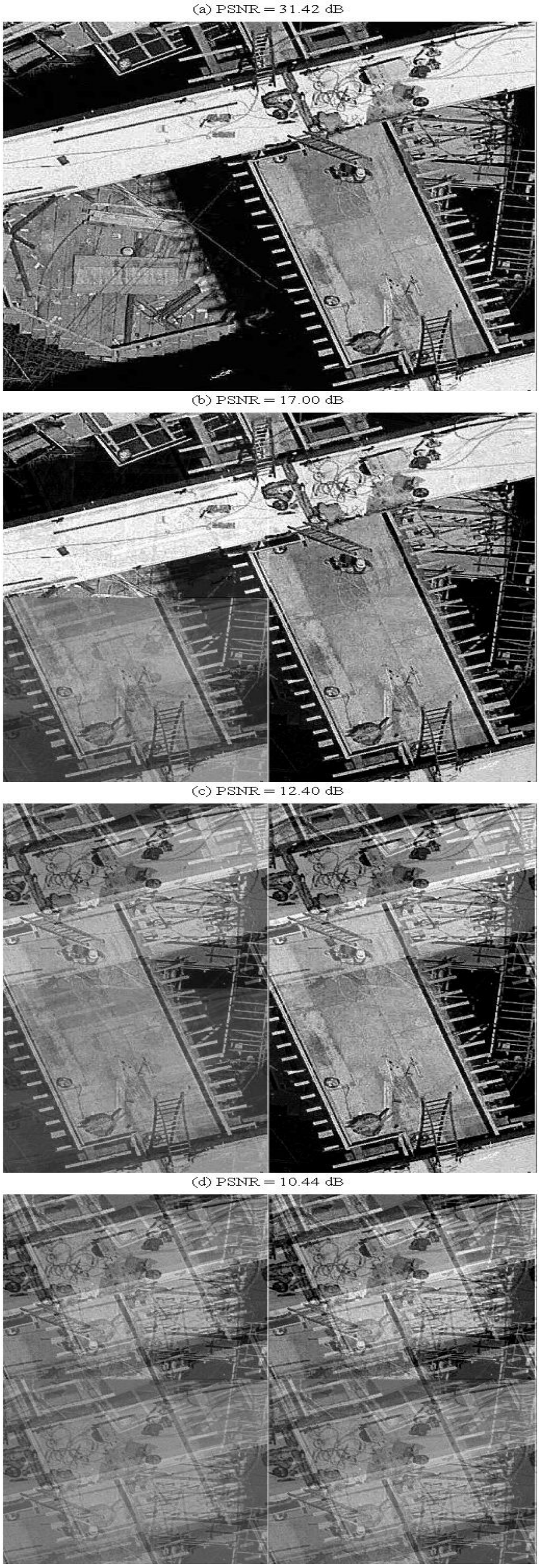
(a) PSNR = 31.42 dB

(b) PSNR = 22.27 dB

(c) PSNR = 16.49 dB

(d) PSNR = 14.81 dB

Figure 4.41: Nitf7 image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



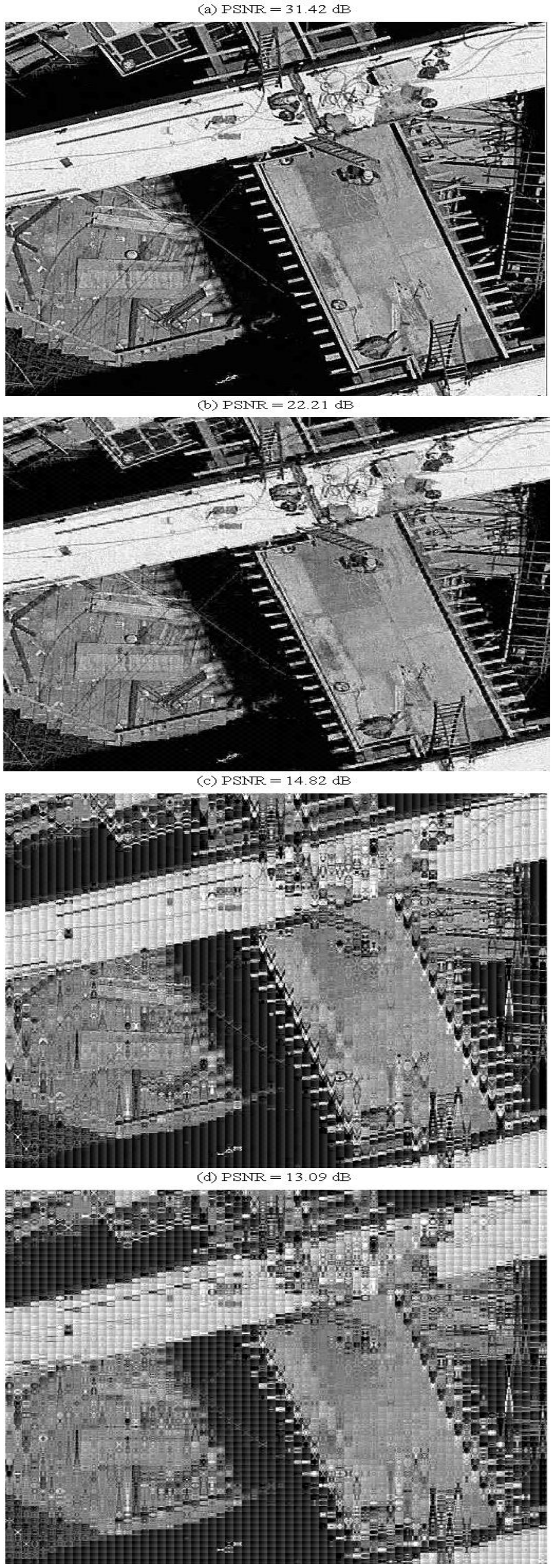
(a) PSNR = 31.42 dB

(b) PSNR = 17.00 dB

(c) PSNR = 12.40 dB

(d) PSNR = 10.44 dB

Figure 4.42: Nitf7 image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



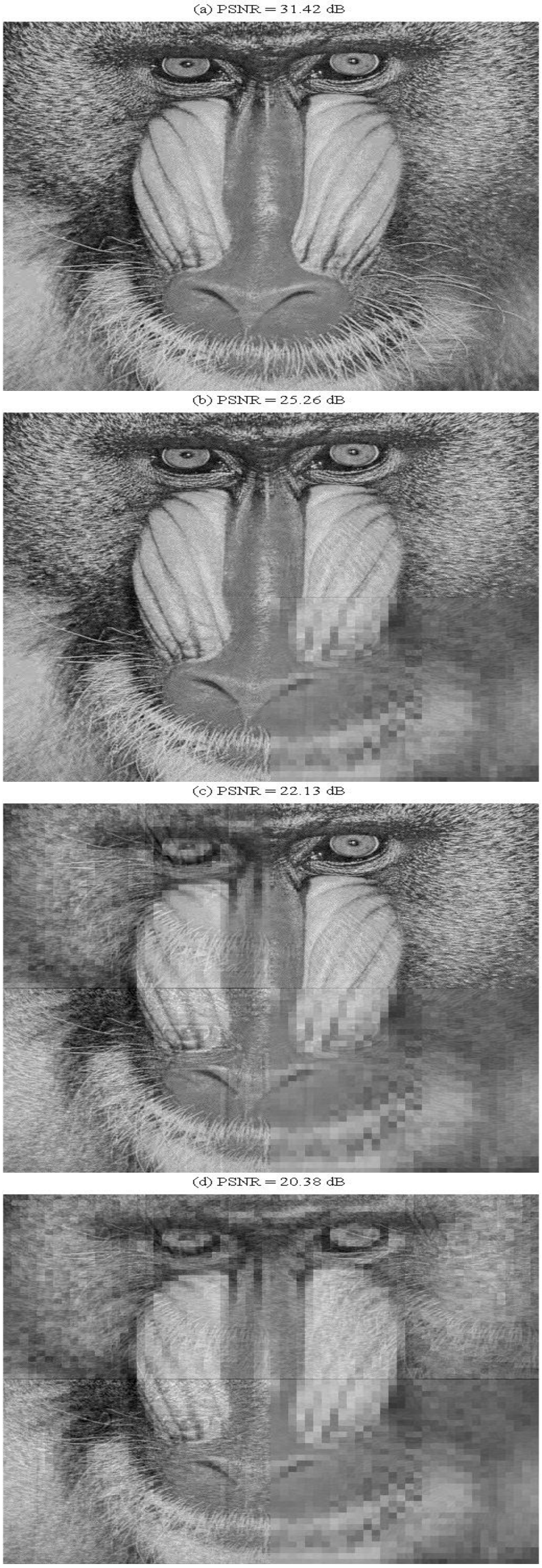
(a) PSNR = 31.42 dB

(b) PSNR = 22.21 dB

(c) PSNR = 14.82 dB

(d) PSNR = 13.09 dB

Figure 4.43: Nitf7 image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



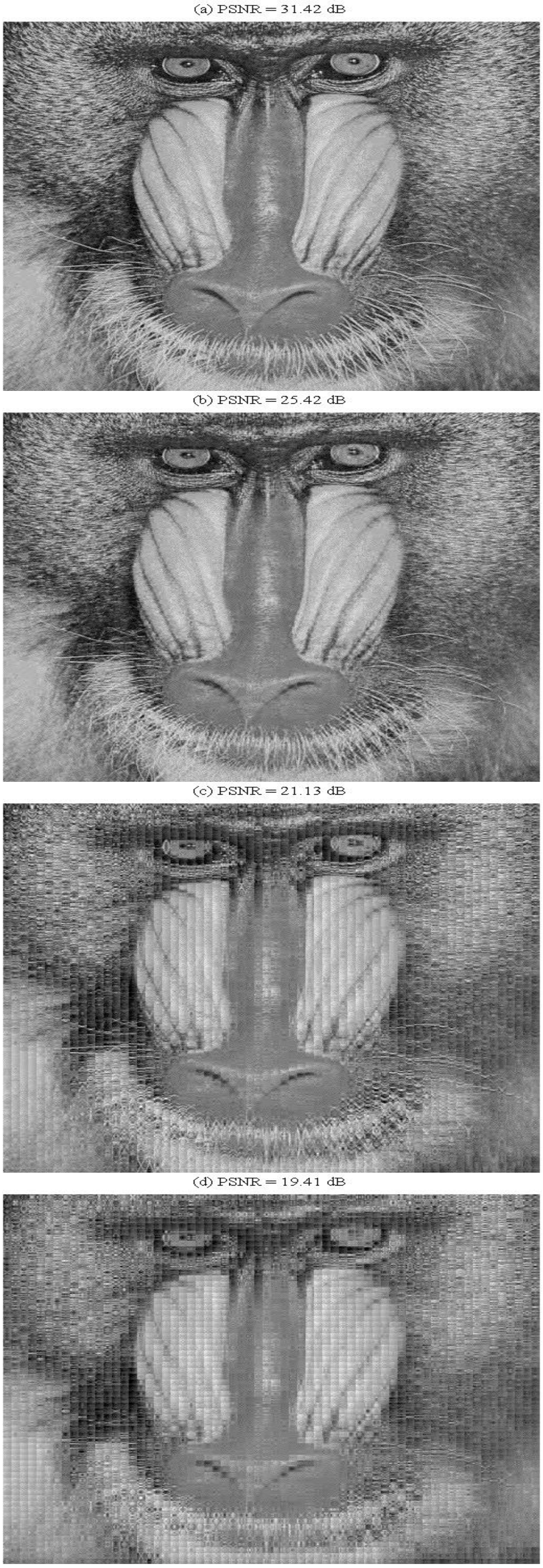
(a) PSNR = 31.42 dB

(b) PSNR = 25.26 dB

(c) PSNR = 22.13 dB

(d) PSNR = 20.38 dB

Figure 4.44: Baboon image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



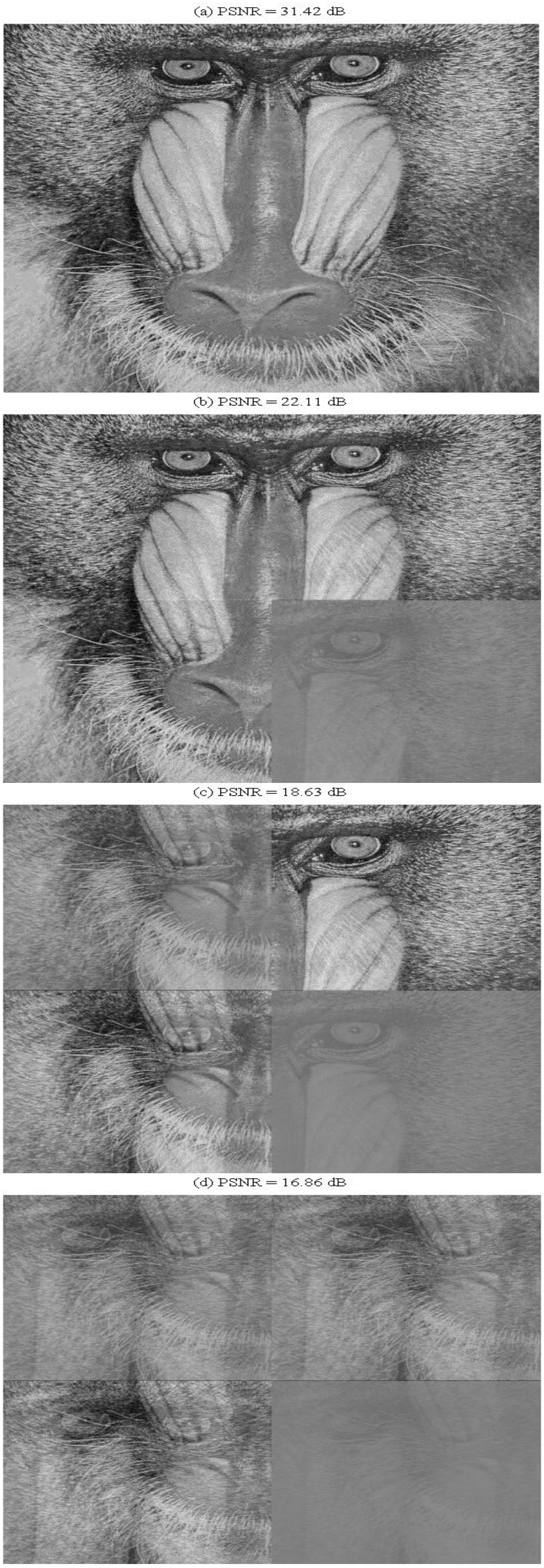
(a) PSNR = 31.42 dB

(b) PSNR = 25.42 dB

(c) PSNR = 21.13 dB

(d) PSNR = 19.41 dB

Figure 4.45: Baboon image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



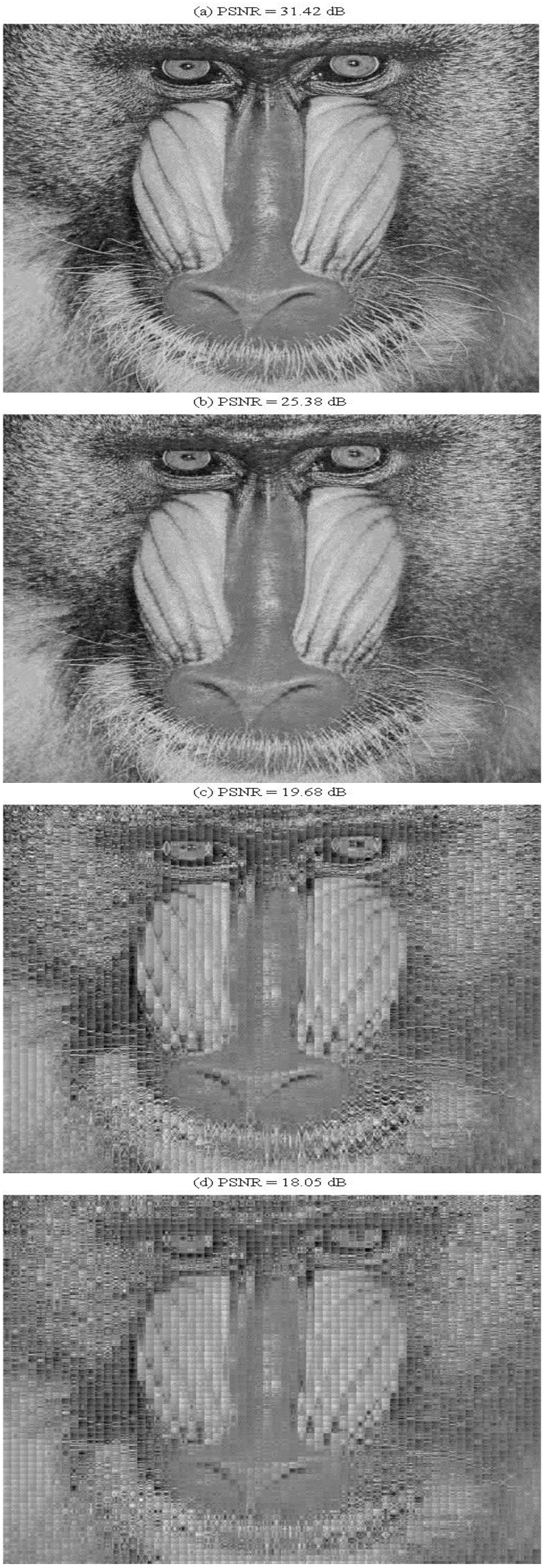
(a) PSNR = 31.42 dB

(b) PSNR = 22.11 dB

(c) PSNR = 18.63 dB

(d) PSNR = 16.86 dB

Figure 4.46: Baboon image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



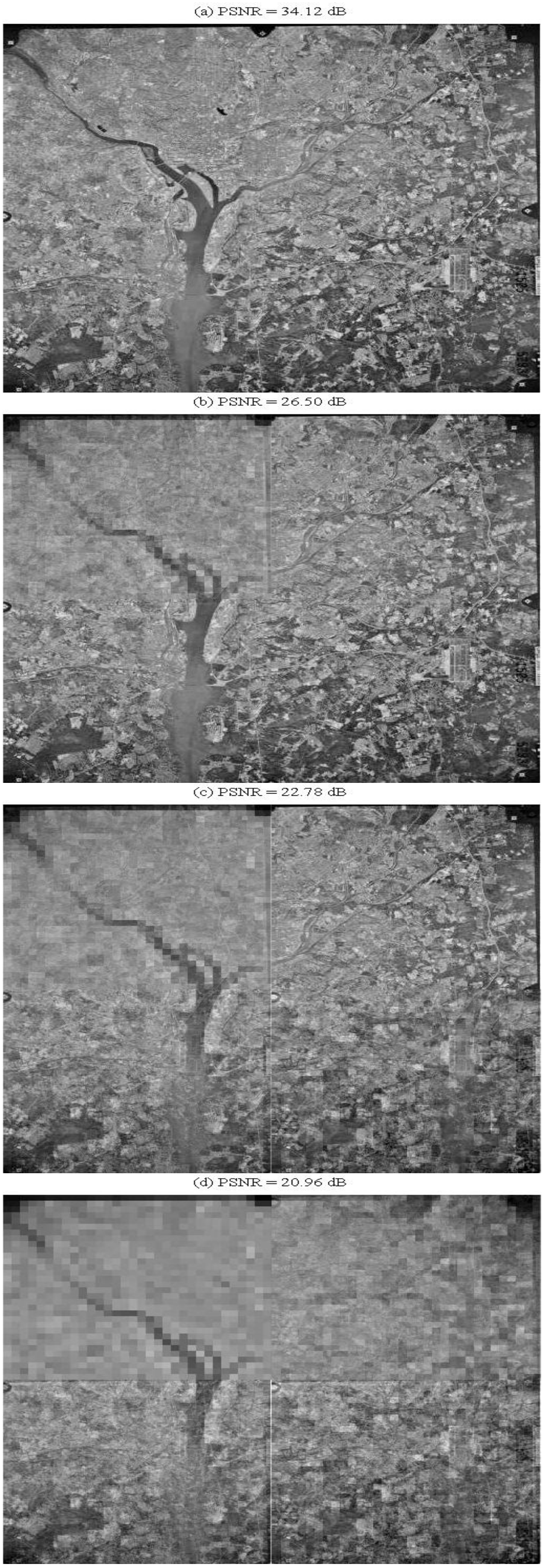
(a) PSNR = 31.42 dB

(b) PSNR = 25.38 dB

(c) PSNR = 19.68 dB

