Robust Coefficient Encoding for Wavelet-based Video Coding

By

Junci Zhang, B.A.Sc.

A thesis submitted to

the Faculty of Graduate and Postdoctoral Affairs

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario, Canada

September 2010

©Copyright

2010 – Junci Zhang

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-71559-8
Our file Notre référence
ISBN: 978-0-494-71559-8

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

The undersigned hereby recommend to

the Faculty of Graduate and Postdoctoral Affairs

acceptance of the thesis,

Robust Coefficient Encoding for Wavelet-based Video Coding

Submitted by **Junci Zhang, B.A.Sc.**

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

Richard M. Dansereau, Thesis Supervisor

Howard M. Schwartz, Chair, Department of Systems and Computer Engineering

Carleton University

September 2010

Abstract

Multiple description coding (MDC) is a promising candidate for coding video information into multiple bitstreams, and it is a very useful technique for video transmission over error-prone packet-switched networks. The three-dimensional (3-D) set partitioning in hierarchical trees (SPIHT) algorithm and its multiple description (MD) extension, the spatial and temporal tree preserving 3-D SPIHT (STTP-SPIHT) algorithm have proved their efficiency in noiseless and noisy channels. In this thesis, we propose an error resilient domain-partitioning based MD video coding algorithm, which is an extension of the STTP-SPIHT method. We extend upon STTP-SPIHT by intentionally inserting a certain amount of redundant information into the multiple substreams of the encoded original video sequence in order to protect those wavelet coefficients in the approximation subband against transmission errors. At the receiver side, this redundant information, along with the other correctly received substreams, are used to predict the missing coefficients in the lost substream. The experimental results in the noiseless and noisy channels demonstrate that our proposed MD video coding system is more robust to random channel bit errors with little increase in its complexity and little loss in noiseless channel performance when compared to STTP-SPIHT.

Acknowledgments

First of all, I would like to offer my sincere gratitude to my supervisor, Professor Richard Dansereau for his knowledge, guidance, and support during the course of this research. His patience and valuable feedback has helped me tremendously in the completion of the thesis work.

Furthermore, I would also like to express special thanks to my parents and brother for providing me with constant support and encouragement during my studies at the university.

Table of Contents

List of Tables

List of Figures

List of Acronyms

1-D	One-dimensional
2-D	Two-dimensional
3-D	Three-dimensional
ARQ	Automatic repeat request
BER	Bit error rate
BFS	Breadth first search
BPP	Bits per pixel
BSC	Binary symmetric channel
CDF	Cohen-Daubechies-Feauveau
CRC	Cyclic redundancy code
DCT	Discrete cosine transform
DWT	Discrete wavelet transform
EZW	Embedded zerotree wavelet
FEC	Forward error correction
FPS	Frames per second
GOP	Group of pictures
HVS	Human visual system
JPEG	Joint photographic experts group
LC	Layered coding
LIP	List of insignificant pixels
LIS	List of insignificant sets
LSP	List of significant pixels

MD	Multiple description
MDC	Multiple description coding
MPEG	Moving pictures experts group
MSE	Mean-squared error
PSNR	Peak signal-to-noise ratio
QoS	Quality of service
RCPC	Rate-compatible punctured convolutional
R-D	Rate-distortion
S-T	Spatio-temporal
SPIHT	Set partitioning in hierarchical trees
STTP-SPIHT	Spatial and temporal tree preserving 3-D set partitioning in hierarchical trees

Chapter 1: Introduction

1.1 Motivation

In today's information era, video streaming over the Internet has become a more and more important way for people to distribute information around the globe. Some examples which use this technology include teleconferencing, telemedicine, distance learning, and live webcasts. However, there are still many challenging technical problems that need to be solved in order to transport real-time video over the Internet while guaranteeing a required quality of service (QoS) level.

In packet-switched networks, such as the Internet, packets are discarded at random when the number of packets sent exceeds transmission capacity without considering the relative importance of the packets. In other words, packet losses are inevitable in such networks. Retransmission of lost packets based on automatic repeat request (ARQ) is one of the most common techniques to protect data against packet losses; however, the additional round-trip delay is not suitable for real-time video applications. Additionally, since packet losses often result from buffer overflow in times of high network load, retransmitted packets will only add even more congestion to an already congested network. Therefore, it is more desirable to develop error-resilient and network-adaptive video coders in order to provide reasonable video quality at various network loss rates.

Most current standardized video coders, including Moving Picture Experts Group-2 (MPEG-2) [1], MPEG-4 [2], H.263 [3], and H.264 [4] achieve high compression efficiency by using motion-compensated prediction to reduce the temporal and statistical redundancy between the video frames. However, this may result in error propagation, where errors due to packet loss in a reference frame propagate to all of the subsequent dependent frames leading to visual artifacts [5]. Without effective control of temporal error propagation, the reconstruction quality of the video sequence can become seriously degraded. Furthermore, these video coders lack desirable features required by today's video applications, such as low computational complexity and full embeddedness for progressive transmission.

The three-dimensional (3-D) set partitioning in hierarchical trees (SPIHT) algorithm made a major breakthrough in video compression. It provides outstanding rate-distortion (R-D) performance with relatively low computational complexity. In contrast to the motion-compensated prediction video coding schemes, the 3-D SPIHT algorithm is an effective wavelet-based video codec which produces an embedded or progressive bitstream. The capability of progressive transmission makes it highly adaptive to channel capacity fluctuation under time-varying network conditions. With a progressive bitstream, the reception and transmission of code bits may be ceased at any point and decoded at lower rates. While the 3-D SPIHT coder has many advantages, it is quite fragile against bit errors in noisy communication channels. SPIHT-encoded bitstreams are highly susceptible to bit errors due to their dependence among wavelet coefficients in constructing a "significance map". A single bit error could potentially lead to loss of synchronization between the encoder and decoder execution paths, which

would lead to uncontrolled degradation of reconstructed video quality. Hence, it is essential to improve the error-resilience in the 3-D SPIHT algorithm in order to achieve robust video transmission over packet loss channels.

There are various techniques which may be utilized to enhance the error-resilience of video transmission over error-prone packet switched networks, and these techniques can be categorized as follows: source coding, channel coding, or error concealment. The focus of our thesis work is on source coding techniques, and examples of such techniques include multiple description coding (MDC) [6]-[8] and layered coding (LC) [9],[10]. In the following subsections, we will elaborate further on each of these coding techniques.

1.1.1 Layered Coding

Unlike traditional coding schemes that generate a single bitstream, MDC and LC generate two or more bitstreams. The main difference between MDC and LC lies in the dependency. In LC, one bitstream is sent as a base layer and the other bitstreams are sent as progressive enhancement layers. The base layer is the most critical layer, and it can be decoded independently of the enhancement layers. On the other hand, the enhancement layers are applied only to refine the base layer quality, and they are not useful by themselves. If the base layer is not received correctly, the information received from the respective enhancement layers is rendered useless. Moreover, an enhancement layer may be decoded only if the base layer and all the previous enhancement layers are received correctly. Therefore, this makes the performance of streaming applications that employ the layered representations sensitive to losses of base layer packets. As a result, the

delivery of the base layer must be guaranteed by using recovery mechanisms such as ARQ or forward error correction (FEC). As we can see, one major obstacle for the adoption of LC in practical networks is that to guarantee a basic level of quality, the base layer must be delivered almost error free.

1.1.2 Multiple Description Coding

MDC is another popular video coding technique which has been proposed for streaming over unreliable channels. In contrast to LC, multiple description (MD) coders encode the video data into multiple bitstreams also referred to as descriptions, and they are normally of equal importance. Each description alone can guarantee a basic level of reconstruction quality of the source, and every additional description can further improve that quality. This is a desirable property in the context of Internet transmission where none of the packets receive preferential treatment. The packets of each description are then potentially routed over multiple, disjoint paths. Each description is individually decodable so that loss of some of the descriptions will not jeopardize the decoding of those descriptions that are correctly received, and the quality of the decoded video improves with more descriptions received in parallel. In other words, MDC can provide adequate quality without requiring retransmission of any lost packets if at least one of the descriptions is received correctly. This characteristic of an MD coder makes it highly suitable for video transmission over unreliable networks. Hence, MDC has attracted a lot of attention as a promising candidate for error-resilient video transmission. The interested reader is referred to [11]-[13] for a more detailed comparison of the performance between MDC and LC.

A generic MD video codec for two descriptions is illustrated in Figure 1.1. The MD encoder creates two descriptions which are sent separately across two channels. If only one of the two descriptions is received, the two side decoders produce lower but acceptable quality reconstructions with side distortions D_1 and D_2. When both descriptions are received, the central decoder produces a high-quality reconstruction with central distortion D_0.

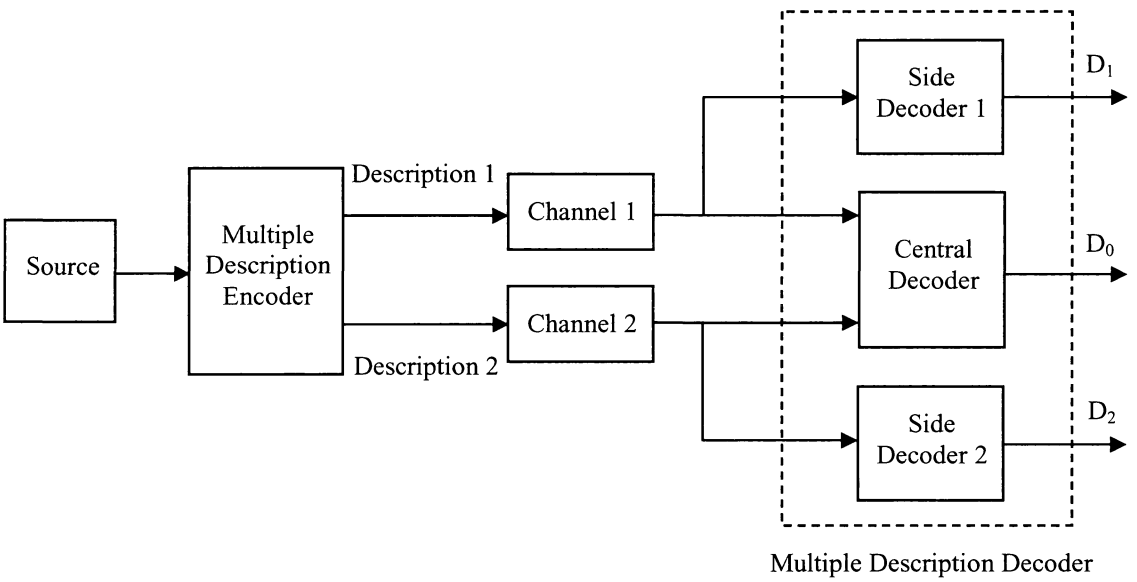


Figure 1.1: Two-description MD video codec.

To date, several multiple-substream generation methods have been proposed for wavelet-based coders to obtain error-resilience at high packet loss rates. One such method was first proposed by Creusere for use with the embedded zerotree wavelet (EZW) algorithm, in which the wavelet coefficients of an image were partitioned into several groups and each group is then independently processed using an EZW coder

[14],[15]. This algorithm allows more uncorrupted information to reach the decoder because a bit error in any one group does not affect the others. Later, Alatan *et al.* demonstrated that the embedded image bitstreams can be delivered with error resilience maintained by demultiplexing the SPIHT bitstream into multiple classes [16]. These sub-classes are protected by different channel coding rates of the rate-compatible punctured convolutional (RCPC) coder [17] to improve the overall performance against channel bit errors. Extending upon Creusere's earlier work to 3-D SPIHT coders, Cho and Pearlman proposed a domain-partitioning based MD video coding scheme with the 3-D SPIHT algorithm, called the spatial and temporal tree preserving 3-D SPIHT (STTP-SPIHT) [18],[19]. In this MD approach, after wavelet decomposition, the 3-D wavelet coefficients are partitioned into two or more equal-sized groups according to their spatial and temporal relationship, and each group is then encoded independently using the 3-D SPIHT algorithm to create multiple embedded 3-D SPIHT substreams. They also applied the channel coding method proposed by Sherwood and Zeger [20] to every packet. The benefit of this domain-partitioning based MD video coding approach is that any bit error in the bitstream belonging to any one block does not affect the other blocks. As a result, this scheme achieves error resilience against transmission errors.

1.2 Thesis Statement

The objective of this thesis is to design a robust domain-partitioning based MDC system for video transmission by intentionally inserting redundant information to each description or substream so that the video sequence may be reconstructed in the presence of packet loss. We then conduct experimentation to demonstrate that our proposed video

coding algorithm provides more resilience to random channel bit errors over the existing domain-partitioning based MD video coding approach.

1.3 Contributions

In order to achieve the objectives described in the previous section, we make two main contributions, and they may be summarized as follows:

- To achieve error resilience against channel errors, we propose a domain-partitioning based MD video coding algorithm based on Cho and Pearlman's previous work for STTP-SPIHT video coding with the insertion of additional redundant information. The redundant data is inserted intentionally into the multiple substreams of the encoded original video sequence so that we can protect those wavelet transform coefficients in the lowest spatio-temporal frequency subband or approximation subband since that is where most of the signal energy is generally concentrated.
- At the decoder side, the missing coefficients in the approximation subband of the lost substream can be recovered by using the additional redundant data along with the correctly received substreams. We then perform a comparative analysis of our proposed MD coding approach with the STTP-SPIHT coding scheme in the noiseless and noisy channels. The experimental results have demonstrated that our proposed algorithm achieves higher level of error resilience in noisy channels when compared to STTP-SPIHT.

Having defined the problem, the concluding section of this chapter describes the general organization of this thesis.

1.4 Thesis Organization

To facilitate a better understanding of the concepts, each chapter is preceded by an introduction highlighting its contents, and it concludes by providing a detailed discussion on relevant concepts. Thus, the remaining chapters in this thesis are organized as follows:

- Chapter 2 presents a brief overview of the discrete cosine transform (DCT) based hybrid video coding, wavelet transform, the basic principles behind the EZW and SPIHT coding algorithms, as well as its extension to MD video coding. The MD coding scheme discussed in this chapter will then provide the basis for our proposed video coding algorithm.
- Chapter 3 provides a detailed discussion on our proposed algorithm. We present the redundancy insertion methods, the modified 3-D SPIHT coder implementation, as well as the root subband recovery technique used to recover missing information.
- Chapter 4 presents the simulation results from our proposed video coding algorithm in noiseless and noisy channels.
- Finally, Chapter 5 presents our conclusions and several possibilities for future research work.

Chapter 2: Background Review

2.1 Introduction

To better understand the algorithms discussed in this chapter, as well as the MD video coding algorithm proposed in this thesis, some background information must be discussed. We begin this chapter by reviewing the DCT-based hybrid video coding. Next, we present an overview of the wavelet transform. This is followed by a brief discussion of EZW, which is a fully embedded wavelet coding algorithm with precise rate control and low complexity. We then introduce the two-dimensional (2-D) SPIHT algorithm along with its extension to 3-D SPIHT video coding approach. Finally, we examine the STTP-SPIHT method which is a very successful domain-partitioning based MD video coder built upon 3-D SPIHT.

2.2 DCT-based Hybrid Video Coding

The most widely used scheme for video compression is based on DCT-based hybrid video coding, which consists of motion-compensated prediction along with transform coding, and this hybrid approach is seen in most current video coding standards, such as MPEG-2 [1], MPEG-4 [2], H.263 [3], and H.264 [4]. In hybrid video coding,

there are generally two basic coding modes, namely intra-frame coding and inter-frame coding.

In intra-frame coding mode, a video frame is divided into blocks of pixels known as macroblocks, and each macroblock is transformed by the DCT. The resulting transformed coefficients are then quantized and entropy coded. In inter-frame coding mode, a macroblock is predicted using motion compensation. To explain this further, each frame is first partitioned into macroblocks. Each macroblock is then temporally predicted from a best-matched block on a previously encoded frame, referred to as reference frame. To find the best-matched block in the reference frame, a specific block-matching algorithm is used. The difference in motion between the current block and its matching block in the reference frame is defined as the motion vector. This process of motion vector determination is called motion estimation. Moreover, the prediction error or residue, which is the difference between the original and the predicted frames, is transformed with the DCT, and quantization is applied to the resulting transform coefficients. The quantized residue DCT coefficients as well as motion information are compressed by some entropy coding method and sent to the decoder. As we can see, fewer bits are required to code the prediction error and motion information than the original video frame.

Although DCT-based hybrid video coding achieves high coding efficiency, it is highly susceptible to transmission errors. An erroneous reference frame may result in error propagation to subsequent frames. Additionally, motion-compensated prediction lacks full embeddedness for progressive transmission, and is thus unable to adapt source coding rate to channel capacity. Unlike motion-compensated prediction coding schemes,

the 3-D SPIHT algorithm is highly adaptive to channel fluctuations with its progressive transmission capability. Before we proceed with our detailed discussion of the 3-D SPIHT algorithm, we will first briefly describe the wavelet transform used in the SPIHT algorithm.

2.3 Discrete Wavelet Transform

Most image compression algorithms use some form of transform-based analysis. A typical image compression system is shown in Figure 2.1, and it consists of three operations at the encoding stage namely the wavelet transform, quantization, and entropy coding. Compression is accomplished by applying a linear transform to decorrelate the image data, quantizing the resulting transform coefficients, and finally entropy coding the quantized values. Over the years, a variety of linear transforms have been developed, and the most popular transforms used in image compression include the DCT and the discrete wavelet transform (DWT).

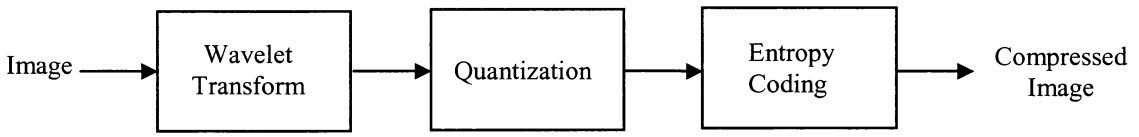


Figure 2.1: An image compression system.

In the 1990's, the Joint Photographic Experts Group (JPEG) established its first international standard for still image compression where the encoders and decoders are

based on the DCT. DCT is well known to achieve relatively good results for moderate compression ratios. However at higher compression ratios, it results in perceptually discomforting blocking artifacts due to the block structure used in JPEG, and the human visual system (HVS) is very sensitive to the presence of these distortions. To completely eliminate the blocking artifacts found in DCT-based methods at high compression ratios, researchers began to incorporate the DWT as a transform tool in compression algorithms.

The DWT has gained widespread acceptance as a powerful tool in audio and image processing, digital communications, and a wide variety of applications in many different fields. It is also increasingly used as an effective solution in image and video coding due to its characteristics of multiresolution analysis [21] and nonblock-based analysis, which are different from conventional usage of transforms such as the DCT. The multiresolution characteristic leads to superior energy compaction and visually pleasing compressed images. In addition, the nonblock-based structure of the wavelet transform completely removes the blocking artifacts. Therefore, these factors contribute to the use of DWT as a transform tool in state-of-the-art standards such as JPEG-2000 image coding [22] and MPEG-4 still texture coding. For a rigorous mathematical description of wavelets, the interested reader is referred to the work by Mallat [23] and Daubechies [24].

Figure 2.2 depicts a multiresolution analysis and synthesis operation of a one-dimensional (1-D) wavelet transform. As illustrated in Figure 2.2, the implementation of the forward 1-D DWT is best described as convolving an input signal with a lowpass (LP) filter and a highpass (HP) filter to produce a lowpass signal (approximation signal) and a highpass signal (detail signal), respectively, after a subsequent downsampling operation

by a factor of two. The lowpass and highpass filter pair used in the analysis stage is also referred to as the analysis filter bank. The reconstruction of the detail and approximation components is performed with an upsampling operation by a factor of two along with another pair of lowpass and highpass filters known as the synthesis filter bank. As we can see, the reconstruction operation is simply the inverse process of the forward decomposition. Finally, the output of the lowpass and highpass filters are summed together to yield the reconstructed signal.