(d) PSNR = 18.05 dB

Figure 4.47: Baboon image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



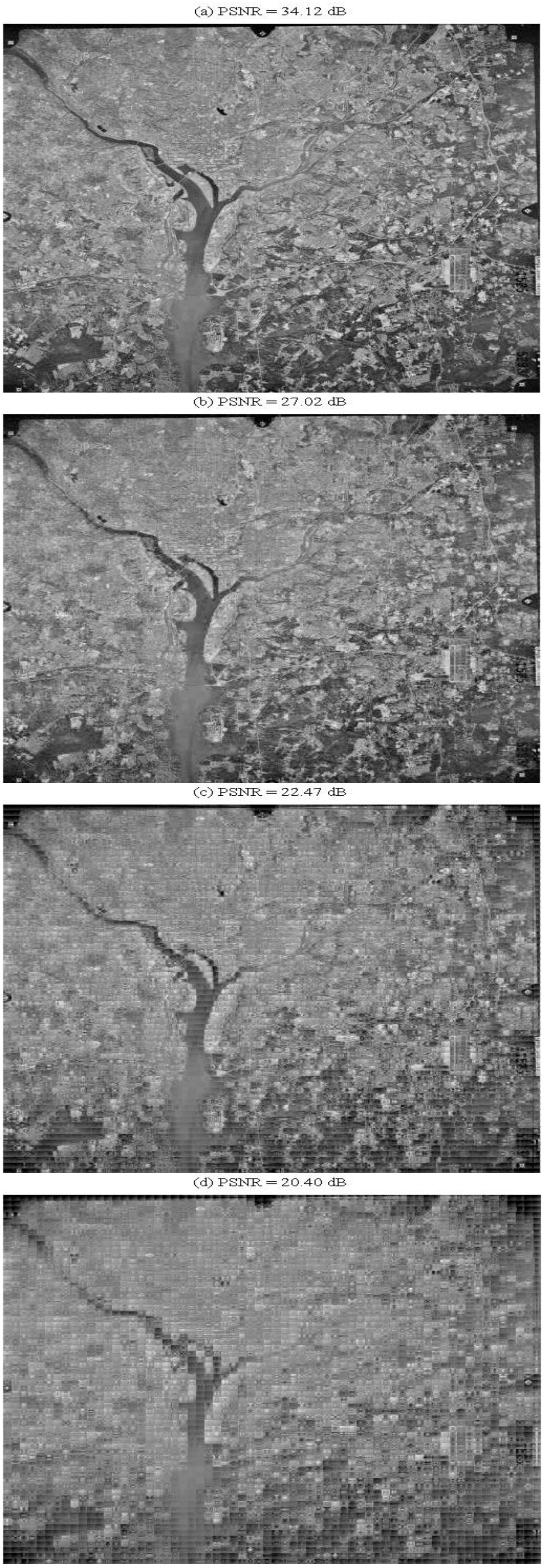
(a) PSNR = 34.12 dB

(b) PSNR = 26.50 dB

(c) PSNR = 22.78 dB

(d) PSNR = 20.96 dB

Figure 4.48: Satellite image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



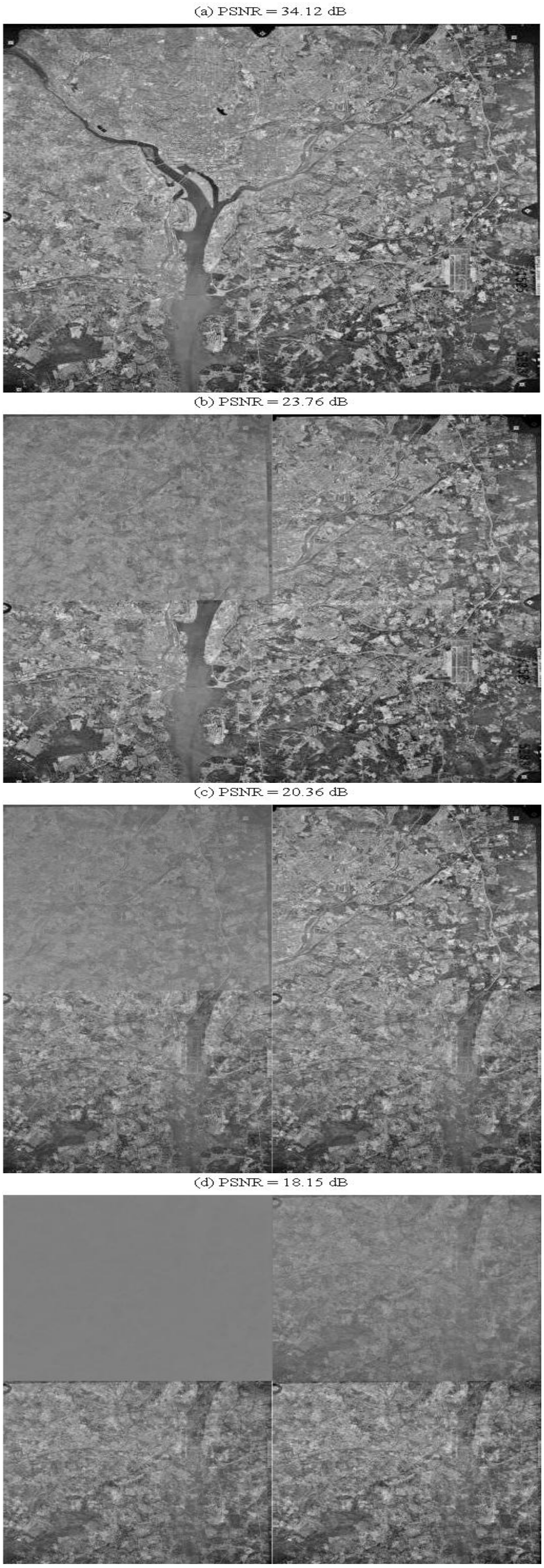
(a) PSNR = 34.12 dB

(b) PSNR = 27.02 dB

(c) PSNR = 22.47 dB

(d) PSNR = 20.40 dB

Figure 4.49: Satellite image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



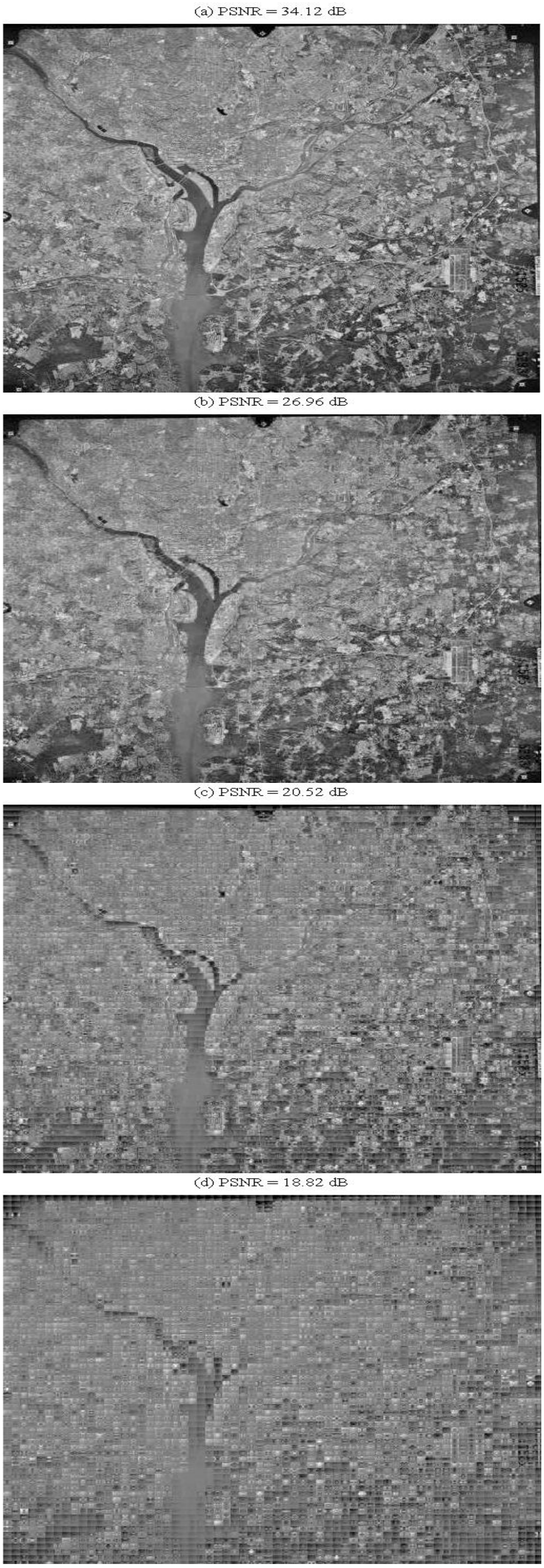
(a) PSNR = 34.12 dB

(b) PSNR = 23.76 dB

(c) PSNR = 20.36 dB

(d) PSNR = 18.15 dB

Figure 4.50: Satellite image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



(a) PSNR = 34.12 dB

(b) PSNR = 26.96 dB

(c) PSNR = 20.52 dB

(d) PSNR = 18.82 dB

Figure 4.51: Satellite image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

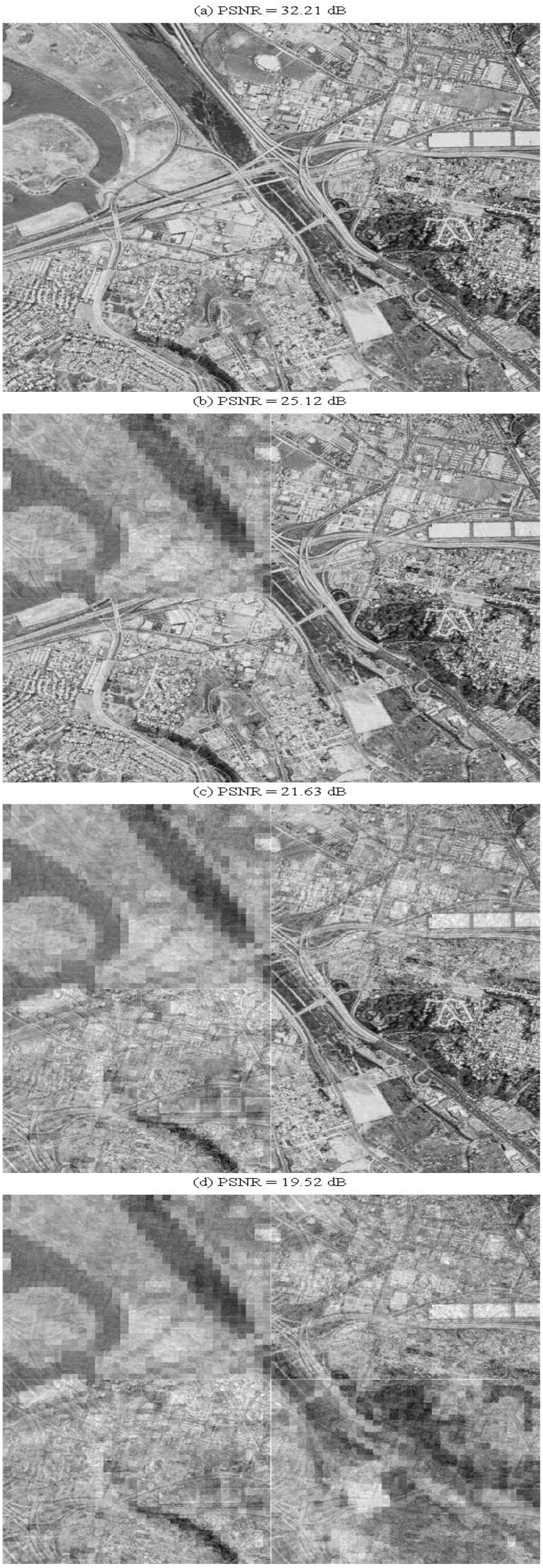


Figure 4.52: Sandiego image reconstruction results for $MDTC\ /Tec1_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

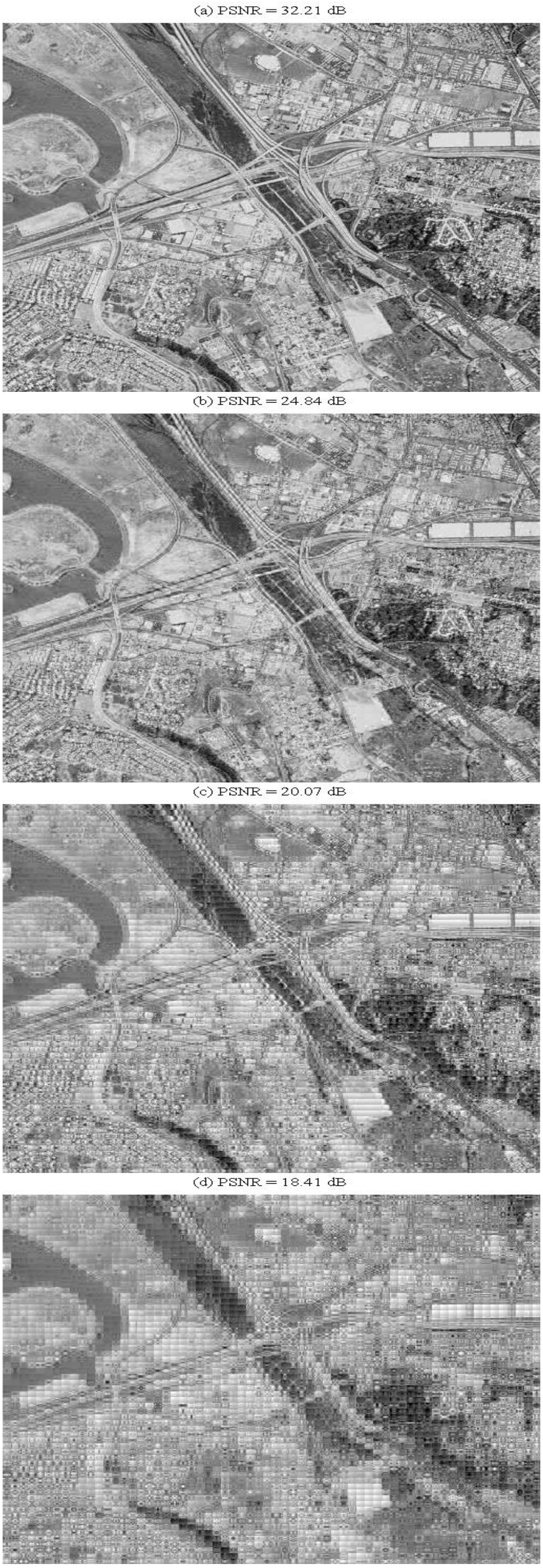


Figure 4.53: Sandiego image reconstruction results for $MDTC\ /Tec2_{DC}$, at 2 bits/sample, with the DC coefficients being communicated reliably with some other means. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