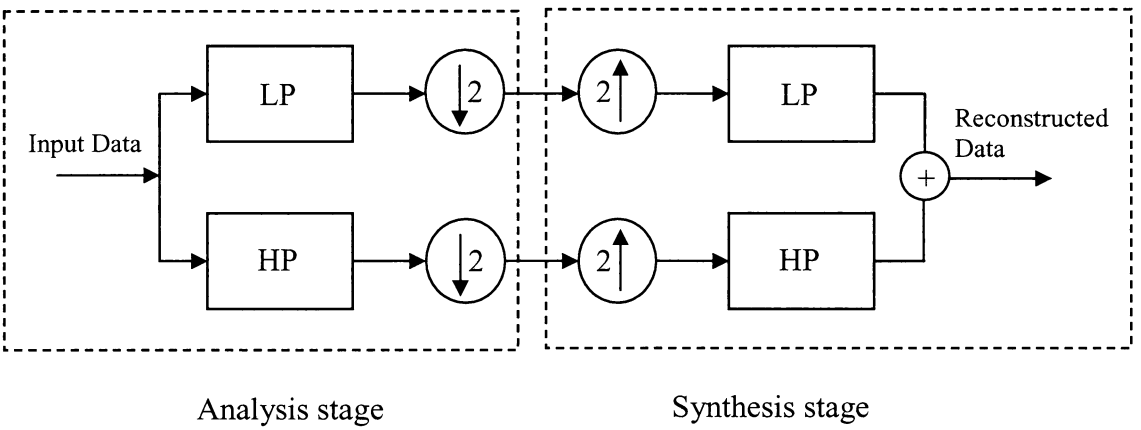
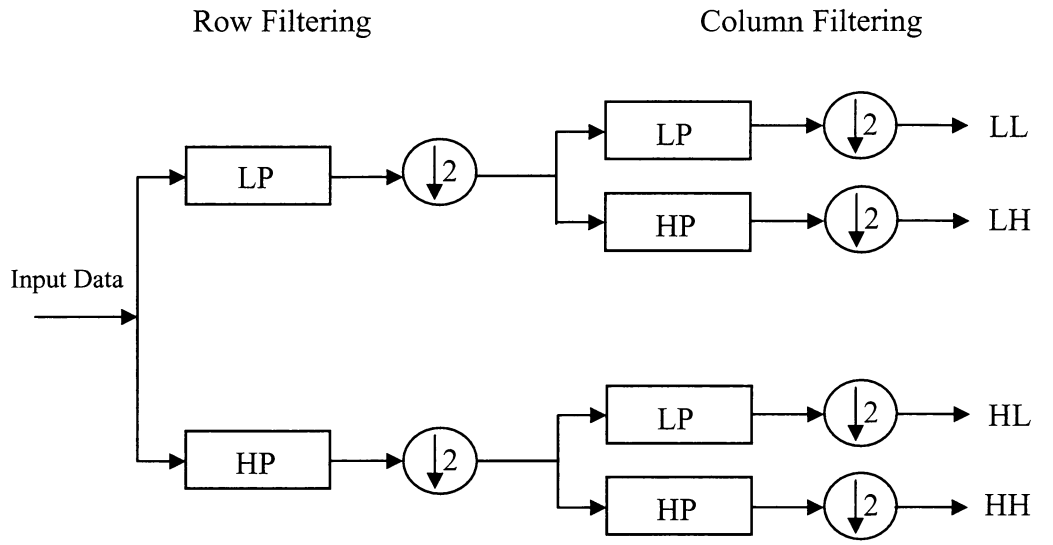


Figure 2.2: One-dimensional DWT analysis and synthesis filter bank.

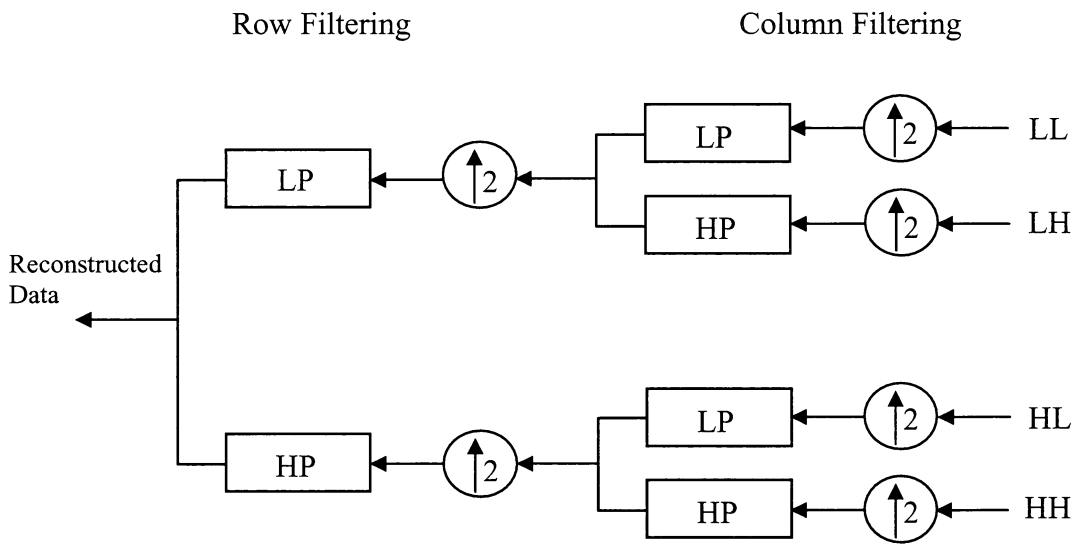
2.3.1 2-D Discrete Wavelet Transform

The 1-D wavelet transform can be extended to a 2-D signal such as an image by performing a 1-D DWT along the rows and then the columns of an image. Figure 2.3(a) and (b) illustrate the analysis and synthesis filter bank structures of a 2-D wavelet transform, respectively. As shown in the analysis stage in Figure 2.3(a), one level of 2-D

wavelet transform decomposes an image into four frequency subbands: one lowpass subband (the approximation subband) called the LL subband, and three highpass subbands (the detail subbands) called the LH, the HL, and the HH subbands. Each subband is one quarter the size of the original image. In addition, the LL subband contains a coarse scale approximation of the original image, and the other three detail subbands LH, HL, and HH exploit image details across the different directions: LH for horizontal, HL for vertical, and HH for diagonal details. In the next level of the transform, we use the LL subband for further decomposition and replace it with four respective subbands. The wavelet transform may be applied recursively to the approximation subband to obtain decomposition at the coarser scales, yielding a hierarchical decomposition or pyramid representation. The synthesis stage in Figure 2.3(b) shows the inverse process of the forward wavelet decomposition, where the original image is reconstructed from the approximation and detail subbands.



(a) Analysis stage



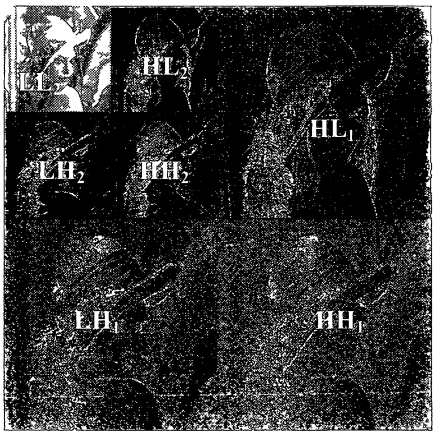
(b) Synthesis stage

Figure 2.3: Two-dimensional DWT. (a) Analysis filter bank. (b) Synthesis filter bank.

Figure 2.4 depicts an illustration of the approximation and detail subbands obtained from a two-level wavelet decomposition on an input image. The approximation subband produced by a 2-D wavelet decomposition contains the low frequency components of the image, whereas the detail subbands contain the high frequency components of the image. As we can clearly see, the approximation subband is essentially a good quality version of the original image at a smaller spatial resolution. In a multi-level wavelet decomposition, the spatial resolution of the approximation subband is divided by a factor of two at each level, producing increasingly smaller versions of the original image. Finally, the reconstruction operation is the inverse process of the forward wavelet decomposition.

(a) Original image

(b) Image after one level of 2-D wavelet decomposition

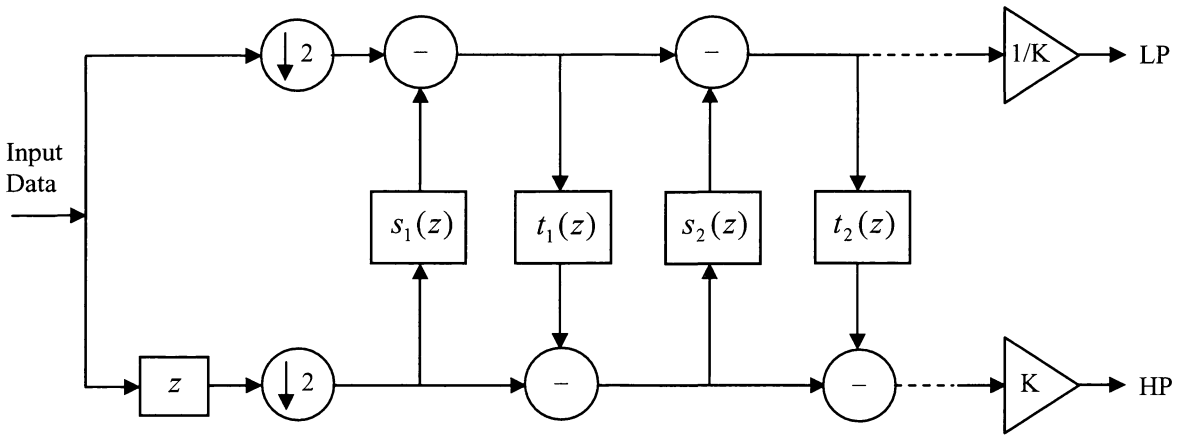


(c) Image after two levels of 2-D wavelet decomposition

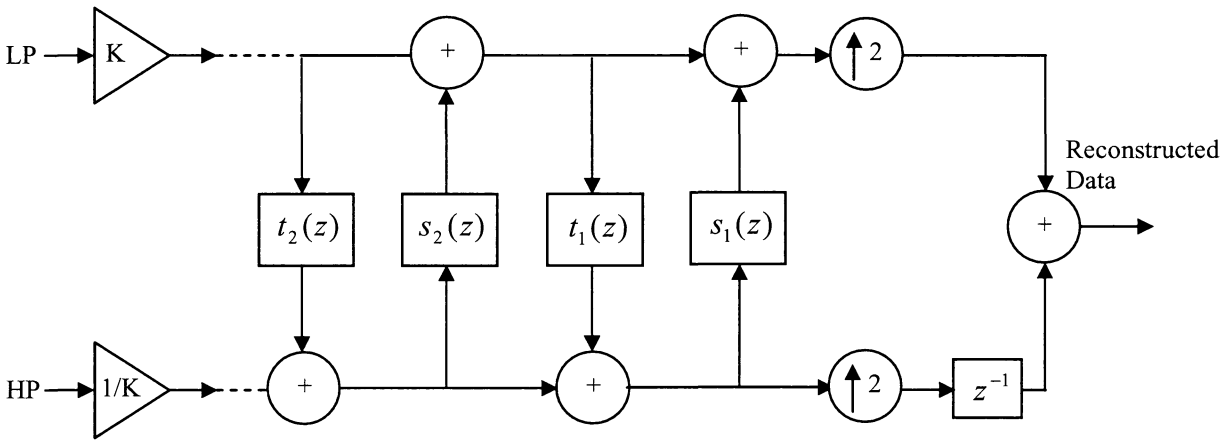
Figure 2.4: Illustration of two levels of 2-D wavelet decomposition. (a) Original image. (b) Image after one level of 2-D wavelet decomposition. (c) Image after two levels of 2-D wavelet decomposition.

Due to the computational complexity involved in constructing the wavelet transform, Sweldens proposed a new approach to construct the biorthogonal wavelets, known as the lifting scheme [25]-[27]. Generally, the lifting scheme consists of three steps: 1) split, 2) predict, and 3) update. The basic principle behind the lifting scheme is to attempt to predict the approximation data from the detail data and to update this in the first step (lifting step). In the next step, the detail data is then predicted from the approximation data (dual lifting step).

A block diagram for this lifting-based implementation of the wavelet transform is illustrated in Figure 2.5. The forward wavelet transform using lifting involves a lazy wavelet, followed by alternating lifting and dual lifting steps, and finally a scaling. The same operations are reversed for the inverse wavelet transform using lifting. When compared with the standard implementation, there is significant reduction in computational complexity when implementing a lifting-based DWT [28]. In some cases, the number of operations can be reduced by a factor of two. Hence a lifting-based strategy is employed when implementing a DWT in this thesis.



(a) Analysis stage



(b) Synthesis stage

Figure 2.5: Wavelet transform using lifting. (a) Analysis stage. (b) Synthesis stage.

2.3.2 3-D Discrete Wavelet Transform

The 2-D wavelet transform can be further extended to a 3-D wavelet transform along the time direction for a video sequence. The 3-D wavelet decomposition is computed by first applying 1-D transform along the temporal dimension until the

required number of temporal decomposition levels is obtained and then performing the 2-D transform on each temporal-transformed frame up to the desired number of spatial decomposition levels. The temporal decomposition is based on the group of pictures (GOP) concept. In order to have a better understanding, let W_X, W_Y, and W_T denote the wavelet transforms along the two spatial (X, Y) directions and one temporal T direction, respectively. The 3-D wavelet packet transform has a pattern of $(W_T \ldots W_T)(W_X W_Y \ldots W_X W_Y)$. Figure 2.6 illustrates a total of 21 subbands obtained from the two-level 3-D wavelet packet transform. Two levels of temporal decompositions are applied on the GOP, followed by two levels of spatial decompositions applied on each temporal-transformed frame.

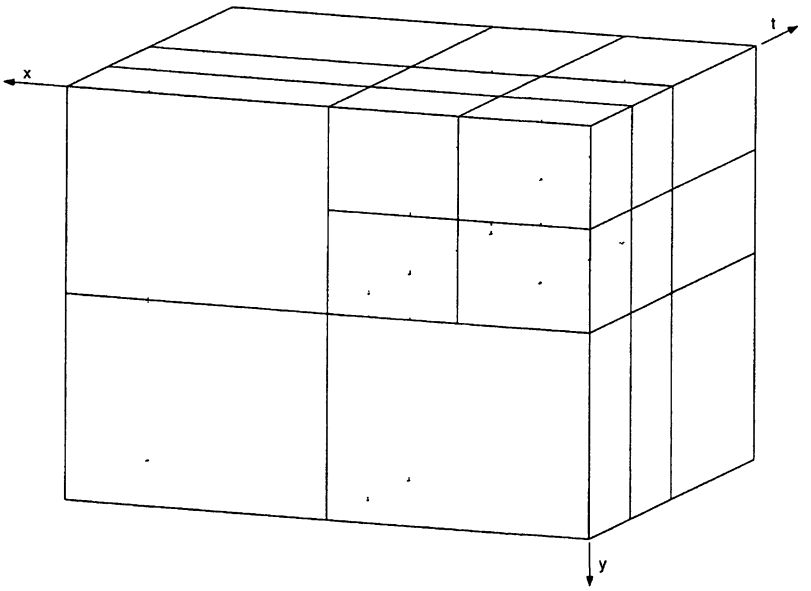


Figure 2.6: A two-level 3-D wavelet packet transform [29].

2.4 EZW Compression

The introduction of the embedded wavelet zerotree image coding techniques developed by Shapiro made a major breakthrough under the transform-coding framework, and has generated a significant improvement in performance compared to previous image coding methods [30]. Zerotrees allow an efficient coding technique of coefficients that will result in embedded coding. According to Shapiro, the EZW algorithm is based on three key concepts: 1) exploiting the self-similarity inherent in the wavelet transform to predict the significant information across scales, 2) successive approximation quantization of the wavelet coefficients, and 3) lossless compression using adaptive arithmetic coding.

To better explain the procedures involved in the encoding of the significance map in EZW, a flow diagram is illustrated in Figure 2.7. There are four symbols used in the algorithm: 1) a positive significant coefficient, 2) a negative significant coefficient, 3) a zerotree root, and 4) an isolated zero. Each coefficient will be assigned one of these four values. A zerotree root is defined as an insignificant coefficient for which all of its descendants are also insignificant. Additionally, an isolated zero is an insignificant coefficient as well, but the difference is that it has significant descendants. The zerotree is based on the hypothesis that if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then all wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant with respect to T. A wavelet coefficient c is said to be insignificant with respect to a given threshold T if $|c| < T$. As a result, once a zerotree root has been encoded, all the descending

coefficients in the higher frequency subbands can be discarded, since they are predicted to be insignificant according to the hypothesis.

The EZW algorithm then uses the bit plane coding method to encode the tree structure, yielding a fully embedded bitstream. The coding algorithm is typically performed in two passes: a dominant pass where significant coefficients are identified and a subordinate pass where such coefficients are refined. With an embedded bitstream, the encoder can terminate the encoding at any point, thus allowing a target rate or a distortion metric to be met exactly. Similarly, given a bitstream, the decoder can cease decoding at any point in the bitstream, and produces reconstructions corresponding to all lower-rate encodings. This property of being able to terminate the encoding or decoding of an embedded bitstream at any specific point is extremely useful in systems that are either rate or distortion constrained. Furthermore, this technique achieves excellent results with absolutely no training, no pre-stored tables or codebooks, and no prior knowledge of the image source.

Since its invention in 1993, many enhancements have been proposed to make the EZW algorithm more robust and efficient. An improved version of the EZW coding algorithm, called SPIHT proposed by Said and Pearlman [31], is one of the most well known EZW derivatives. SPIHT adopts a similar concept as that of the zerotree structure of EZW but with a different parent-children relationship. According to Said and Pearlman, results produced from the SPIHT coding algorithm in most cases surpass those obtained from EZW due to the order of coding procedure and widely associated tree structure. The details of the 2-D and 3-D SPIHT algorithm will be described in the following section.

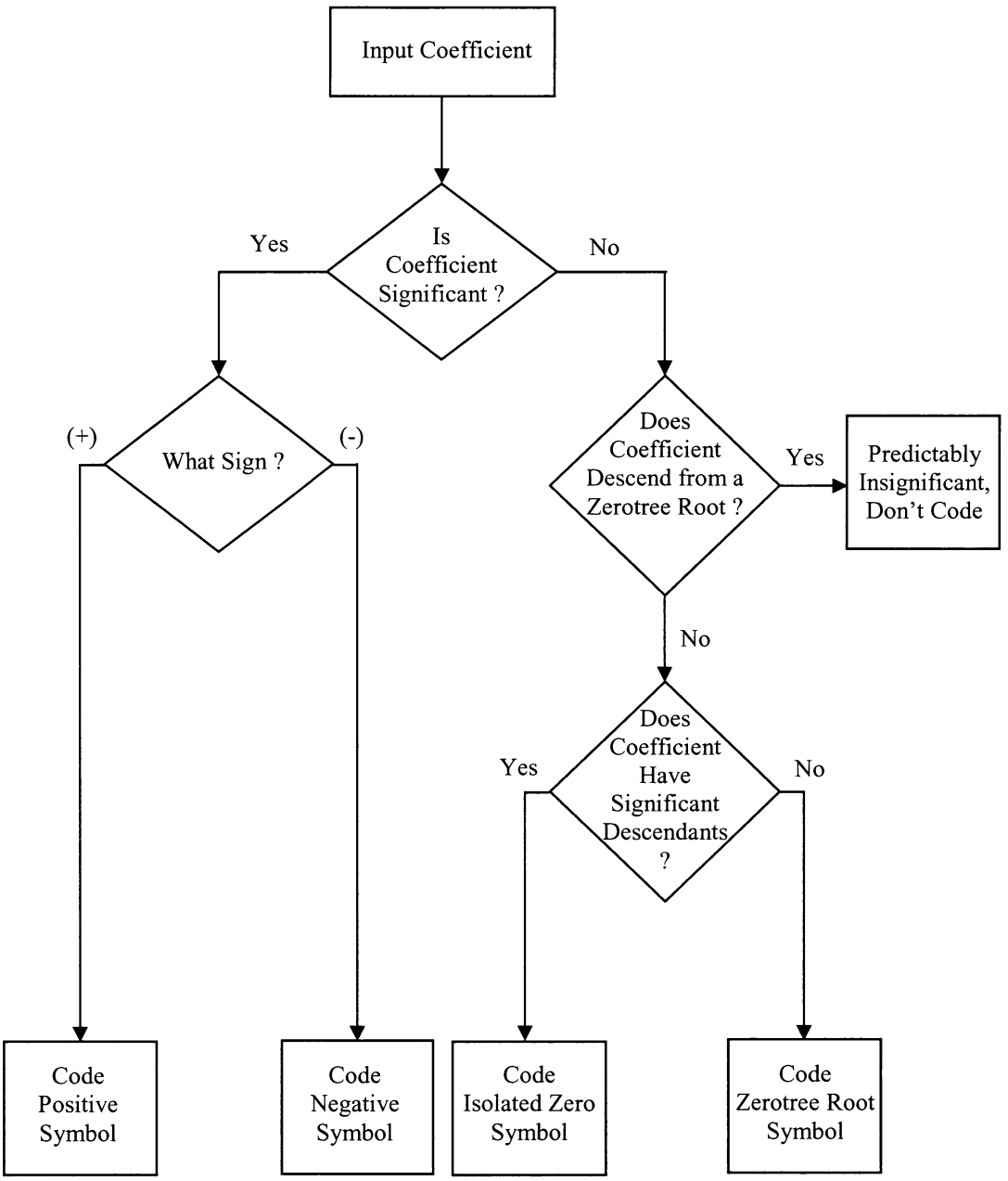


Figure 2.7: Flow chart for encoding a coefficient of the significance map.

2.5 SPIHT Algorithm

2.5.1 2-D SPIHT

In 1996, Said and Pearlman proposed a more efficient representation of the EZW algorithm, referred to as SPIHT [31]. SPIHT is a fully embedded wavelet coding algorithm known for its low computational complexity and excellent performance. This algorithm is very similar to Shapiro's original work. It essentially exploits the inherent similarities across the subbands in a wavelet decomposition of an image. Additionally, it generates an embedded bitstream which incorporates the concepts of ordering the coefficients by magnitude and transmitting the most significant bits first. This progressive property of SPIHT makes it adaptive to channel capacity fluctuation under time-varying network conditions.