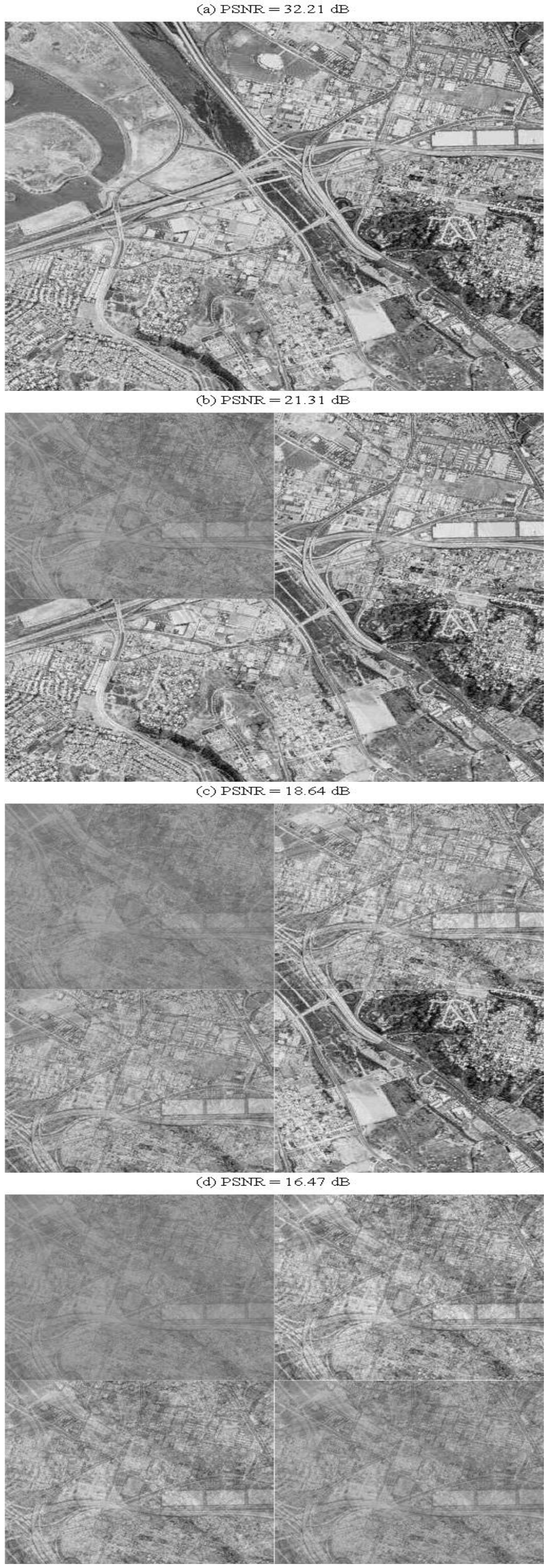


Figure 4.54: Sandiego image reconstruction results for $MDTC\ /Tec1_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

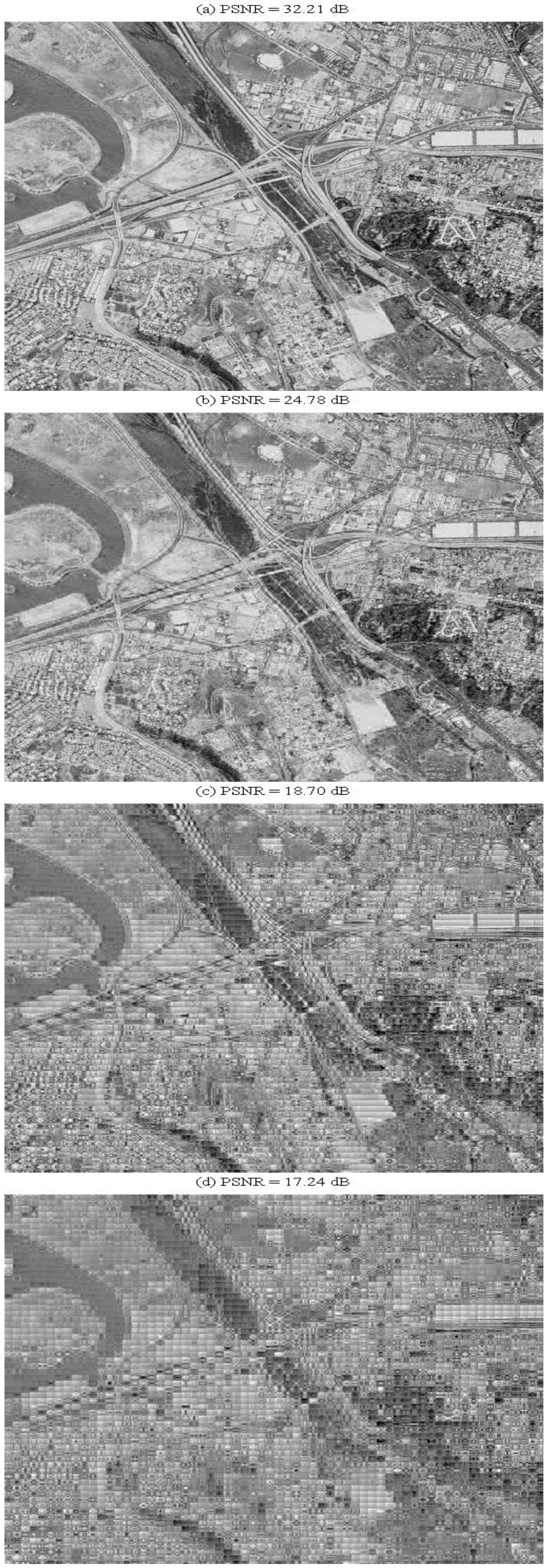


Figure 4.55: Sandiego image reconstruction results for $MDTC\ /Tec2_{\overline{DC}}$, at 2 bits/sample with the DC coefficients being communicated along with the four data streams. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

4.6.2 Image coding using wavelet-based MDTC

In this section, a new and simple MDC scheme based on the one described in [12, 13] is defined [107, 108]. In this method, we employ the discrete wavelet transform (DWT) instead of the DCT. Experimental results show that the proposed approach leads to a more graceful degradation of image quality with an increase in the loss in descriptions. Also with this technique, we do not need to consider reliable transmission of the DC components by some other means.

The discussion before assures that random variables have Gaussian distribution, which is a base condition for using correlating transform. Theoretically, Mallat [64] has proved that the histograms of wavelet transform coefficients of natural images can be modeled by a family of Gaussian distribution. So we can use MDTC within wavelet domain on a strong basis.

The wavelets have often been employed in transform image coders. Although wavelets share many properties with the DCT (e.g. decorrelation), they also allow better localization in both frequency and space [14]. As mentioned before in Section 3.4, the DWT decomposes the original spatial-domain signal into various decomposition levels that comprise a number of subbands, each of them consists of coefficients that indicate the horizontal and vertical spatial frequency characteristics of the original samples [109]. The first level decomposition, which will be used in our coder, includes four subbands, $LL1$, $HL1$, $LH1$ and $HH1$ [109]. We consider the case of four descriptions. This method is designed to operate on source vectors with uncorrelated components. Such condition is obtained by forming vectors from DCT or DWT components. We refer to DCT-based coder used in [12, 13] as MDTC/DCT coder.

As cited earlier, the implementation of the MDTC/DCT coder proceeds in the following steps:

1. The source image is transformed by an 8x8 DCT transformation.
2. The DCT coefficients are uniformly quantized.
3. The quantized DCT coefficients are split into 4 vectors (descriptions).
4. Correlating transform is applied to the 4 vectors.
5. Entropy coding is applied to each vector.

In step 3, the four vectors are formed from quantized DCT coefficients separated to the maximum in frequency and space with the DC coefficients assumed to be communicated reliably by some other means.

Our coding process based on DWT is implemented as follow [107]:

1. The source image is transformed by the 1-level biorthogonal $B_{9/7}$ wavelet transform [110] obtaining therefore the four subbands: LL1, HL1, LH1, and HH1.
2. The four vectors (descriptions) are formed: $\begin{cases} LL1 \rightarrow description1 \\ HL1 \rightarrow description2 \\ LH1 \rightarrow description3 \\ HH1 \rightarrow description4 \end{cases}$
3. The DWT coefficients (the four vectors) are uniformly quantized.
4. Correlating transform (as described in section 2) is applied to the 4 vectors.
5. Entropy coding is applied.

This technique is referred to as **MDTC/DWT.** A typical MDTC DWT-based image coder is depicted in Figure 4.56.

Redundancy of 0.1 bit/sample is evenly allocated to the four descriptions. The bit rate is estimated by sample scalar entropies. Simulation results for the nine test images, previously used in Section 4.6.1, for the two coders are given in Figures 4.57-4.65. In all figures, the average PSNR is reported as a function of the bit rate for the case of one packet dropped. We can observe that the MDTC/DWT performance increase rapidly with the bit rate and outperforms MDTC/DCT for higher bit rates, but the reconstruction quality of MDTC/DWT is worse than MDTC/DCT slightly at higher part of about 1 bit/sample.

To illustrate further the robustness of our approach for erasure channels, we have measured the PSNR of the restored images in the cases of 1, 2 and 3 packets dropped at a bit rate of 2 bits/sample.

The corresponding results are reported in Tables 4.9-4.11. We clearly see from the table that using MDTC/DWT results in an improvement in the reconstructed image quality when compared to

MDTC/DCT. For a qualitative comparison, the subjective qualities of the nine test images, for different levels of reconstruction with MDTC/DCT and MDTC/DWT coders are depicted in Figures 4.66–4.83 where Figures (a), (b), (c) and (d) illustrate respectively the situations of 0, 1, 2, and 3 packets lost. It is easily noticeable that for the wavelet transform based system the reconstruction quality degradation is graceful with the number of packets lost, and even in the case of three packet loss, the system can still provide a better and more useful reconstruction of the original image compared to the DCT-based system.

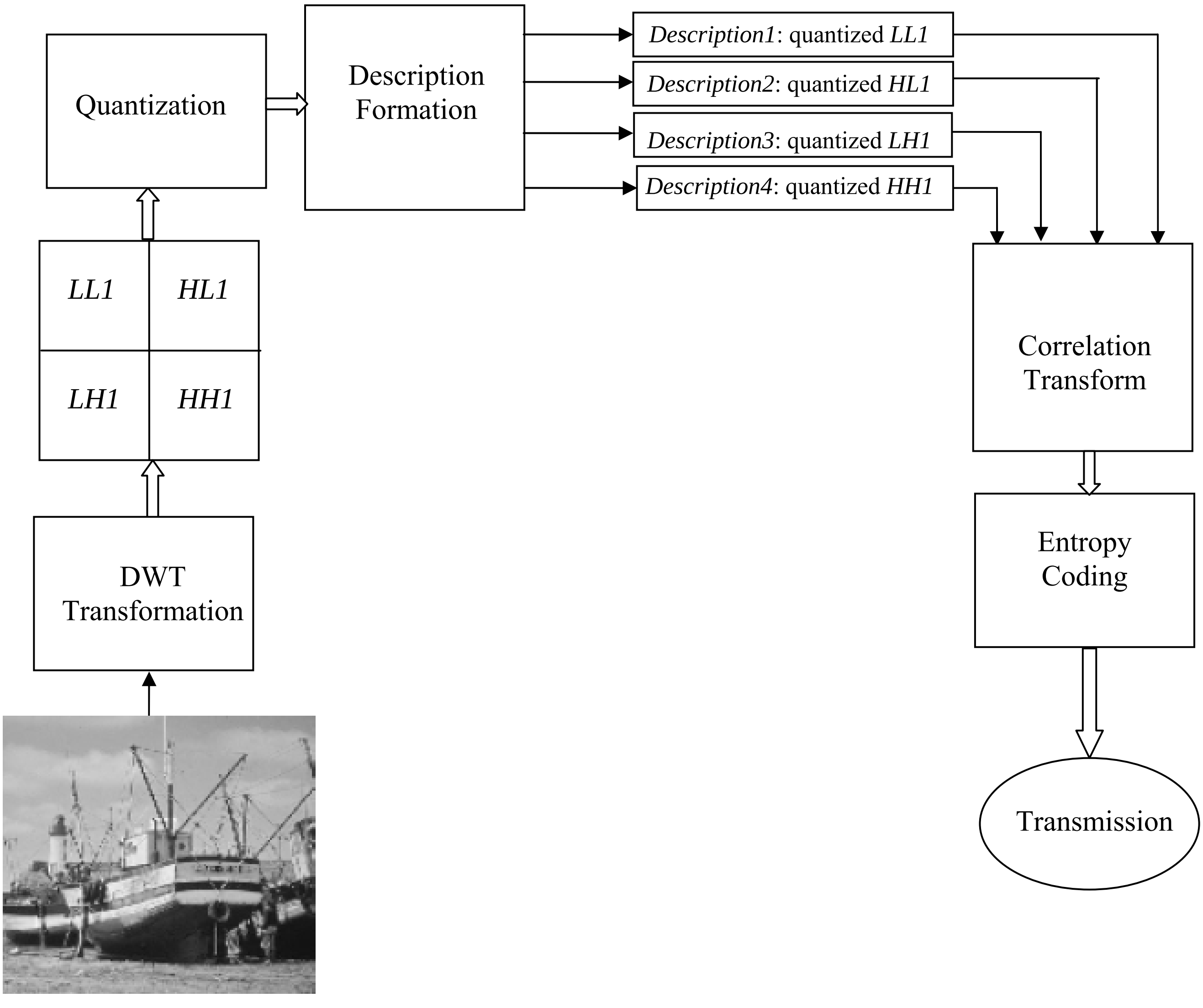

Quantization
Description Formation
Description1: quantized *LL1*
Description2: quantized *HL1*
Description3: quantized *LH1*
Description4: quantized *HH1*
LL1
HL1
LH1
HH1
Correlation Transform
Entropy Coding
DWT Transformation
Transmission

Figure 4.56: Four descriptions MDTC DWT-based image coder.

No. of	Lena image		Goldhill image		Boat image	
packets	*MDTC/DCT*	*MDTC/DWT*	*MDTC/DCT*	*MDTC/DWT*	*MDTC/DCT*	*MDTC/DWT*
dropped	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	42.50	37.23	39.43	34.64	41.94	35.53
1	31.16	36.97	27.83	34.48	28.65	35.39
2	26.55	34.04	25.38	29.40	25.22	31.81
3	25.04	31.45	24.71	27.70	23.59	28.09

Table 4.9: PSNR as a function of the number of packets lost for Lena, Goldhill and boat images.

No. of	Lighthouse image		House image		Nitf7 image	
packets	*MDTC/DCT*	*MDTC/DWT*	*MDTC/DCT*	*MDTC/DWT*	*MDTC/DCT*	*MDTC/DWT*
dropped	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	39.93	32.75	37.34	29.91	31.42	26.88
1	24.95	32.63	22.52	29.67	21.64	25.79
2	21.22	24.55	19.93	21.58	18.00	18.65
3	20.65	22.40	17.79	19.80	16.13	17.04

Table 4.10: PSNR as a function of the number of packets lost for Lighthouse, House and Nitf7 images.

No. of	Baboon image		Satellite image		Sandiego image	
packets	*MDTC/DCT*	*MDTC/DWT*	*MDTC/DCT*	*MDTC/DWT*	*MDTC/DCT*	*MDTC/DWT*
dropped	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)	PSNR (dB)
0	31.42	28.12	34.12	31.21	32.21	29.16
1	25.26	26.44	26.50	30.55	25.12	28.40
2	22.13	22.38	22.71	23.73	21.48	21.94
3	20.38	20.17	20.96	22.14	19.38	20.20

Table 4.11: PSNR as a function of the number of packets lost for Baboon, Satellite and Sandiego images.

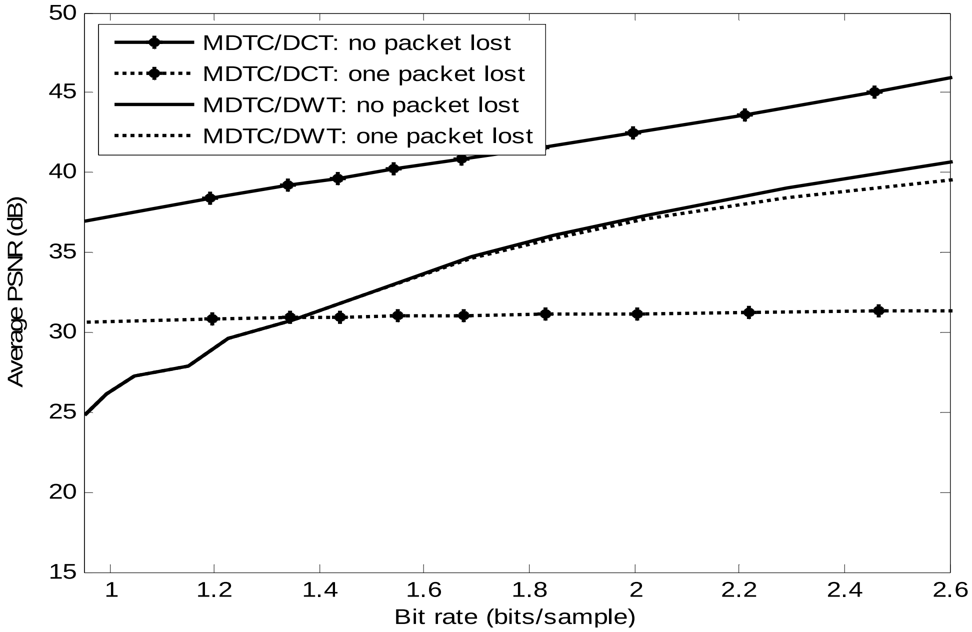


Figure 4.57: Average PSNR versus bits per sample, ‘Lena’ image.