In the SPIHT algorithm, the image is first decomposed into a number of subbands by means of hierarchical wavelet decomposition as previously mentioned in section 2.3.1. The subband coefficients are then grouped into a tree structure known as a spatial orientation tree, which efficiently exploits the correlation between the frequency bands. Figure 2.8 illustrates the spatial orientation tree structure defined with two levels of wavelet spatial decomposition. The arrows in Figure 2.8 indicate the parent-children relationship in subband pyramid. The start of the arrow line is the parent coefficient, and the end of arrow indicates four children coefficients.

The tree structure is defined in such a way that each node consists of 2 x 2 adjacent wavelet coefficients, and each coefficient inside the node, except for those in the lowest frequency subband, has either no descendants (the leaves) or four descendants of the same spatial orientation in the next higher frequency subband. However, the

offspring branching rule is different for the lowest frequency subband (the tree roots), where one of the coefficients in each node, indicated by the star in Figure 2.8, has no descendants. Hence, the parent-offspring linkage, with the exception of the highest and lowest frequency subbands, can be defined as follows:

$$O(i,j) = \{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\} \tag{2.1}$$

where $O(i,j)$ represents a set of coordinates of all the offspring of the wavelet coefficient at location (i,j).

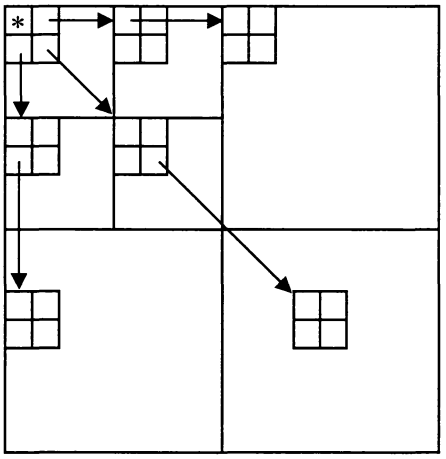


Figure 2.8: Parent-offspring dependency in 2-D SPIHT.

Once the spatial orientation tree structure has been defined, the next step is to encode the wavelet coefficients to create an embedded bitstream. The SPIHT algorithm codes a wavelet by transmitting information about the significance of a wavelet

coefficient. A magnitude test is performed to decide whether a coefficient is significant or not. A coefficient is regarded as "significant" if its magnitude is greater than or equal to a given threshold; otherwise, it is called "insignificant". This set-partitioning test approach is based on the idea that, with a spatial orientation tree, if the magnitude of a parent coefficient in the lower frequency subband is insignificant, it is highly likely that the magnitudes of its descendants are also insignificant. Additionally, the aim is to locate the significant bits in each bit plane of the wavelet coefficients with the minimum tree cost; therefore, it is important to implement an efficient search algorithm.

SPIHT employs the breadth first search (BFS) algorithm [32]. In other words, the search always begins from the highest energy bit plane level and continues down to the lowest energy bit plane. The scanning pattern of the subbands in a three-level transform can be demonstrated in Figure 2.9. For a three-level transform, the scan begins from the lowest frequency subband, denoted as LL_3, and scans subbands HL_3, LH_3, and HH_3, at which point it then continues on to the next level and so on. To generalize this for an N-level transform, the scan starts at LL_N, and scans HL_N, LH_N, and HH_N in this pattern. It then moves on to level (N-1) and the same pattern is applied. The scanning operation is performed in this fashion until we reach the highest frequency subband. Each coefficient within a given subband is scanned before any coefficient in the next subband. Moreover, the scanning of the coefficients is performed in such a way that no child node is scanned before its parent.

In a practical implementation, the search path at every traversal of a bit plane is stored in three ordered lists in terms of types of the branches and the significance of the wavelet coefficients: a list of insignificant sets (LIS), a list of insignificant pixels (LIP),

and a list of significant pixels (LSP). In all three lists, each entry is identified by a coordinate (i, j).

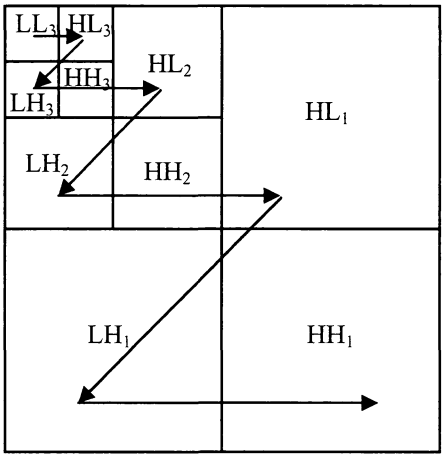


Figure 2.9: Scanning order of subbands for encoding a significance map.

The coding algorithm consists of the following three stages: initialization, sorting pass, and refinement pass. At the initialization stage, the coordinates of all the coefficients in the lowest frequency subband are added to the LIP; the coordinates of all the coefficients in the lowest frequency subband, with the exception of those which do not have descendants, are added to the LIS; and LSP is set as an empty list. In addition, the initial threshold T is set to equal to 2^n, and n is defined as follows:

$$n = \lfloor \log_2 (\max_{(i,j)} \{|c_{i,j}|\}) \rfloor \tag{2.2}$$

where $c_{i,j}$ represents the wavelet transform coefficient at coordinate (i, j).

During the sorting pass, the coding algorithm first begins to sort all the elements in the LIP then in the LIS. Each wavelet coefficient inside the LIP is examined. If a coefficient becomes significant or $|c_{i,j}| \geq 2^n$, its sign is coded and its coordinate is moved to the LSP; otherwise, the coefficient remains in the LIP and no more bits are generated. Similarly, the sets in the LIS are sequentially evaluated. When a set is found to be significant in the LIS, it is removed from the list and partitioned into several new subsets and isolated coefficients. These isolated coefficients are examined again for significance. The new subsets with more than one element are added back to the end of the LIS, whereas the single-coordinate sets are added to the end of the LIP or the LSP, depending on whether they are insignificant or significant, respectively. This basically means that the significant coefficients are added to the end of the LSP, while the insignificant coefficients are added to the end of the LIP.

After the sorting pass, the next step is to perform the refinement pass. The refinement pass refines the entries found in the LSP, except for those coefficients added in the most recent sorting pass, with one additional bit of precision. After the first iteration, the significance threshold is divided by two (n is decremented by one), and these two passes (sorting and refinement) are applied again at the lowered threshold. The process continues through successive halving of the threshold until a specified rate constraint or a distortion requirement is reached.

The overall 2-D SPIHT encoding process is illustrated in Figure 2.10. The block diagram in Figure 2.10 shows that an original image is first passed through a wavelet transform, and the resulting wavelet coefficients are then grouped into a spatial orientation tree. In the next stage, the wavelet coefficients in each spatial orientation tree

are encoded in the sorting and refinement phases to create an embedded bitstream. The output bitstream is then further compressed with an entropy encoder to produce the final bitstream.

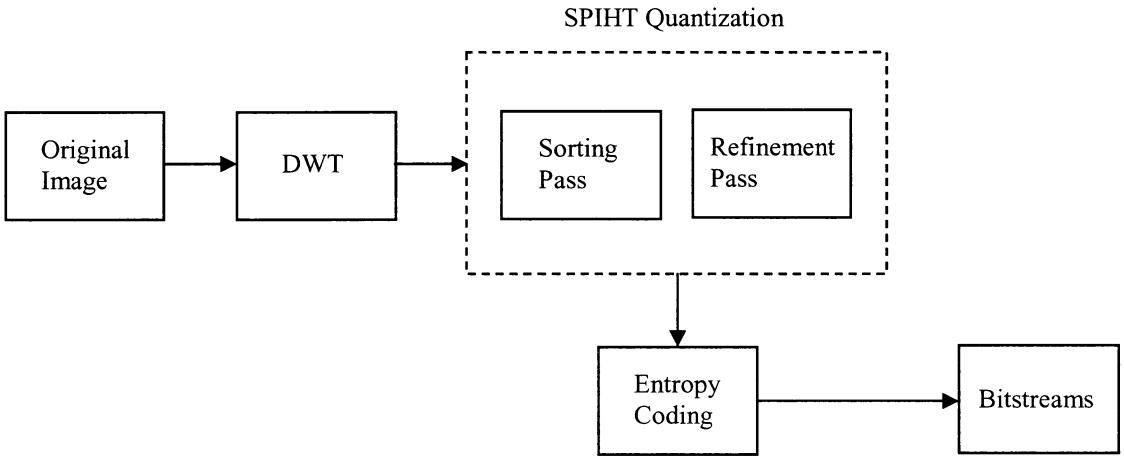


Figure 2.10: SPIHT coding process.

Motivated by the outstanding performance of the 2-D SPIHT method in image compression, researchers have extended the concept to 3-D for video coding, which will be discussed in the next sub-section.

2.5.2 3-D SPIHT

The 3-D SPIHT scheme used for video coding is extended from the 2-D SPIHT, and is shown to have low complexity and high compression efficiency [33],[34]. Similarly, it is a wavelet-based coding algorithm, adopting the parent-children tree structure and bit plane coding. The sorting and refinement stages in the 3-D SPIHT

algorithm are similar to that of the 2-D SPIHT case, except 3-D SPIHT uses 3-D instead of 2-D tree sets.

In the 3-D SPIHT algorithm, the first step is to apply wavelet transform on a number of consecutive frames called GOP in both spatial and temporal domains. As we recall from section 2.3.2, the 3-D wavelet decomposition is computed by first applying a 1-D transform in the temporal dimension until the required number of temporal decomposition levels is obtained, and then performing the 2-D transform on each temporal-transformed frame up to the desired number of spatial decomposition levels. It is important to choose a suitable size for the GOP, since it has an immediate effect on the compression algorithm. Generally, a larger GOP results in better R-D performance, but requires longer coding delay and more memory [35] which may be unacceptable in real-time video applications. With smaller GOP sizes, the boundary effect becomes significant, that is the peak signal-to-noise ratio (PSNR) values decrease somewhat abruptly at the GOP boundaries [36]. As a result, this can potentially degrade the overall coding performance. The dip in PSNRs at the GOP boundaries will be discussed in further details in the next chapter. It has been demonstrated in [35] that a reasonable choice for the GOP size would be 16 frames. In our experimental setup, we also choose 16 frames in each GOP.

After the 3-D wavelet decomposition, the wavelet coefficients are then grouped into a 3-D tree structure. Figure 2.11 demonstrates the tree structure of the spatio-temporal (s-t) relation for the 3-D SPIHT compression algorithm. The extension from the 2-D case is to form a node in 3-D SPIHT as a cube consisting of 2 x 2 x 2 adjacent wavelet coefficients, and each coefficient inside the node, except for those in the highest

and lowest frequency subbands, has eight descendants in the next higher level. However, the offspring branching rule is different for the lowest frequency subband, where one of the coefficients in each node has no descendants. Hence, a simple extension of the parent-offspring linkage to 3-D hierarchical tree, except at the highest and lowest frequency subbands, can be defined as the following:

$$\begin{aligned} O(i,j,k) = \{&(2i,2j,2k),(2i,2j+1,2k), \\ &(2i+1,2j,2k),(2i+1,2j+1,2k), \\ &(2i,2j,2k+1),(2i+1,2j,2k+1), \\ &(2i,2j+1,2k+1),(2i+1,2j+1,2k+1)\} \end{aligned} \tag{2.3}$$

where $O(i,j,k)$ represents a set of coordinates of all the offspring of the wavelet coefficient at location (i,j,k).

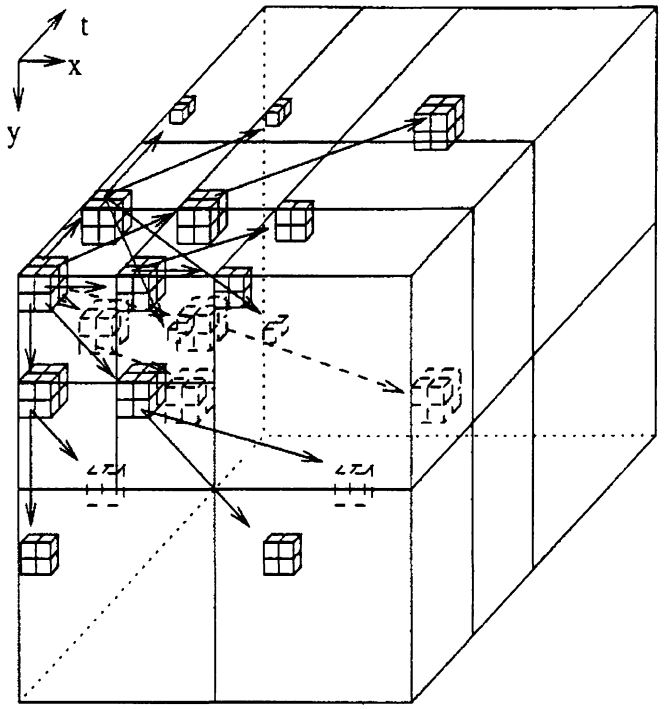


Figure 2.11: Spatio-temporal orientation tree for 3-D SPIHT [36].

Furthermore, the coding procedure of the 3-D SPIHT is similar to that of the 2-D SPIHT: initialization, sorting pass, and then refinement pass. The only difference between 2-D and 3-D is the sorting of the tree structure. A video compression coding system based on 3-D SPIHT showing the encoding and decoding procedures is illustrated in Figure 2.12. The block diagram in Figure 2.12 consists of three parts: 3-D wavelet transform, 3-D SPIHT coder, and entropy coding. The basic procedure is that a segment of a video sequence to be coded is first subband transformed. After 3-D DWT transformation, the 3-D SPIHT algorithm is applied to the resulting multiresolution pyramid. Then, the entropy encoder can be selectively used to further compress the output bitstream. The decoder does exactly the same operations but in the opposite

direction: entropy decoding, then 3-D SPIHT decoding, and finally inverse wavelet transformation.

This efficient source coding capability along with many desirable features, such as its full embeddedness for progressive transmission and low computational complexity, make the 3-D SPIHT an attractive candidate for multimedia applications.

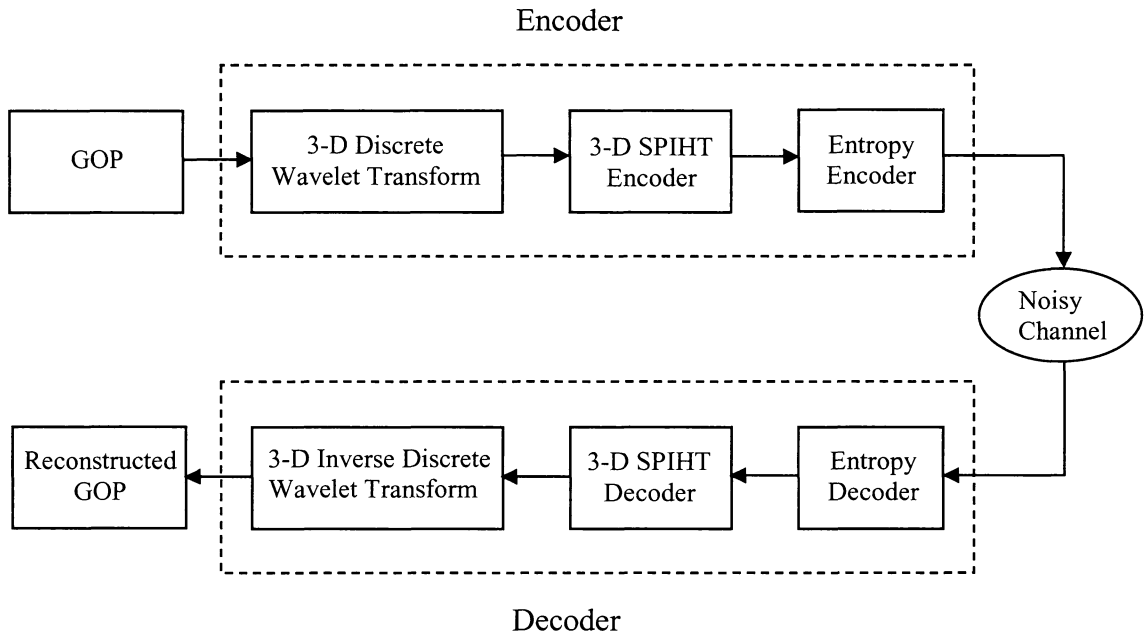


Figure 2.12: 3-D SPIHT video coding system.

2.6 STTP-SPIHT

As we described from the previous section, the 3-D SPIHT coder has proved its efficiency and its real-time capability in video compression. However, one major drawback of the 3-D SPIHT algorithm is that the encoded bitstreams are extremely sensitive to data losses due to the dependence among wavelet coefficients in constructing

a significance map. A single-bit transmission error may lead to a loss of synchronization between the encoder and decoder execution paths, which would lead to a total collapse of decoded video quality. In other words, when a single bit error occurs in a bit conveying significance of a wavelet coefficient or a set of wavelet coefficients, the decoding algorithm deviates from the encoder's execution path, giving erroneously decoded data beyond the point of the error. Therefore, it is very important to improve the error-resilience in the 3-D SPIHT algorithm so that we can achieve robust video transmission over packet loss channels.

In recent years, numerous sophisticated MD video coders have been proposed in the literature to make video transmission resilient to channel errors. In this section, we will specifically present a very successful domain-partitioning based MD coding algorithm called STTP-SPIHT, and it provides the basis for our thesis. Cho and Pearlman developed the STTP-SPIHT coding scheme, derived from Creusere's earlier work with images [14],[15], for partitioning the 3-D wavelet transform coefficients into independent coding units, so that an error in any one unit does not affect the others [18],[19]. This method efficiently exploits the spatial similarity inside each frame as well as the temporal similarity between frames.

The STTP-SPIHT coding method has been proven to provide excellent results in both noisy and noiseless channel conditions while preserving all the desirable properties of the 3-D SPIHT algorithm. The basic concept of this error resilient video compression algorithm is to partition the 3-D wavelet coefficients into some number P of different groups according to their spatial and temporal relationships, and each group is then

independently encoded using the 3-D SPIHT algorithm to create P independent embedded 3-D SPIHT substreams.

In Figure 2.13, we demonstrate an example of partitioning the 3-D wavelet transform coefficients into four independent groups ($P = 4$), denoted by 'a', 'b', 'c', and 'd', and each group retains the same spatial-temporal tree structure as the 3-D SPIHT algorithm. The s-t blocks represented by 'a', 'b', 'c', and 'd' correspond to the top left, top right, bottom left, and bottom right portions of the image sequences, respectively. Each of these s-t blocks is then independently encoded using the 3-D SPIHT algorithm, so that four independent embedded 3-D SPIHT substreams are created. Furthermore, these four bitstreams are interleaved in blocks to produce the final embedded STTP-SPIHT bitstream. The advantage of coding the wavelet coefficients with multiple and independent bitstreams is that a single bit error affects only one of the P substreams while the others are received unaffected. This basically implies that any decoding failure in one substream only affects the associated s-t region in the GOP rather than the full extent of GOP. Thus, the wavelet coefficients represented by a corrupted bitstream are reconstructed at reduced accuracy, and those represented by the error-free streams are reconstructed at full encoder accuracy. As we can see, this substantially increases channel error robustness over a wide range of bit error rates.