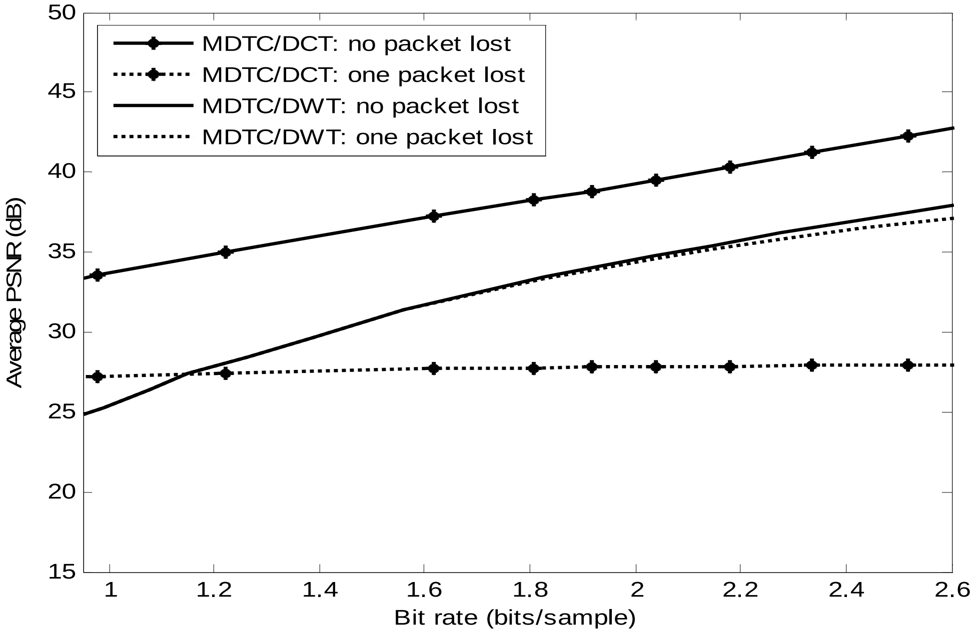


Figure 4.58: Average PSNR versus bits per sample, ‘Goldhill’ image.

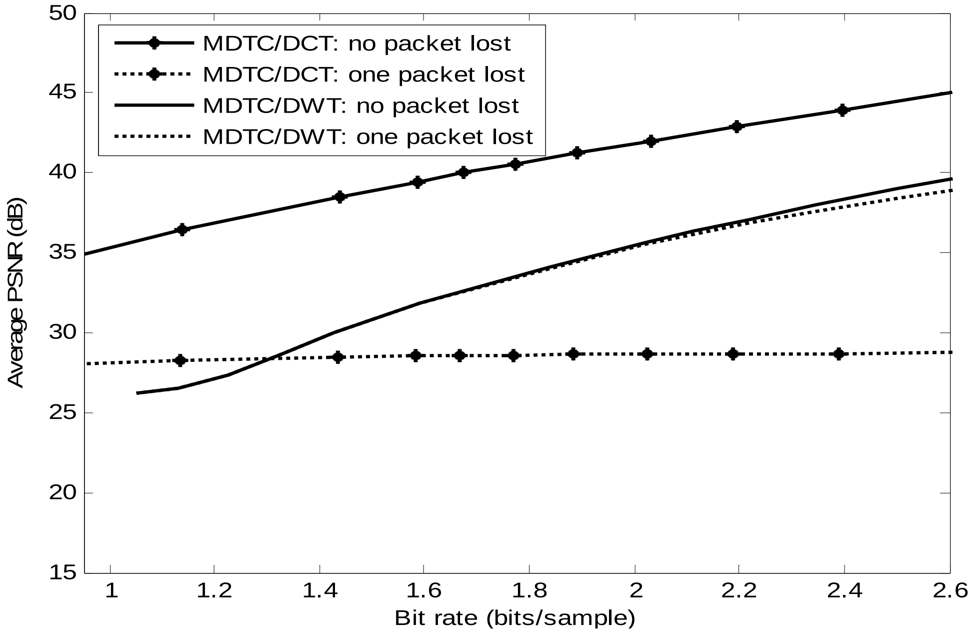


Figure 4.59: Average PSNR versus bits per sample, ‘Boat’ image.

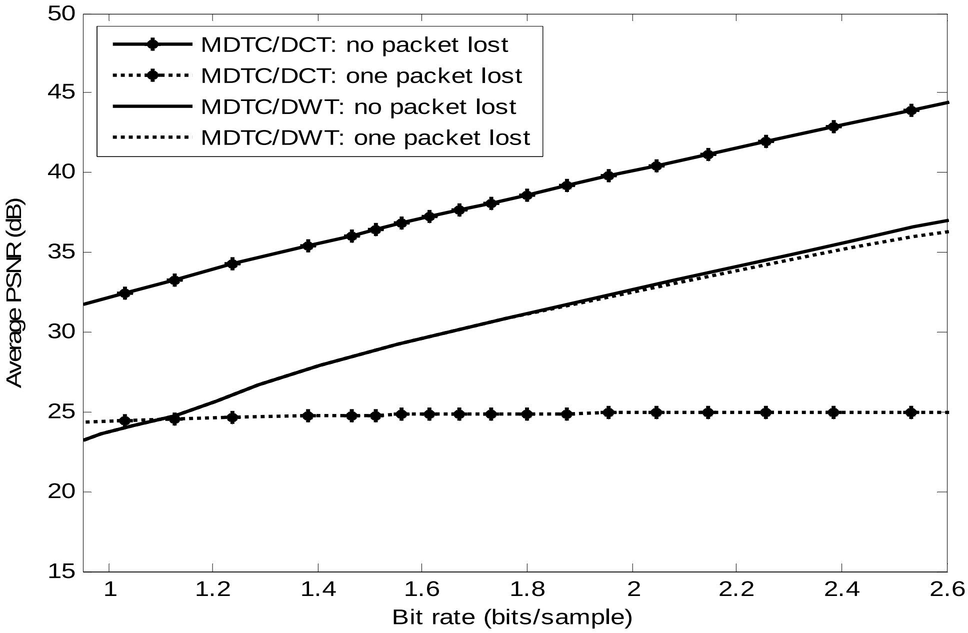


Figure 4.60: Average PSNR versus bits per sample, ‘Lighthouse’ image.

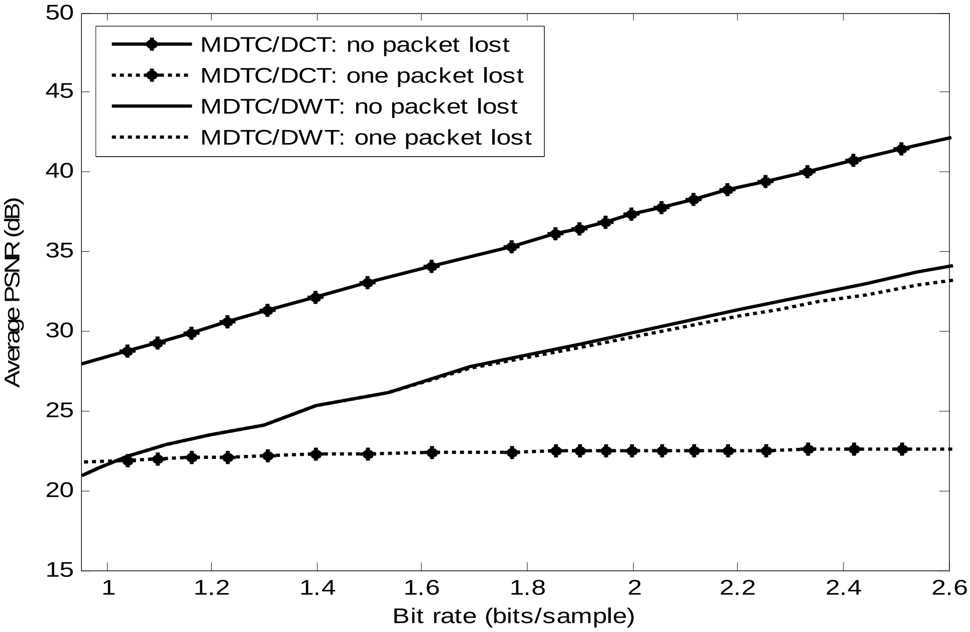


Figure 4.61: Average PSNR versus bits per sample, ‘House’ image.

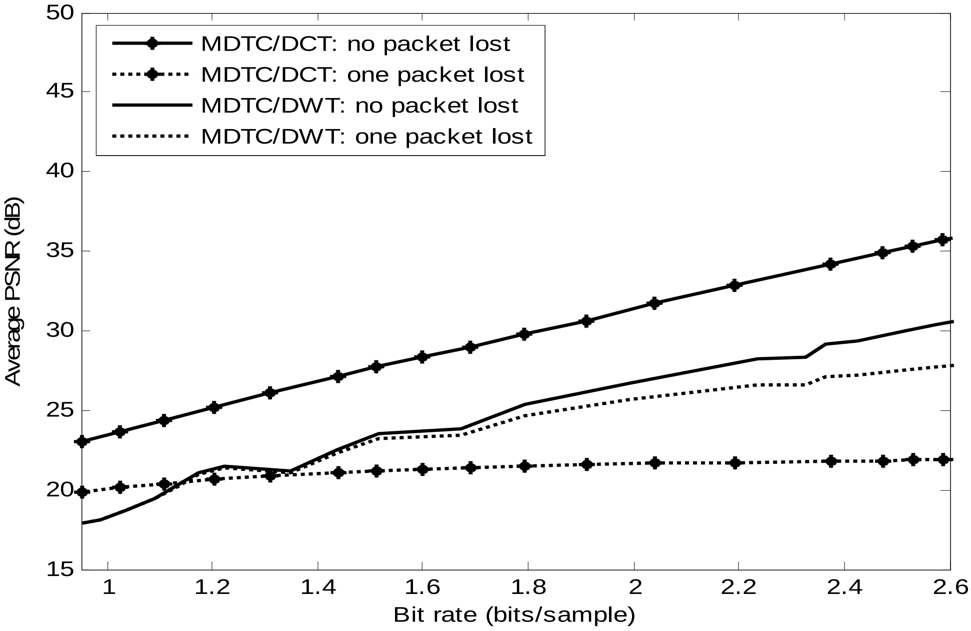


Figure 4.62: Average PSNR versus bits per sample, ‘Nitf7’ image.

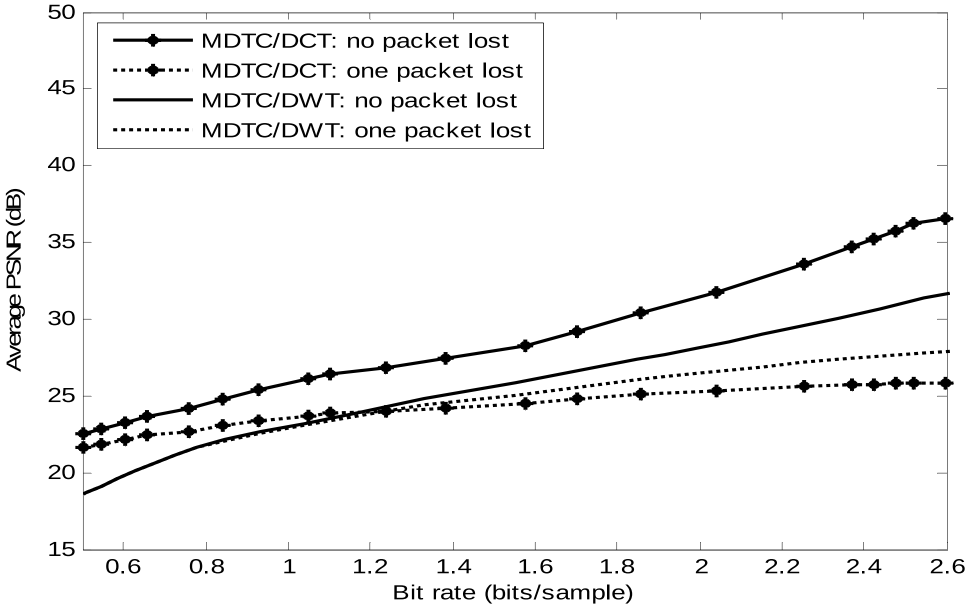


Figure 4.63: Average PSNR versus bits per sample, ‘Baboon’ image.

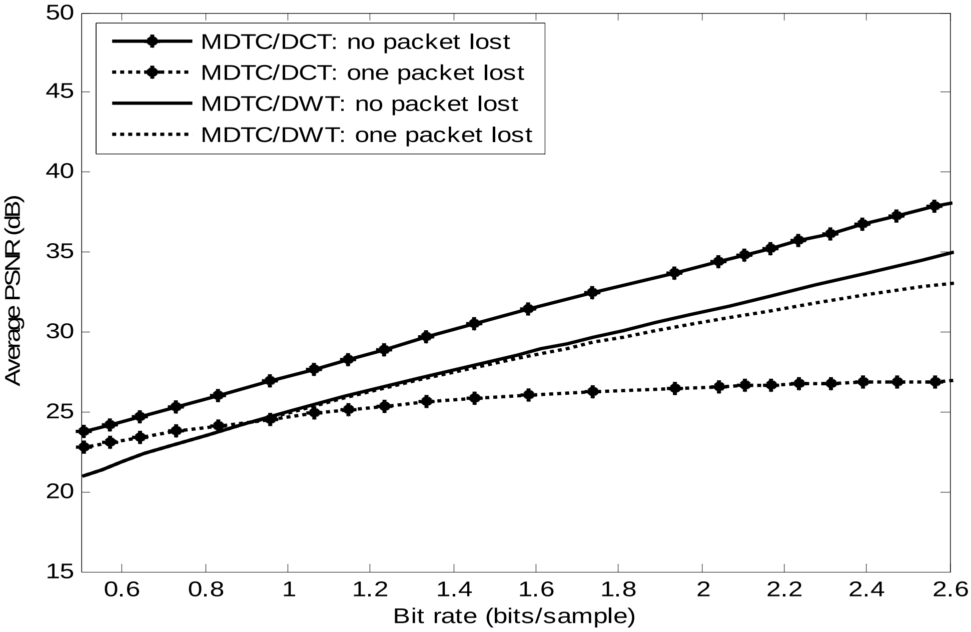


Figure 4.64: Average PSNR versus bits per sample, ‘Satellite’ image.

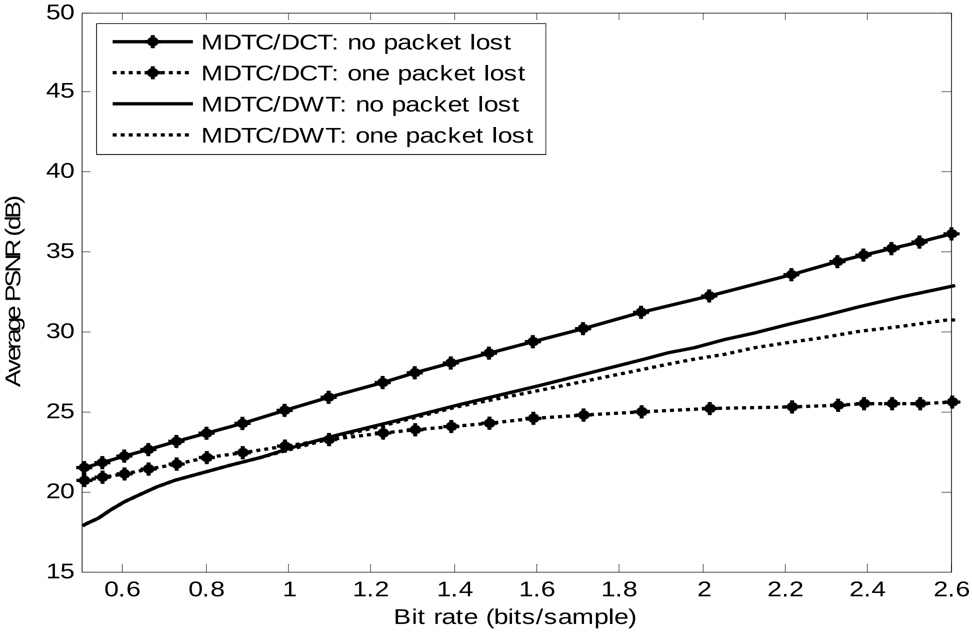
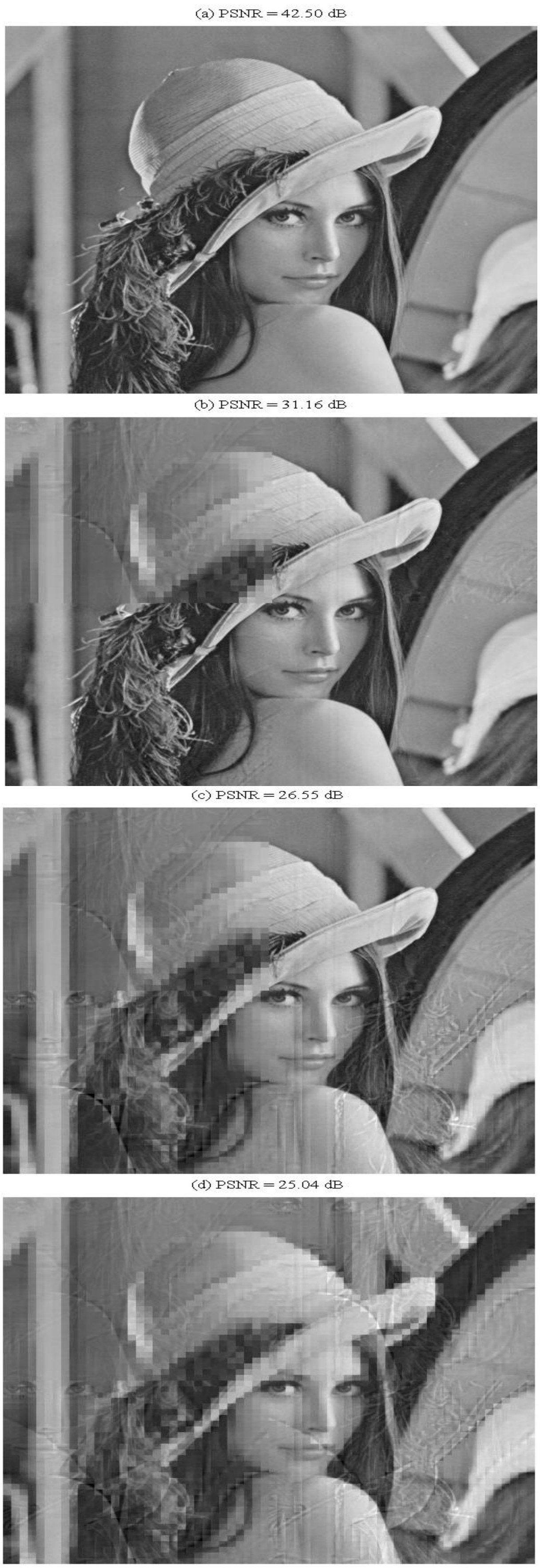


Figure 4.65: Average PSNR versus bits per sample, ‘Sandiego’ image.



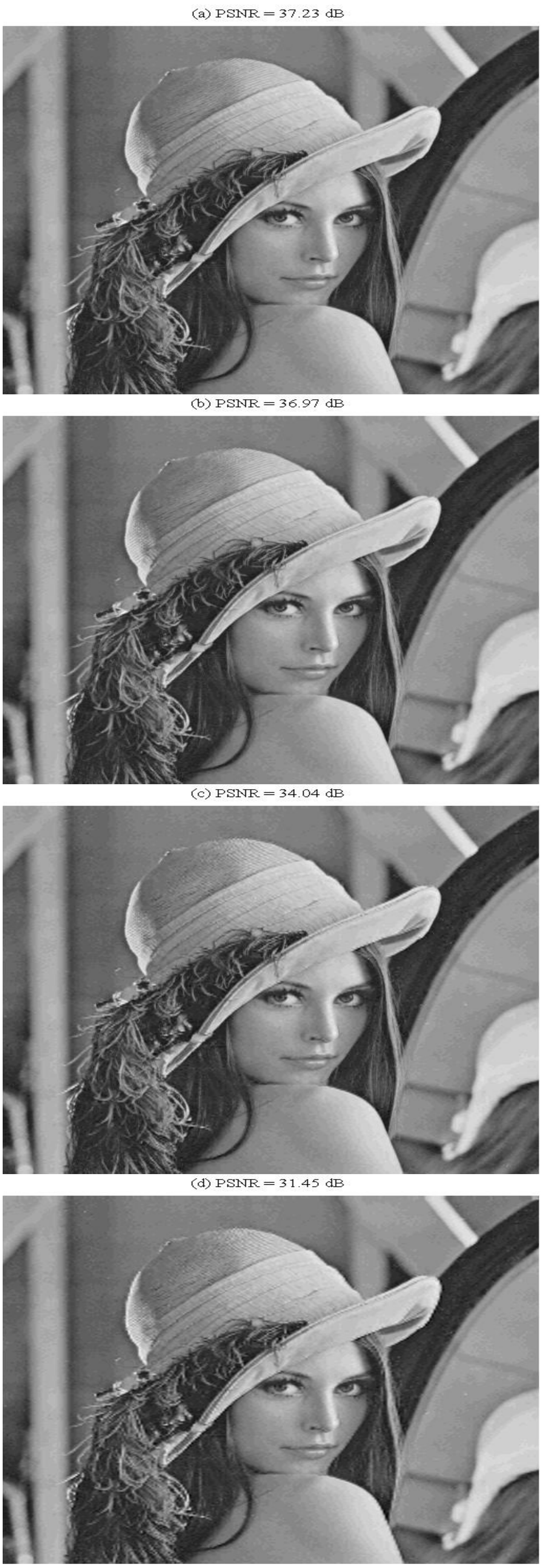
(a) PSNR = 42.50 dB

(b) PSNR = 31.16 dB

(c) PSNR = 26.55 dB

(d) PSNR = 25.04 dB

Figure 4.66: Lena image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



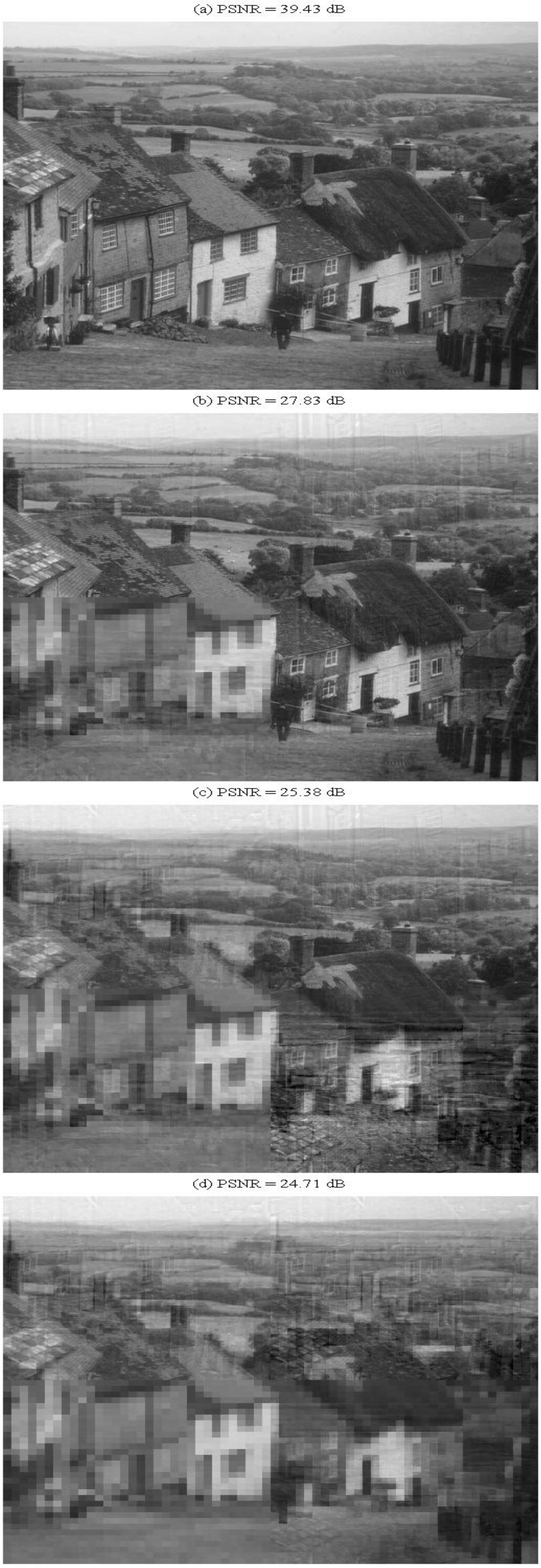
(a) PSNR = 37.23 dB

(b) PSNR = 36.97 dB

(c) PSNR = 34.04 dB

(d) PSNR = 31.45 dB

Figure 4.67: Lena image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



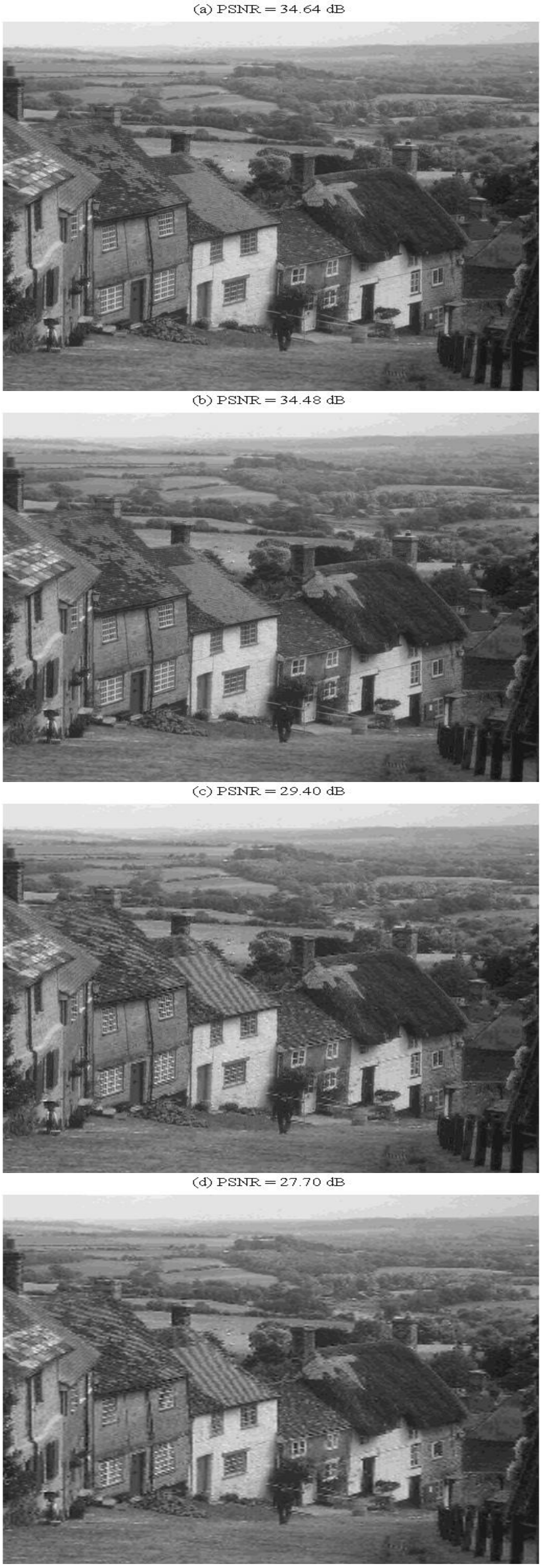
(a) PSNR = 39.43 dB

(b) PSNR = 27.83 dB

(c) PSNR = 25.38 dB

(d) PSNR = 24.71 dB

Figure 4.68: Goldhill image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



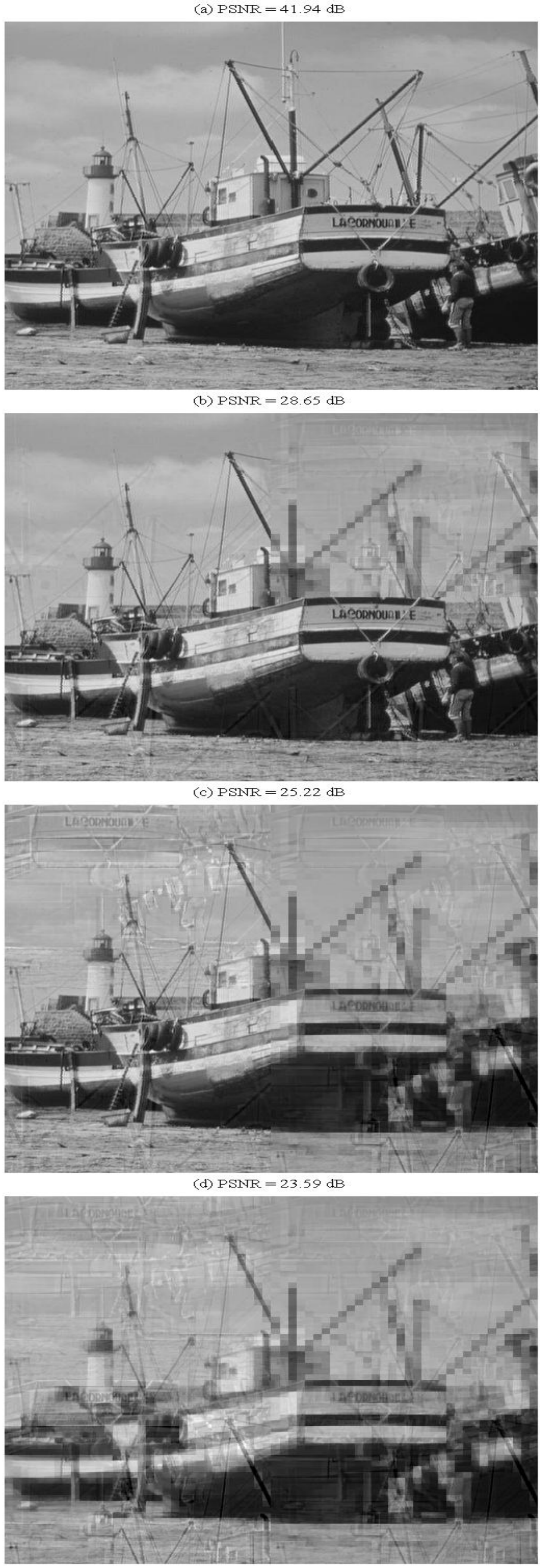
(a) PSNR = 34.64 dB

(b) PSNR = 34.48 dB

(c) PSNR = 29.40 dB

(d) PSNR = 27.70 dB

Figure 4.69: Goldhill image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



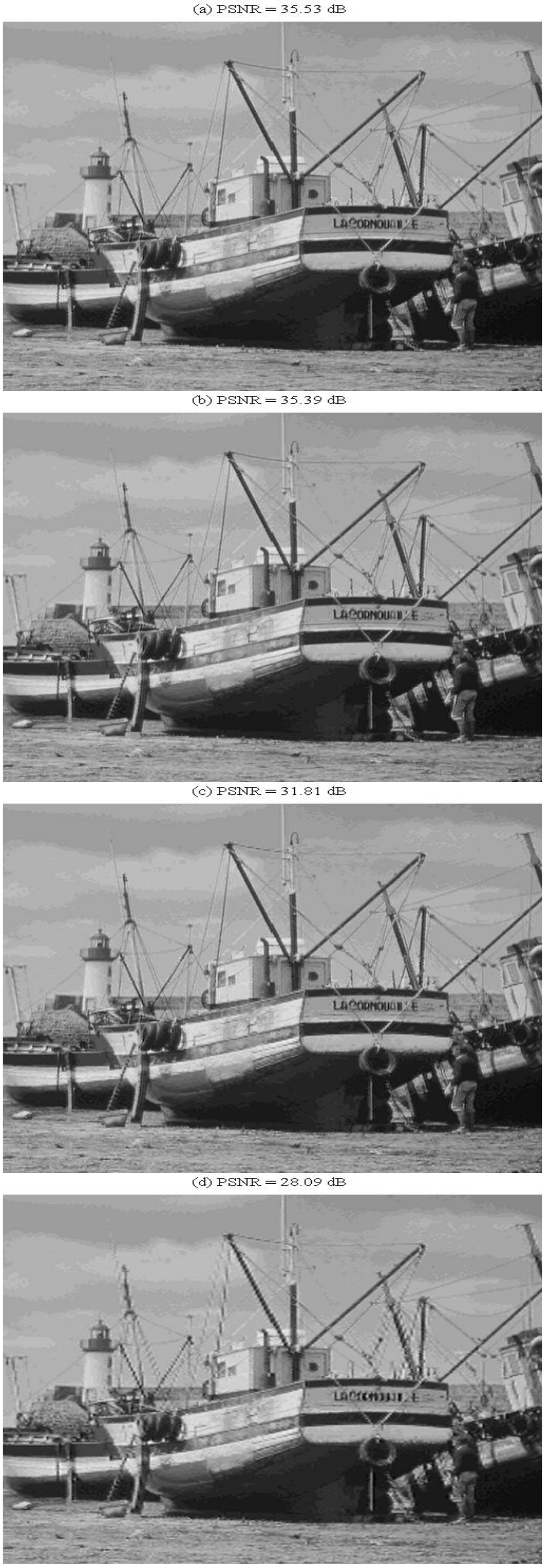
(a) PSNR = 41.94 dB

(b) PSNR = 28.65 dB

(c) PSNR = 25.22 dB

(d) PSNR = 23.59 dB

Figure 4.70: Boat image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



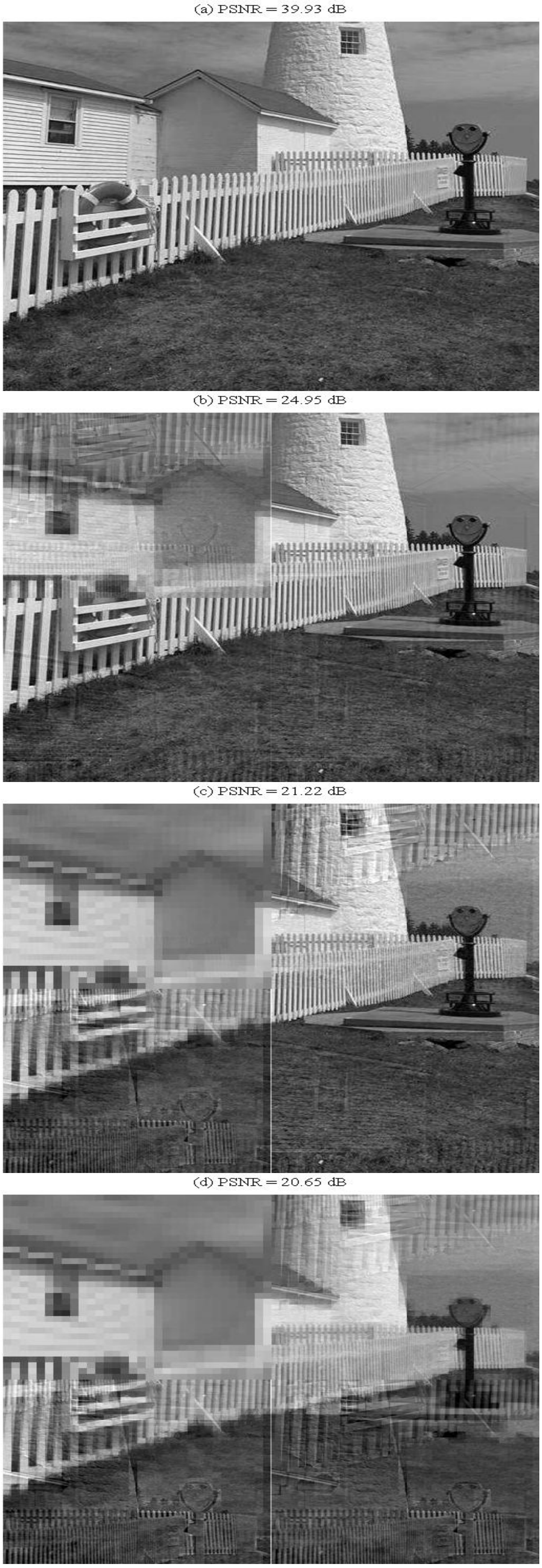
(a) PSNR = 35.53 dB

(b) PSNR = 35.39 dB