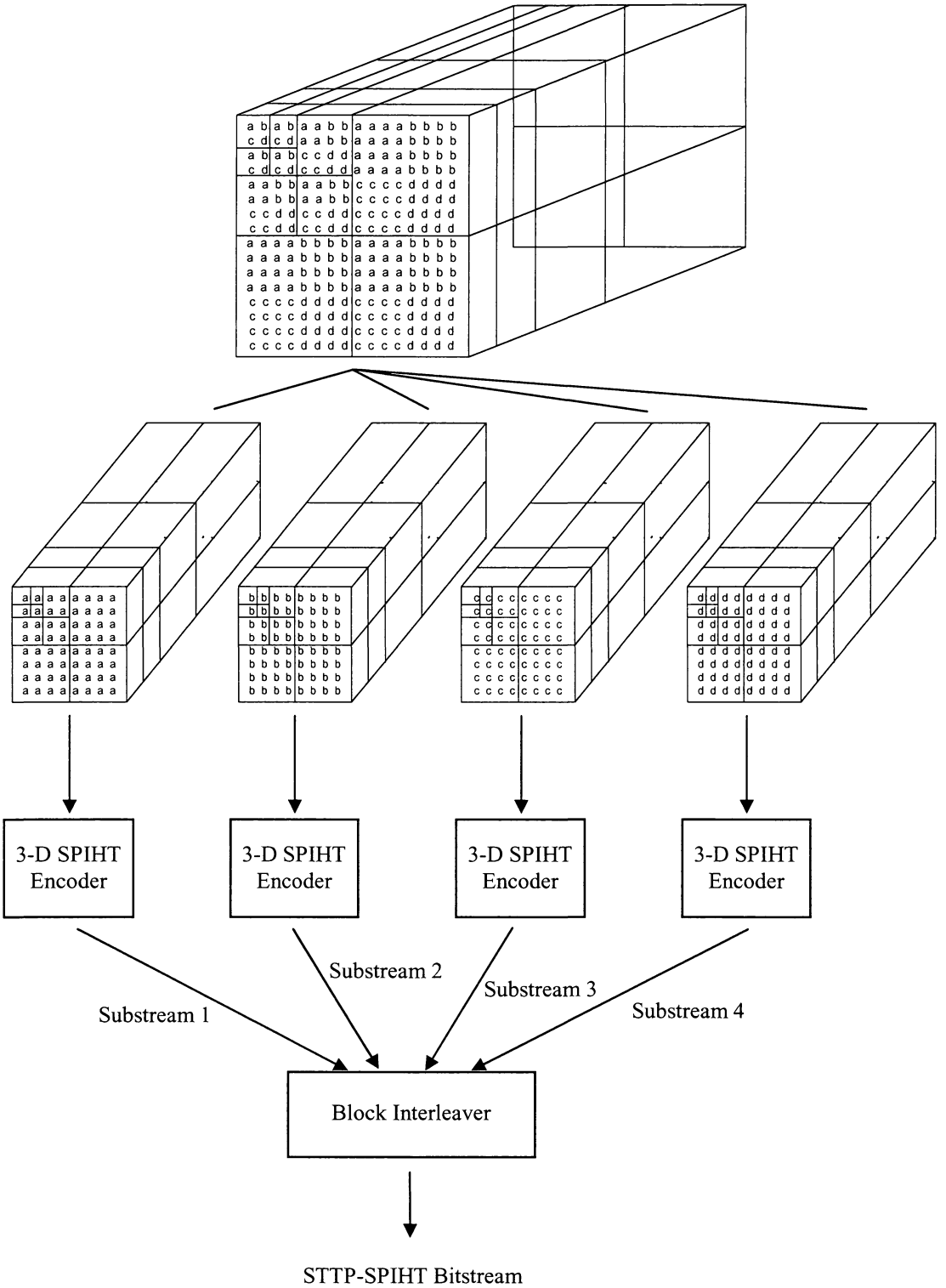


Figure 2.13: Structure of the 3-D STTP-SPIHT video compression algorithm [19].

2.7 Summary

DWT has gained widespread acceptance as a transform tool used in compression algorithms due to its characteristics of multiresolution analysis and nonblock-based analysis, which are different from the conventional transforms such as the DCT. The use of DWT in image coding applications has become much more interesting after Shapiro proposed his popular zerotree structure. Furthermore, the wavelet zerotree image coding technique provided the basis for the development of the 2-D and 3-D SPIHT algorithms. Later, Cho and Pearlman proposed the STTP-SPIHT method which is more robust against channel bit errors in comparison to the 3-D SPIHT algorithm, and this coding method has direct relevance to our thesis.

In the next chapter, we extend upon Cho and Pearlman's work [19], and propose a domain-partitioning based video coding algorithm which is more resilient to random channel bit errors.

Chapter 3: Proposed Algorithm

3.1 Introduction

In the previous chapter, we presented some useful background information which included the DCT-based hybrid video coding, the wavelet transform, the 3-D SPIHT algorithm, and the STTP-SPIHT coding scheme. In this chapter, we will present our proposed domain-partitioning based MD video coding algorithm which is derived from the STTP-SPIHT coding algorithm [19] presented in section 2.6. Although the STTP-SPIHT coding method achieves higher resilience against random channel bit errors when compared to the 3-D SPIHT algorithm, it is still quite susceptible to very early decoding failure, which results in one or more small regions with lower resolution than the surrounding area. In some cases, these artifacts occur in an important region causing visual discomfort. To avoid this, early decoding errors should be prevented so as to guarantee a minimum quality of the whole region.

As we recall from our previous discussions of the 3-D SPIHT algorithm in chapter 2, the earlier parts of the SPIHT-encoded bitstream contribute most to the reconstructed picture quality, and the later parts of the bitstream cannot be decoded without the earlier parts. Hence, the earlier parts of the bitstream for each GOP are much more important than the later parts. In other words, the wavelet coefficients inside the root subband are the most important among all the other coefficients since that is where

most of the signal energy is concentrated. A small error in the estimation of missing coefficients in the root subband may have a large impact on the overall distortion. Therefore, it is crucial to carefully protect these wavelet coefficients against transmission errors, and one common approach to achieve this goal is to inject additional redundancy into the multiple substreams or descriptions of the encoded original video sequence. Using these observations, we attempt to improve upon the STTP-SPIHT algorithm by intentionally inserting a certain amount of redundant data to protect those wavelet coefficients in the root subband, and this redundant information along with the correctly received substreams are then used to recover the missing coefficients of the root subband at the receiver side. However, redundant data also requires more bits to encode which results in reduced coding efficiency. Thus, our primary objective in designing our proposed MD video coder is to minimize redundancy while meeting an end-to-end distortion requirement that takes transmission loss into account.

We will focus on describing three specific design issues associated with our proposed video coding algorithm, and these issues will be discussed in further details in the following sub-sections:

1) The methods used to efficiently insert the additional redundancy in the MD coder so that the important wavelet coefficients in the root subband are well protected against transmission errors.
2) The modifications made to the implementation of the conventional 3-D SPIHT encoder and decoder in order to accommodate for the additional redundancy that is inserted into the multiple substreams.

3) The recovery mechanism used to recover the missing coefficients in the root subband at the decoder side.

3.2 System Overview

Before we proceed to the discussion of our proposed framework, we should describe the notations and assumptions which will be used throughout this chapter:

- For the purpose of all the illustrations and equations in this chapter, we choose to demonstrate the simplest scenario where four different substreams (P = 4) are created at the encoder. Furthermore, we assume that each GOP consists of sixteen frames (for the reasons stated in section 2.5.2), and the frames are spatially and temporally decomposed by three levels of wavelet decompositions. We also note that it is certainly possible to have higher values of P, and we will illustrate the cases of P = 4 and P = 10 in our simulation results in the next chapter.
- Let the symbol WC^{RS} represent the root subband, where WC denotes the wavelet transform coefficients resulting from an s-t wavelet transform. The root subband is defined by the shaded gray area illustrated in Figure 3.1, which is composed of the lowest s-t frequency subband in a GOP. The root subband can also be referred to as the approximation subband (as mentioned in section 2.3.2). Hence, the wavelet transform coefficients for the four independent groups (P = 4) are represented by WC_a, WC_b, WC_c, and WC_d.

- We create a new redundant data set using predictive coding, denoted by R_{e_1}, R_{e_2}, R_{e_3}, and R_{e_4}.
- We create another redundant data set which uses the predictive coding with a sub-pixel correction shift, denoted by S_{e_1}, S_{e_2}, S_{e_3}, and S_{e_4}.

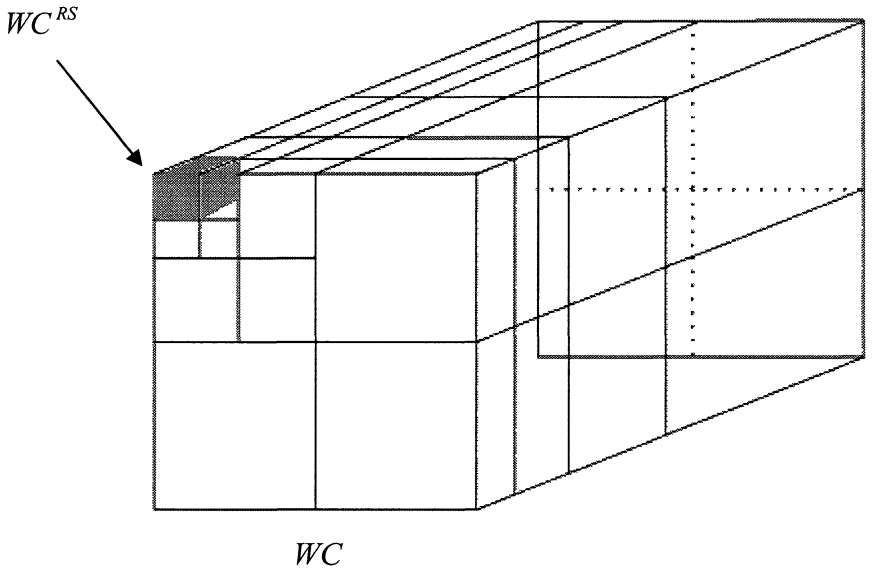


Figure 3.1: Illustration of the root subband.

A general framework of our proposed MD video coding method can be illustrated in Figure 3.2. In this framework, we implemented two additional features: insert redundant data at the encoder side and root subband recovery at the decoder side. The sixteen-frame segments are sequentially processed by the 3-D wavelet transform which consists of a temporal transformation followed by a spatial transformation ($t+2D$

structure). This basically means that all levels of temporal transformation are performed first on the GOP, and the spatial transformation is then applied to each temporal-transformed frame. After the subband/wavelet transformation, the next stage is to partition the 3-D wavelet transform coefficients into some number P of different groups according to their spatial and temporal relationships. We then intentionally insert redundant information into each group in order to protect those wavelet coefficients in the root subband. Each group is encoded independently using the modified 3-D SPIHT algorithm, so that P independent embedded 3-D SPIHT substreams are created. The P substreams are then interleaved in appropriate fixed size units, such as packets, prior to transmission. We note that the implementation of the conventional 3-D SPIHT encoder and decoder has to be modified to account for the additional redundancy. We will describe the implementation details of the modified 3-D SPIHT coder used in our proposed algorithm in the next section. After transmitting the packets across noisy channels, the interleaved bitstream will be first de-interleaved, and each substream is decoded independently using the modified 3-D SPIHT decoder. By using the additional redundant data, the root subband recovery technique is then applied to recover the missing coefficients in the lost subbstream. After the substreams have been independently decoded, the decoded wavelet coefficients are reordered according to their spatial and temporal relationships. Finally, the inverse 3-D wavelet transformation is applied to produce the final reconstructed GOP.