(c) PSNR = 31.81 dB

(d) PSNR = 28.09 dB

Figure 4.71: Boat image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



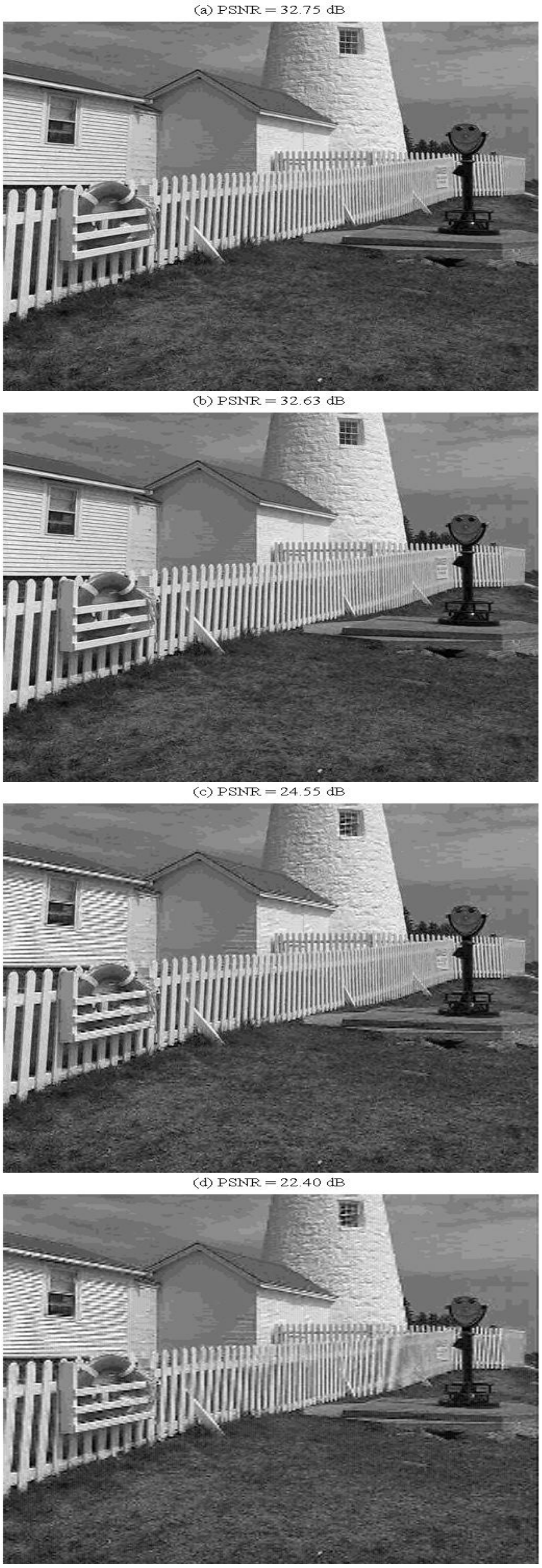
(a) PSNR = 39.93 dB

(b) PSNR = 24.95 dB

(c) PSNR = 21.22 dB

(d) PSNR = 20.65 dB

Figure 4.72: Lighthouse image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



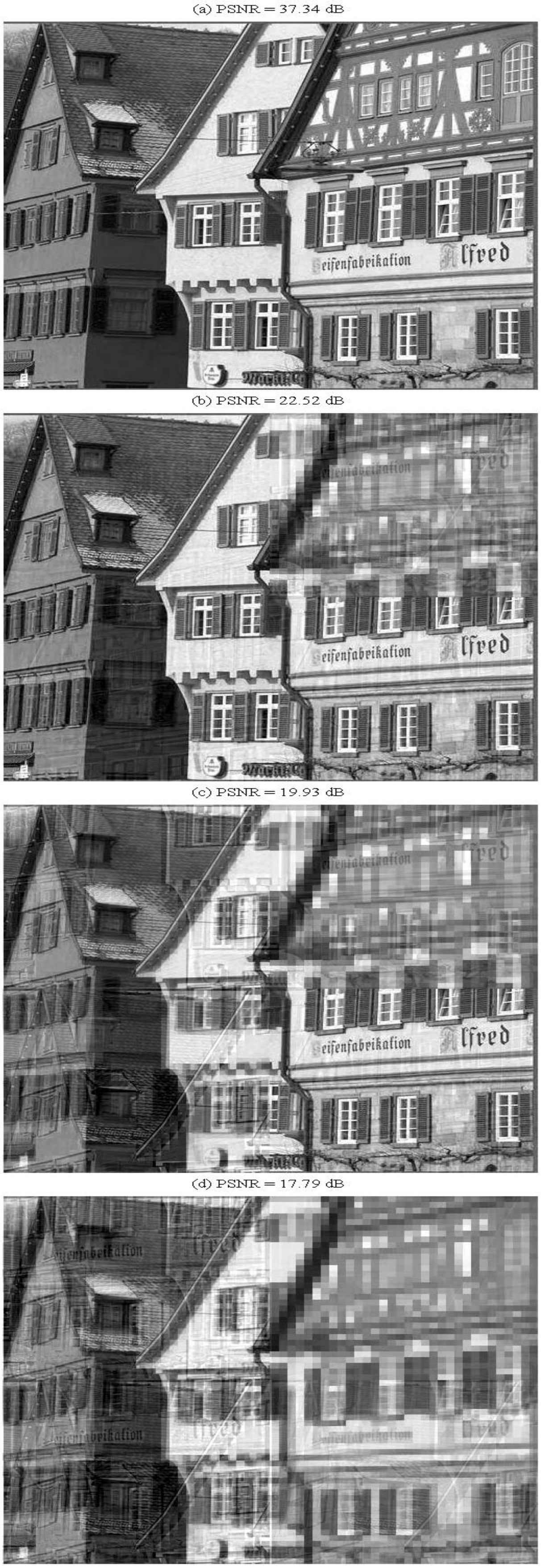
(a) PSNR = 32.75 dB

(b) PSNR = 32.63 dB

(c) PSNR = 24.55 dB

(d) PSNR = 22.40 dB

Figure 4.73: Lighthouse image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



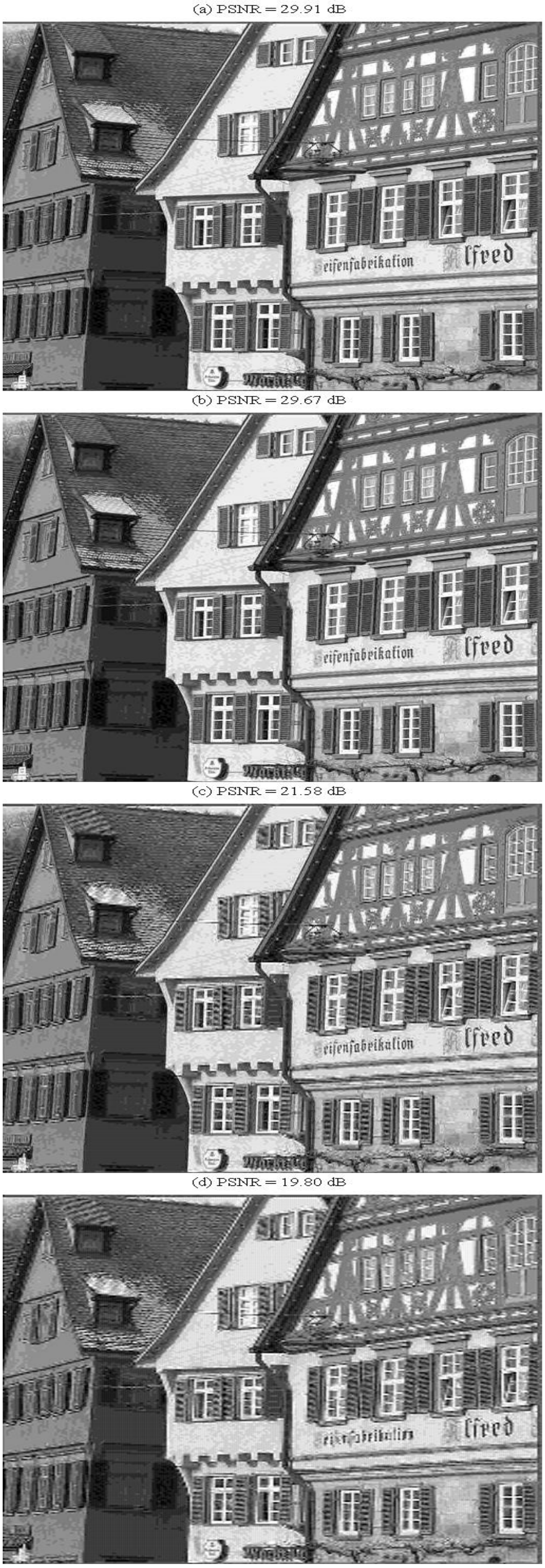
(a) PSNR = 37.34 dB

(b) PSNR = 22.52 dB

(c) PSNR = 19.93 dB

(d) PSNR = 17.79 dB

Figure 4.74: House image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



(a) PSNR = 29.91 dB

(b) PSNR = 29.67 dB

(c) PSNR = 21.58 dB

(d) PSNR = 19.80 dB

Figure 4.75: House image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

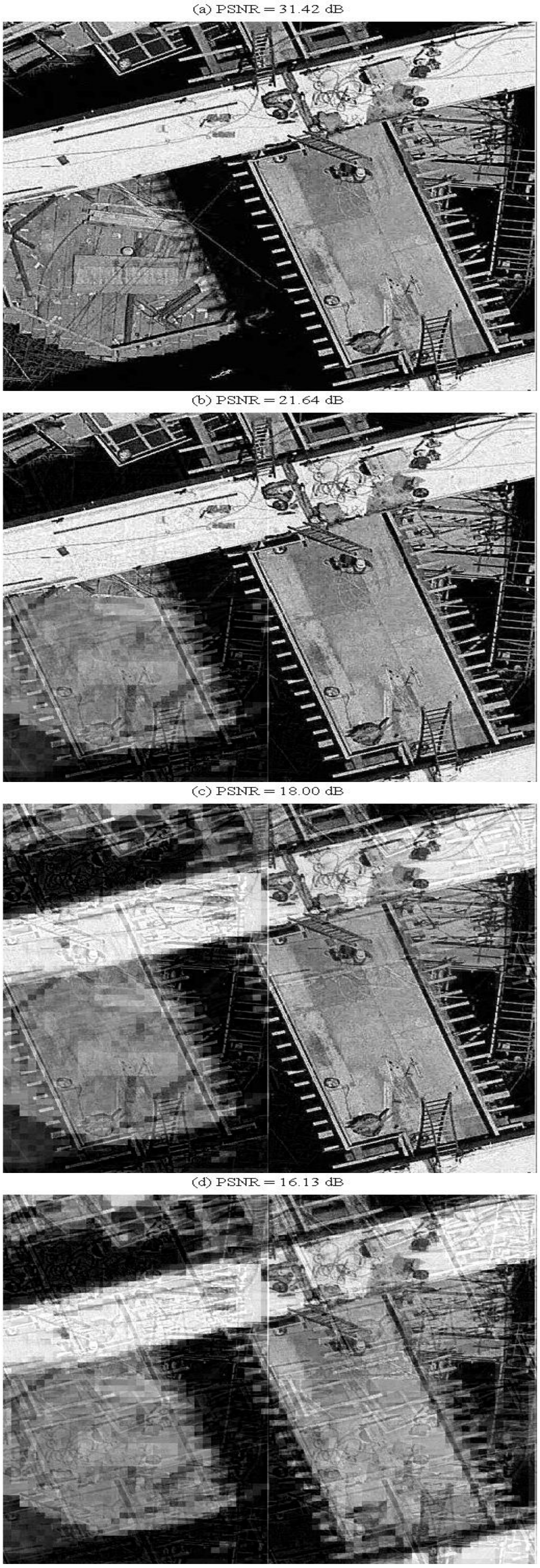


Figure 4.76: Nitf7 image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

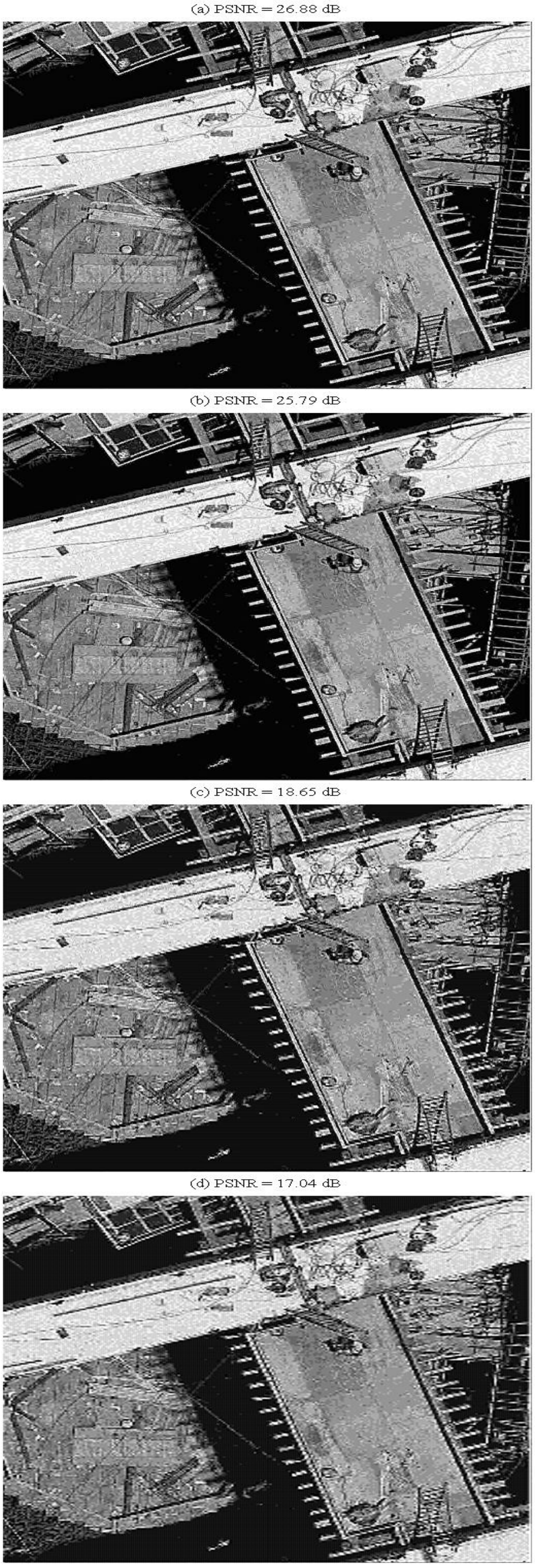
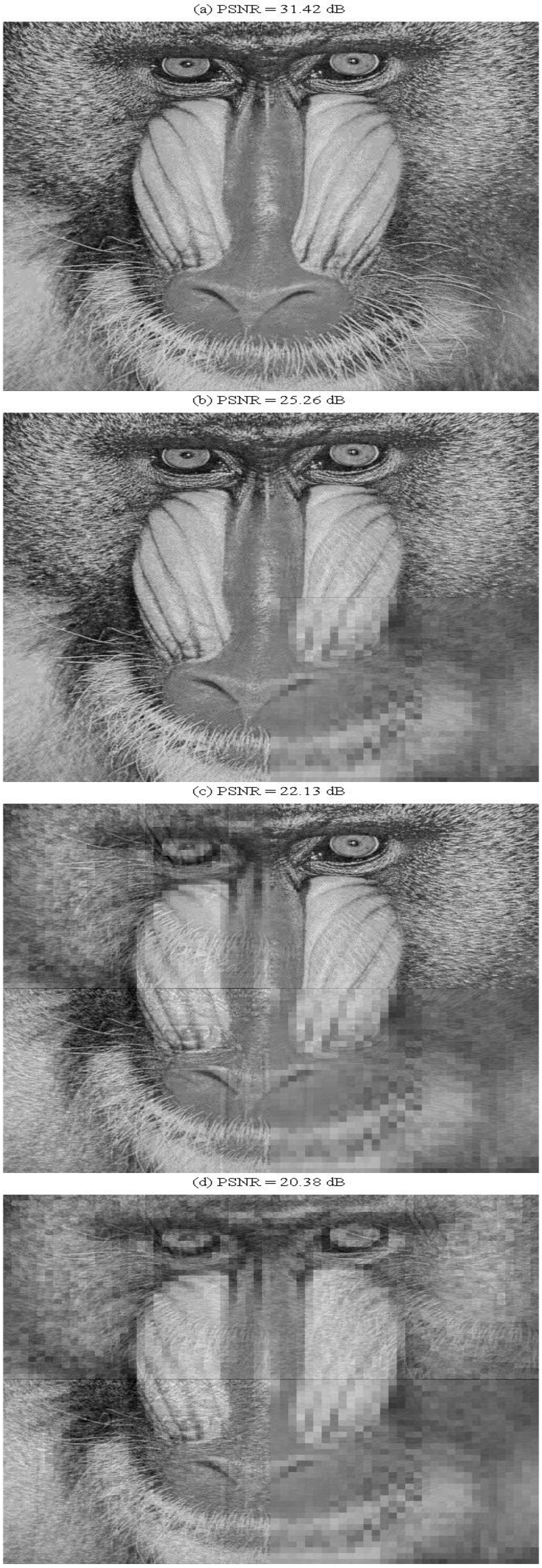


Figure 4.77: Nitf7 image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



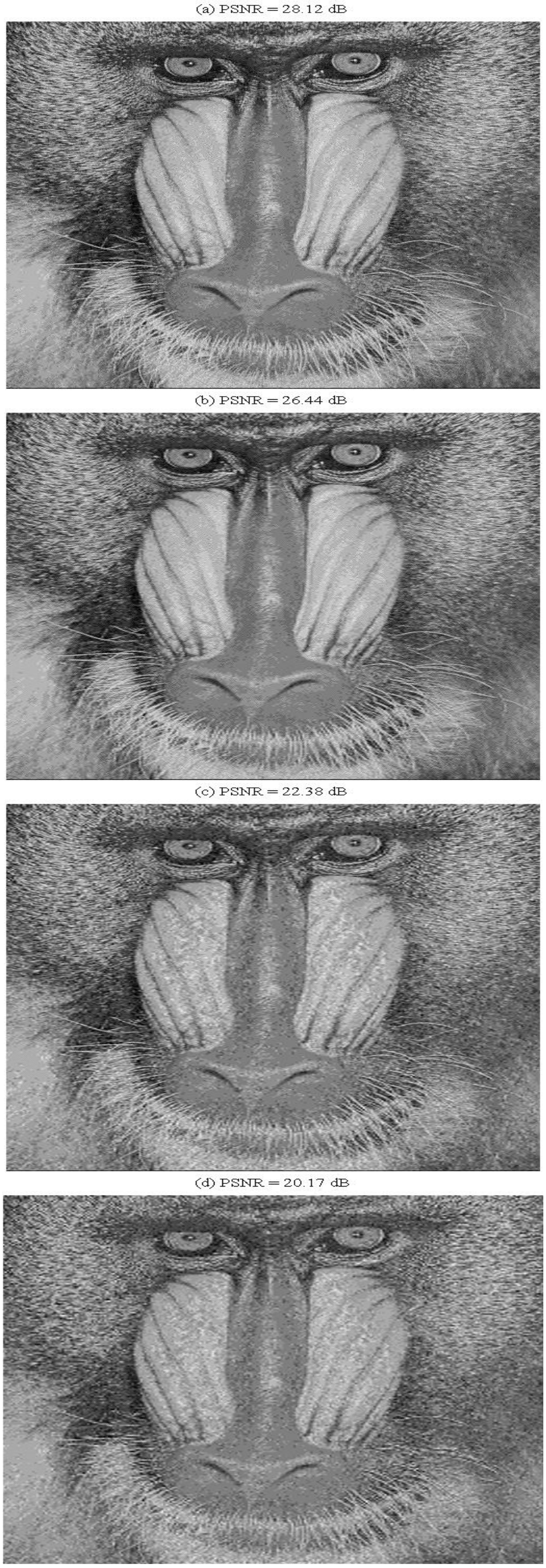
(a) PSNR = 31.42 dB

(b) PSNR = 25.26 dB

(c) PSNR = 22.13 dB

(d) PSNR = 20.38 dB

Figure 4.78: Baboon image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.



(a) PSNR = 28.12 dB

(b) PSNR = 26.44 dB

(c) PSNR = 22.38 dB

(d) PSNR = 20.17 dB

Figure 4.79: Baboon image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

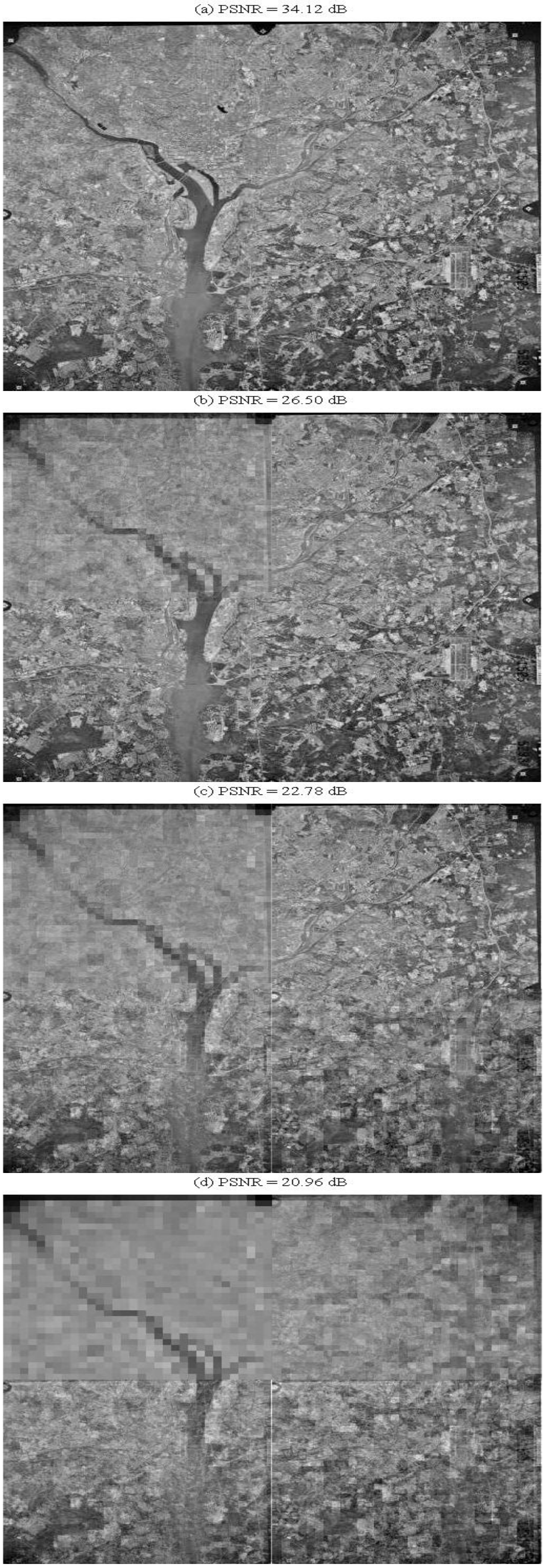


Figure 4.80: Satellite image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

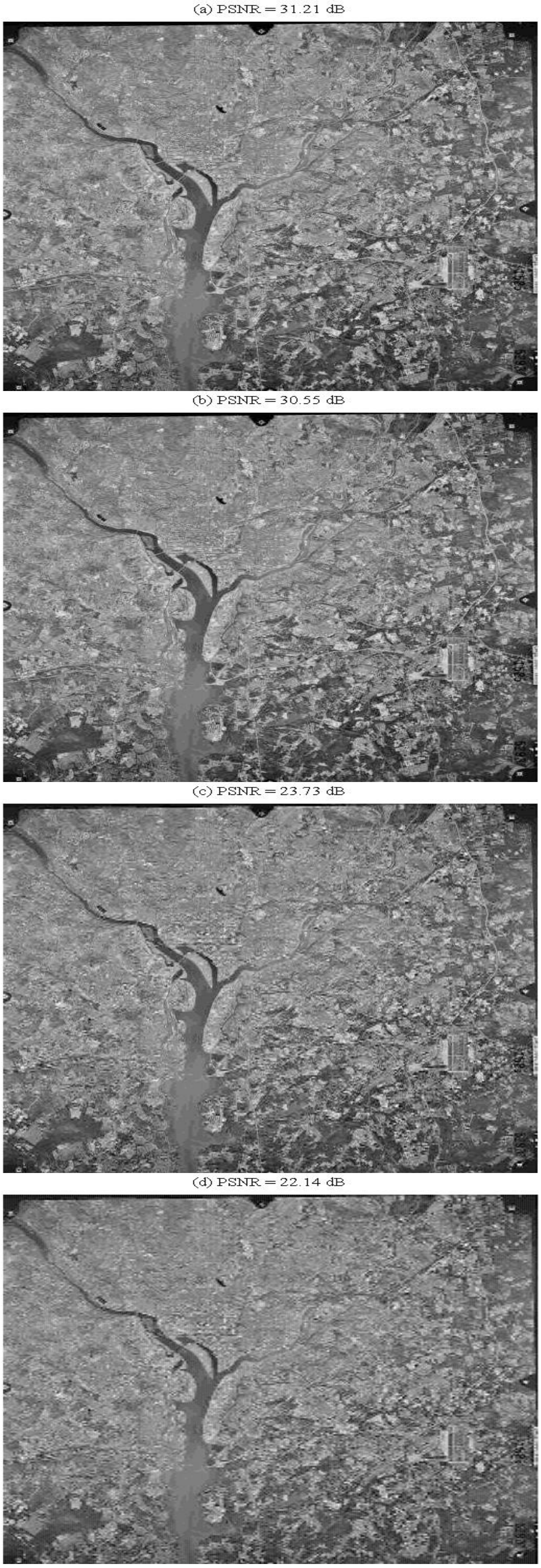


Figure 4.81: Satellite image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

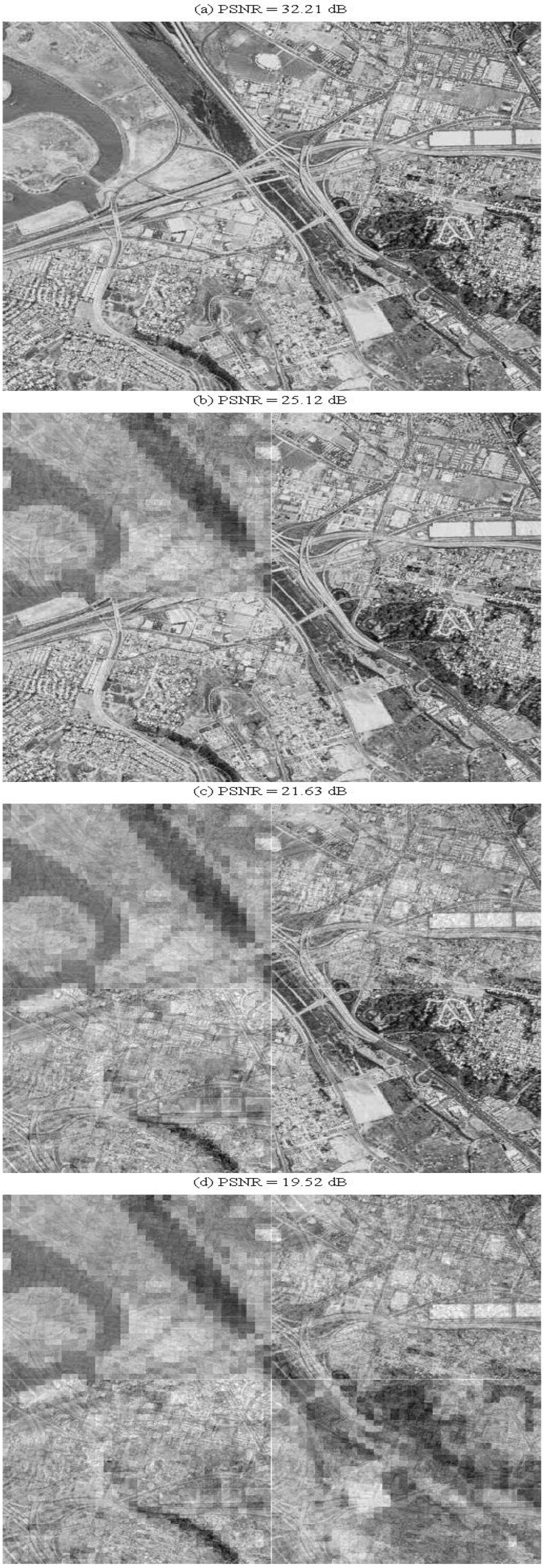


Figure 4.82: Sandiego image reconstruction results for *MDTC/DCT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

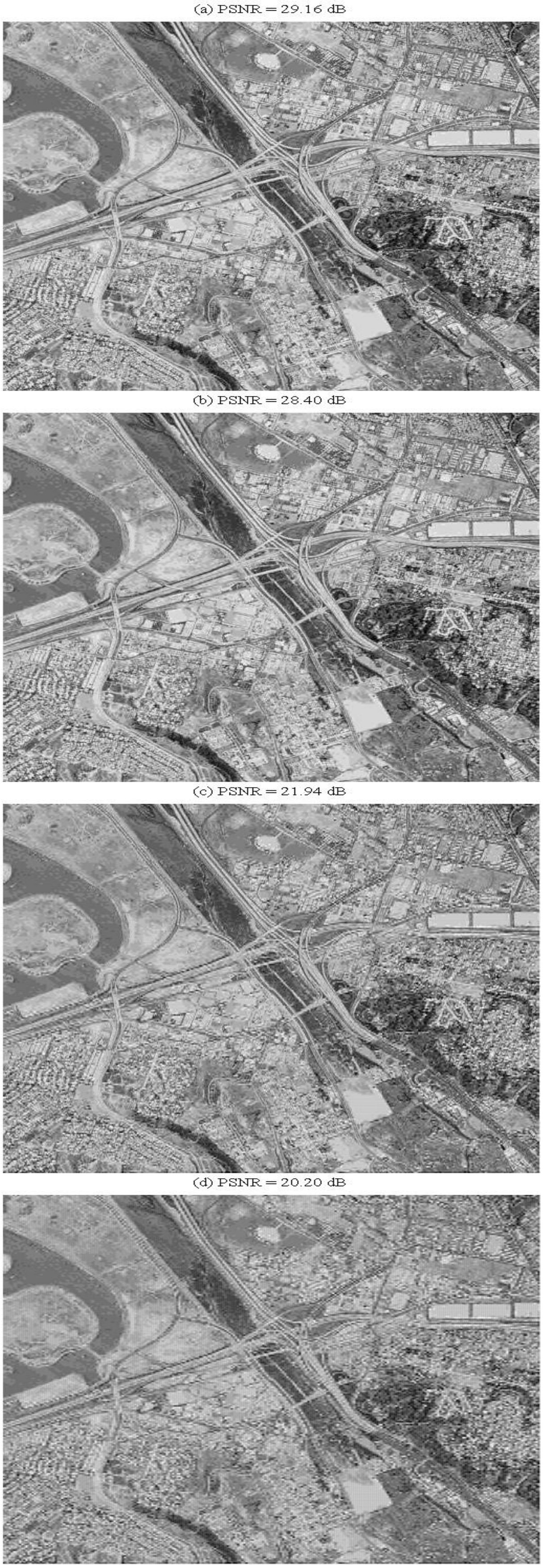


Figure 4.83: Sandiego image reconstruction results for *MDTC/DWT*, at 2 bits/sample. (a) No packet is lost; (b) 1 packet is lost; (c) 2 packets are lost; (d) 3 packets are lost.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Thesis summary

In this thesis we have investigated the problem of transmitting image data over heterogeneous packet erasure channels using Multiple Description Transform Coding (MDTC).

Firstly, based on the work of Goyal et al. [9, 12], we have proposed a different DCT based MDTC image coder for the case of four descriptions (packets). The difference relies mainly on the way the four packets are formed. The performance has been evaluated for 9 ***512x512*** greyscale test images grouped into three image types: three low frequency (Lena, Boat, Goldhill), three medium frequency (Lighthouse, Nitf7, House), and three high frequency (Satellite, Mandrill, SanDiego) images. Simulation results have shown that our employed technique leads to an improvement in performance in terms of rate/distortion. In addition, using the proposed scheme, the DC coefficients obtained after the application of DCT need not be communicated reliably by some other means as suggested by Goyal et al in their MDTC coder.

In a second part, we have proposed a new simple wavelet based MDTC image coder for the case of four packets. The considered scheme uses the wavelet transform instead of the DCT transform. We have carried out a comparative study of DCT- and DWT-based coding on the nine previously mentioned test images. Our proposed approach shows to be more robust when transmitting images through unreliable networks. Through some experiments, we conclude that even if the wavelet-based MDTC system receives only one description (the three other descriptions being lost), it can still restore image with better reconstruction quality with respect to DCT-based MDTC system. Also, the same significant superiority in reconstruction quality applies if two descriptions are lost (and with a lower difference if only one is lost).

5.2 Future directions

Further work may include:

- The coding scheme presented here is based on high-rate entropy estimates for uniformly quantized Gaussian random variables. Further investigations could be done, including the effect of non-uniform quantization and the divergence from Gaussianity.
- One interesting problem is to study the possibility of incorporating, in the proposed scheme, the run length coding and Huffman coding to get a JPEG like coder.
- In order to be well adapted to image transmission on Internet, the proposed 4 descriptions MDTC/DWT image coder can be extended to the case of 8 descriptions by designing the corresponding correlating transform.
- Another interesting question in a future is to extend this work to handle video data transmission by developing the necessary algorithms.

REFERENCES

[1] Goyal, V. K., "Multiple description coding: Compression meets the network," IEEE Signal Processing Magazine 18, 74-93, 2001.

[2] C. Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," IEEE Journal Selected Areas on Communications, 17(5): 1-17, may 1999.

[3] Q. Chen and T. Fisher, "Image coding using robust quantization for noisy digital transmission," IEEE Transaction on Image Processing 7, 496–505, 1998

[4] J. Garcia-Frias and J. Villasenor, "An analytical treatment of channel-induced distortion in entropy coded image subbands," in: Data Compression Conference, Snowbird, UT, 1997.

[5] D. Redmill and N. Kingsbury, "Still image coding for noisy channels," in: International Conference on Image Processing, Austin, TX, 94, pp. 95–99, 1994.

[6] P. Sherwood and K. Zeger, "Progressive image coding for noisy channels," IEEE Signal Processing Letters, 4:189–191, 1997.

[7] P. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," IEEE Transactions on Communication, 46 (12), 1555–1559, December 1998.

[8] N. Tanabe and N. Farvardin, "Subband image coding using entropy coded quantization over noisy channels," IEEE Journal Selected Areas on Communication, 10:926–943, 1992.

[9] C. Shannon, "A mathematical theory of communication," The Bell System Technical Journal, vol. 27, pp. 379–423, 623–656, July, Octobre 1948.

[10] V. Chande, N. Farvardin and H. Jafarkhani, "Image communication over noisy channels with feedback," in: Data Compression Conference, Snowbird, UT, 1999.

[11] El Gamal and T. Cover, "Achievable rates for multiple descriptions," IEEE Transactions on Information Theory, vol.IT-28, no.6, pp.851–57, November1982.

[12] V. K. Goyal and J. Kovacevic, "Generalized multiple description coding with correlating transforms," IEEE Transactions on Information Theory, 47: 2199–2224, 2001.

[13] V. K. Goyal, J. Kovacevic, R. Arean and M. Vetterli, "Multiple description transform coding of images," International Conference on Image Processing, Chicago, Illinois, 674-678, USA 1998.

[14] S. Lin and D. J. Costello, "Error control coding, Fundamentals and applications," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[15] T. M. Cover and J. A. Thomas, "Elements of Information Theory," John Wiley & Sons, New York, 1991.

[16] A. Gersho and R. Gray, "Vector Quantization and Signal Compression," Kluwer Academic

Publishers, Boston, MA 1992.

[17] J. G. Proakis, "Digital Communications," McGraw-Hill, New York City, New York, USA, Fourth Edition, 2001.

[18] M. Nelson, "Data compression with the burrows-wheeler transform," Dr. Dobb's Journal, 1996.

[19] A. H. Reeves. French Patent 852 183, October 23, 1939; U.S. Patent 2 272 070, February 3, 1942.

[20] J. R. Pierce, "The early days of information theory," IEEE Transactions on Information Theory, IT-19(1):3:8, January 1973.

[21] H. S. Black and E. O. Edson, "PCM equipment," Elec. Eng., 66:1123:1125, November 1947.

[22] S. P. Lloyd, "Least squares quantization in PCM," IEEE Transactions on Information Theory, IT-28(2):129-137, March 1982. Originally an unpublished Bell Telephone Laboratories tech. memo., 1957.

[23] J. Max, "Quantizing for minimum distortion," IRE Transactions on Information Theory, IT-6(1):7-2, March 1960.

[24] V. K. Goyal, "Beyond traditional transform coding," Ph. D. thesis, University of California Berkeley, fall. 1998.

[25] R. C. Wood, "On optimum quantization," IEEE Transactions on Information Theory, IT-15(2):248-252, March 1969.

[26] T. Berger, "Optimum quantizers and permutation codes," IEEE Transactions on Information Theory, IT-18(6):759-765, November 1972.

[27] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," IRE Int. Conv. Rec., part4, 7:142-163, 1959. Reprinted with changes in Information and Decision Processes, ed. R. E. Machol, McGraw-Hill, pp. 93-126, New York, 1960.

[28] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communication, COM-28(1):84-95, January 1980.

[29] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," IEEE Transactions on Information Theory, 35(2):299-315, March 1989.

[30] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression," Kluwer Academic Publishers, Boston, MA 1992.

[31] W. R. Bennett, "Spectra of quantized signals," Bell Syst. Tech. J., 27(3):446-472, July 1948.

[32] H. Gish and J. P. Pierce, "Asymptotically efficient quantizing," IEEE Transactions on Information Theory, IT-14(5):676-683, September 1968.

[33] N. Farvardin and J. W. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources," IEEE Transactions on Information Theory, IT-30(3):485-497, May 1984.

[34] M. Ammar, "Optimisation d'un schéma de codage d'image à base d'une TCD. Application à un codeur JPEG pour l'enregistrement numérique à bas débit," thèse de Doctorat, ENST Paris, Janvier 2002.

[35] L. G. Roberts, "Picture coding using pseudo-random noise," IRE Transactions on Information Theory, IT-8(2):145-154, February 1962.

[36] S. P. Lipshitz, R. A. Wannamaker and J. Vanderkooy, "Quantization and dither: A theoretical survey," J. Audio Eng. Soc., 40(5):355-375, May 1992.

[37] R. M. Gray and T. G. Stockham, "Dithered quantizers," IEEE Transactions on Information Theory, 39(3):805-812, May 1993.

[38] Stephen So, BEng (Hons), "Efficient block quantization for image and speech coding," Ph. D. thesis, Faculty of Engineering and Information Technology, Griffith University, Brisbane, Australia, March 2005.

[39] A. Gersho and B. Ramamurthi, "Image coding using vector quantization," in Proc. IEEE Int. Conf. Acoust., speech, Signal Processing, pp. 428-431, 1982.

[40] V. Bhaskaran and K. Konstantinides, "Image & Video Compression Standards," 2nd Edition, Kluwer International Series, Boston, 1997.

[41] Simon Lucey, "Low-Rate Image Compression using Vector Quantization," Bachelor of Electrical and Electronic Engineering, Faculty of Engineering and Information Technology, University of Southern Queensland, November 2005.

[42] S. A. Khayam, "The Discrete Cosine Transform (DCT): theory and applications," Department of Electrical & Computer Engineering, Michigan State University, March 10th 2003.

[43] K.P. Subbalakshmi, "Joint Source-Channel Decoding of Variable-Length Encoded Sources with Applications to Image Transmission," Ph. D. thesis, Simon Fraser University, July 2000.

[44] K. Sayood, "Introduction to Data Compression," San Francisco, California: Morgan Kaufmann Publishers Inc. 1996.

[45] G. Davis and A. Nosratinia, "Wavelet-based image coding: an overview," Applied and Computational Control, Signals, and Circuits, vol. 1, no. 1, pp. 205-269, 1998.

[46] H. Anton and C. Rorres, "Elementary Linear Algebra," 7th edition, Anton Textbooks, Inc., New York 1994.

[47] A. N. Netravali and J. O. Limb, "Picture coding: A review", Proc. IEEE, vol. 68, no. 3, pp. 366-406, March 1980.

[48] M. Effros, H. Feng and K. Zeger, "Suboptimality of the Karhunen-Loeve transform for transform coding", IEEE Transactions on Information Theory, vol. 50, no. 8, pp. 1605-1619, Aug. 2004.

[49] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, pp. 90–93, Jan. 1974.

[50] K. Rao and P. Yip, , "Discrete Cosine Transform, " Academic Press Inc., 1250 Sixth Avenue, San Diego, CA 92101, 1990.

[51] W. B. Pennebaker and J. L. Mitchell, "JPEG – Still Image Data Compression Standard," New York: International Thomson Publishing, 1993.

[52] S. Kim and S. Lee, "Image vector quantizer based on a classification in the DCT domain," IEEE Transactions on Communication, 39(4), 549-556, 1991.

[53] H. C. Reeve and J.S. Lim, "Reduction of blocking effect in image coding", in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 1212-1215, 1983.

[54] H. S. Malvar and D. H. Staelin, "The LOT: transform coding without blocking effects", IEEE Trans. Acoust., Speech, Signal Processing, vol. 37(4), pp. 553-559, Apr. 1989.

[55] C. Archer and T.K. Leen, "A generalized Lloyd-type algorithm for adaptive transform coder design", IEEE Transactions on Signal Processing, vol. 52, no. 1, pp. 255-264, Jan. 2004.

[56] W. Chen and C.H. Smith, "Adaptive coding of monochrome and color images", IEEE Trans. Commun., vol. COM-25(11), pp. 1285-1292, Nov. 1977.

[57] J. Woods, "Subband Image Coding," Signal Processing Series, Boston: Kluwer Academic. Ed. 1991.

[58] M. Vetterli and J. Kovacevic, "Wavelet and Subband Coding," Englewood Cliffs, NJ, Prentice Hall PTR, 1995.

[59] P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques," IEEE ASSP Magazine 4 (3), 4-20. July 1987.

[60] D. Esteban and C. Galand, "Application of quadrature mirror filters to split band voice coding schemes," in Proc. IEEE Int. Conf. Acoustic, Speech, Signal Processing, pp.191-195, 1977.

[61] J. G. Proakis and D. G. Manolakis, "Digital Signal Processing: Principles, Algorithms, and Applications," 3rd Edition, New Jersey: Prentice-Hall, 1996.

[62] J. D. Johnston, "A filter family designed for use in quadrature mirror filters banks," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 291-294. 1980.

[63] M. Antonini, T. Gaidon, P. Mathieu and M. Barlaud, "Wavelet transform and image coding", in Wavelets in Image Communication, M. Barlaud, Ed. Amsterdam: Elsevier, pp. 65-188, 1994.

[64] S. G. Mallat, "A theory of multiresolution signal decomposition: the wavelet representation," IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-11, 674-693, July 1989.

[65] I. Daubechies, "Orthonormal bases of compactly supported wavelets," Comm. Pure Appl. Math 41, 909-996, 1988.

[66] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[67] D. Gabor, "Theory of communication", Proc. IEE, 1936.

[68] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. Image Processing, vol. 1, no. 2, pp. 205-220, Apr. 1992.

[69] C. Valens, "A really friendly guide to wavelets," Technical Report, 1999. Available: http://perso.wanadoo.fr/polyvalens/clemens/download/arfgtw.pdf

[70] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code", IEEE Transactions on Communication, vol. 31(4), pp. 532-40, Apr. 1983.

[71] Panrong Xiao, "Image Compression by wavelet transform," Master of Science thesis, East Tennessee State University, August 2001.

[72] T. Frajka, "Image Coding Subject to Constraints," Ph. D. thesis, University of California, Sandiego, 2003.

[73] M. Vetterli, "On Fourier and wavelets: representation, approximation, and compression," presented at Wavelet and Multifractal Analysis 2004, Cargese, Corsica, July 2004. Available: http://www.inrialpes.fr/is2/people/pgoncalv/WAMA2004/lectures/Vetterlilecture.pdf.

[74] S. D. Servetto, K. Ramchandran, V. Vaishampayan and K. Nahrstedt, "Multiple description wavelet based image coding," IEEE Transactions on Image Processing, vol. 9, pp. 813-826, May 2000.

[75] A. A. Mavlankar, "Robust video communication using motion-compensated lifted 3-D wavelet coding," Master thesis, Technical University of Munich, September 2004.

[76] H. Coward, "Joint source-channel coding: development of methods and utilization in image communications,"Doctoral thesis, Science and Technology University of Norway, 2001.

[77] H. S. Witsenhausen, "On source networks with minimal breakdown degradation," Bell Syst. Tech. J., 59(6):1083-1087, July-August 1980.

[78] Z. Zhang and T. Berger, "New results in binary multiple descriptions," IEEE Transactions on Information Theory, IT-33(4):502-521, July 1987.

[79] N. Jayant, "Subsampling of a DPCM speech channel to provide two self-contained half-rate channels,"Bell Syst. Tech. Journal, Vol. 40, No. 4, April 1981.

[80] N. Jayant and S. Christensen, "Effects of packet losses in waveform coded speech and improvements due to an odd-even sample-interpolation procedure," IEEE Transactions on Communications, Vol. 29, No. 2, February 1981.

[81] J. K. Wolf, A. D. Wyner, and J. Ziv, "Source coding for multiple descriptions," Bell Syst. Tech. J., 59(8):1417-1426, October 1980.

[82] L. Ozarow, "On a source-coding problem with two channels and three receivers," Bell Syst. Tech. J., 59(10):1909-1921, December 1980.

[83] R. Ahlswede, "The rate distortion region for multiple descriptions without excess rate," IEEE Transactions on Information Theory, IT-31(6):721-726, November 1985.

[84] Z. Zhang and T. Berger, "Multiple description source coding with no excess marginal rate," IEEE Transactions on Information Theory, 41(2):349-357, March 1995.

[85] R. Zamir, "Gaussian codes and Shannon bounds for multiple descriptions," IEEE Transactions on Information Theory, vol. 45, pp. 2629–2635, Nov. 1999.

[86] M. Pereira, "Multiple description image and video coding for noisy channels," Doctoral

thesis, Sofia-Antipolis University, Nice June 2004.

[87] R. Venkataramani, G. Kramer, and V. K. Goyal, "Bounds on the achievable region for certain multiple description coding problems," in Proc. IEEE Int. Symp. Information Theory, Washington, DC, p. 148, June 2001.

[88] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," IEEE Transactions on Information Theory, 37(2):269-275, March 1991.

[89] V. Koshelev, "Multilevel source coding and data-transmission theorem," in Proc. VII All-Union Conf. Theory of Coding and Data Transm. Vilnius, USSR, pt. 1, pp. 85–92, 1978.

[90] V. Koshelev, "Hierarchical coding of discrete sources," Probl. Pered. Inform., vol. 16, no. 3, pp. 31-49, 1980.

[91] V. Koshelev, "An evaluation of the average distortion for discrete scheme of sequential approximation,"Probl. Pered. Inform. vol. 17, no. 3, pp. 20–33, 1981.

[92] V. A. Vaishampayan, "Design of multiple description scalar quantizers," IEEE Transactions on Information Theory, 39(3):821-834, May 1993.

[93] S.D. Servetto, "Multiple description wavelet based image coding," Image Processing, IEEE Transactions on Image Processing, Volume: 9 Issue: 5, Page(s): 813 -826, May 2000.

[94] P.G., Sherwood, K. Xiaodong Tian, Zeger, "Efficient image and channel coding for wireless packet networks," International Conference on Image Processing, 2000. Proceedings. 2000, Volume: 2 Page(s): 132 -135 vol.2,10-13 Sept. 2000.

[95] V. A. Vaishampayan and J. Domaszewicz, "Design of entropy-constrained multiple description scalar quantizers," IEEE Trans. Inform. Th., 40(1):245-250, January 1994.

[96] T. Guionnet, C. Guillemot and E. Fabre, "Soft decoding of multiple descriptions," In IEEE International Conference on Multimedia, ICME, 26-29, Lausanne, Suisse, August 2002.

[97] V. A. Vaishampayan and J.-C. Batllo, "Asymptotic analysis of multiple description quantizers," IEEE Transactions on Information Theory, vol. 44, pp. 278–284, Jan. 1998.

[98] S. N. Diggavi, N. J. A. Sloane, and V. A. Vaishampayan, "Design of asymmetric multiple description lattice vector quantizers," in Proc. Data Compression Conf., Snowbird, UT, Mar. 2000.

[99] M. Fleming and M. Effros, "Generalized multiple description vector quantization," in Proc. Data Compression Conf., Snowbird, UT, Mar. 1999.

[100] J. A. Kelner, V. K Goyal, and J. Kovacevic, "Multiple description lattice vector quantization: Variations and extensions," in Proc. Data Compression Conf., Snowbird, UT, Mar. 2000.

[101] M. Orchard, Y. Wang, V. A. Vaishampayan, and A. R. Reibman, "Redundancy rate-distortion analysis of multiple description coding using pairwise correlation transforms," in Proc. IEEE Int. Conf. Image Processing, Santa Barbara, CA, Oct. 1997.

[102] Y. Wang, T. Orchard, and A. R. Reibman, "Multiple description image coding for noisy channels by pairing transform coefficients," in Proc. IEEE 1997 1st Workshop Multimedia Signal Processing, Princeton, NJ, June 1997.

[103] V. Goyal and J. Kovacevic, "Optimal multiple description transform coding of Gaussian

vectors," in Proc. Data Compression Conf., Snowbird, UT, Mar. 1998.

[104] T. M. Cover and J. A. Thomas, "Elements of Information Theory," New York: Wiley, 1991.

[105] K. Khelil, R. E. Bekka, M. Bouziani and A. Djebbari, "Multiple Description Coding of Images Using Correlating Transforms," Colloque TELECOM'2007 & 5ème JFMMA, Fès (Maroc), 14-16 Mars 2007.

[106] S. Rout, "Orthogonal vs. Biorthogonal Wavelets for Image Compression," Master of Science thesis, Virginia Polytechnic Institute and State University, August 2003.

[107] K. Khelil, R. E. Bekka, A. Djebari and J. M. Rouvaen," Multiple description wavelet-based image coding using correlating transforms," Int. J. Electron. Commun. (AEU-) 61, 411–417, 2007.

[108] K. Khelil, R. E. Bekka, A. Djebari and J. M. Rouvaen, "Wavelet based image coding using multiple description transform coding," OPTIQUE'06, INPT, RABAT, 19-20 Avril 2006.

[109] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: an overview," IEEE Trans Consumer Electron; 46:1103–27, 2000.

[110] I. Daubechies, "Ten lectures on wavelets," CBMS Conf. Lect. Notes Ser. Appl. Math. 61, SIAM , Philadelphia, PA, 1992.