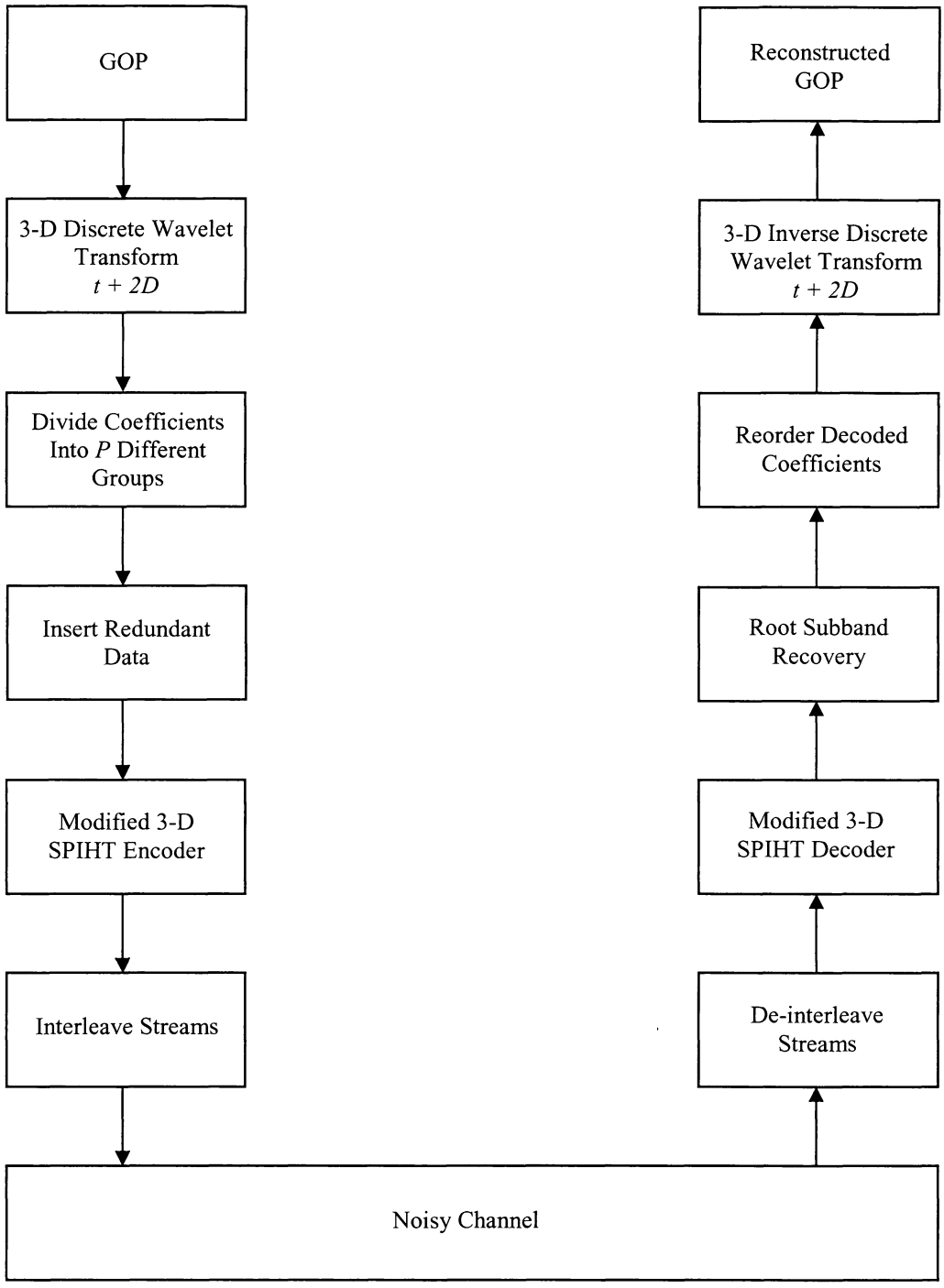


Figure 3.2: General framework of proposed MD video coding method.

In the next section, we will discuss the design of our proposed MD video coder in further detail which includes domain partitioning, the use of predictive coding and sub-pixel shifted predictive coding to efficiently insert the redundant data into the multiple substreams of the encoded original video sequence in order to protect the root subband against transmission errors, the modifications made to the conventional 3-D SPIHT encoder and decoder so that the coder can incorporate the redundant information, and finally the root subband recovery technique used at the receiver side to recover the missing coefficients in the lost substream.

3.3 Error Resilient Multiple Description Coder

3.3.1 Domain Partitioning

In our proposed domain-partitioning based MD video coding algorithm, we first partition the 3-D wavelet transform coefficients WC into P different groups according to their spatial and temporal relationships, which uses the same partitioning rule as that of the STTP-SPIHT coding method [19]. An example of partitioning the wavelet transform coefficients WC into four different groups (P = 4) is illustrated in Figure 3.3. As we mentioned earlier in our system overview, we choose to only illustrate the case of P = 4 for simplicity. The four independent groups are denoted by WC_a, WC_b, WC_c, and WC_d, and each of which retains the s-t tree structure of 3-D SPIHT [19]. The 3-D SPIHT video compression algorithm is just the case when P = 1. In the case of P = 10, the spatial dimensions of the root subband is 5 x 2 instead of 2 x 2. As was mentioned previously in chapter 2, by coding the wavelet coefficients with multiple and independent substreams,

the errors in any one of the P substreams are isolated and do not affect the other substreams. Additionally, missing coefficients in one lost substream can be estimated at the decoder from the other correctly received substreams.

Each substreams has its own SPIHT header information. These headers are necessary for the receiver to decode the substreams correctly, and should be carefully protected from channel errors. The header information for each substream includes the following: the initial thresholds of video data and redundant data, the spatial and temporal decomposition levels, the dimensions of WC, and the number of substreams P and the substream index. Moreover, as the number of substreams P is increased, the total size of the header information is also increased. In the case of lower bit rates, the increasingly significant size of overhead information compared to data information will be detrimental to video quality. One possible resolution is to use the concept of a global header [19]. Using this idea, we can place the common parameters in the global header. The benefit of using the global header is to reduce the overall size of the overhead information and to enable the encoder to use more bits to encode video sequence at the same bit rate. We assume in our thesis that the global header is received correctly at the decoder.

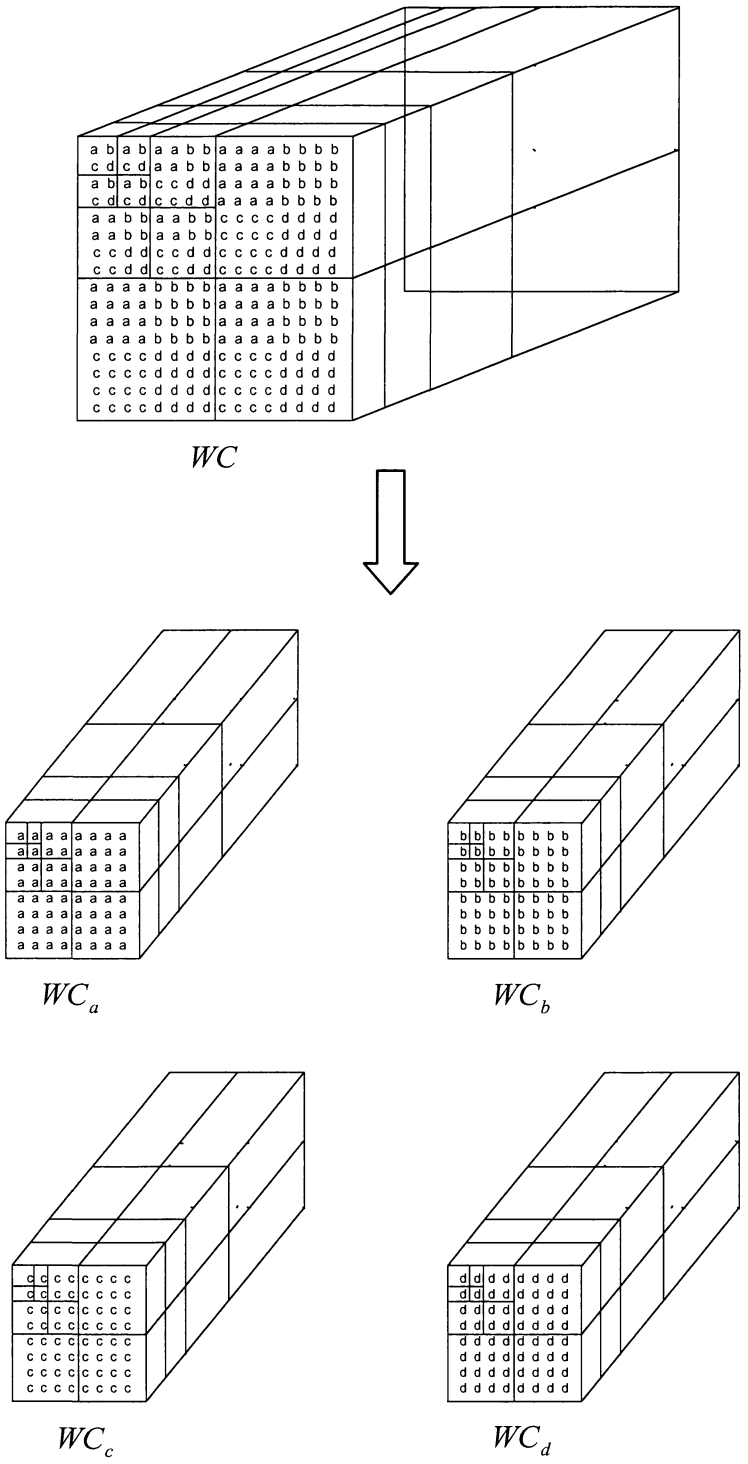


Figure 3.3: An example of partitioning the 3-D wavelet transform coefficients into four independent groups ($P = 4$).

3.3.2 Redundancy Insertion using Predictive Coding

In the next stage, we need to insert the redundant information into each substream of the original video sequence in order to protect the important wavelet coefficients in the root subband WC^{RS} from channel errors. Typically, in a video sequence, the values of adjacent wavelet coefficients inside the approximation subband of each frame are highly correlated, and hence we can obtain a great deal of information about a coefficient value by inspecting its neighbouring coefficient values. This property is exploited in predictive coding where an attempt is made to predict the value of a given coefficient based on the values of its surrounding coefficients. Therefore, we propose to apply the concept of predictive coding [37]-[40], also known as differential coding, to calculate the redundant data set in our proposed algorithm, since this prediction method is computationally simple and efficiently exploits the inherent spatial correlation inside each frame to estimate the missing wavelet coefficients at the decoder. Essentially, predictive coding removes spatial redundancy between the adjacent coefficients inside each frame, and only encodes the difference between the neighbouring coefficients, referred to as the prediction error. The prediction errors tend to have a smaller dynamic range than that of the original coefficient values, and hence can be coded more efficiently [41]. As a result, we decide to insert the prediction error as the redundant information into the multiple substreams. The predictive coding concept is based on the principle that there is significant correlation among the neighbouring coefficients within each frame. Thus, at the receiver side, we can use the prediction residual along with the other correctly received substreams to predict the missing coefficients in the root subband of the lost substream.

In Figure 3.4, we illustrate three configurations used in our thesis to determine the prediction error, but other configurations are certainly possible as well. Again, for simplicity, we show the case for four different groups ($P = 4$). We specifically choose these three configurations so that all directions (horizontal, vertical, and diagonal) are considered when calculating the difference between adjacent coefficients inside each frame. The characters 'a', 'b', 'c', and 'd' represent those wavelet transform coefficients in the root subband. In addition, the arrows indicate how we perform subtraction between the two neighbouring coefficients. As we can clearly see from Figure 3.4, configurations 2 and 3 have a definite advantage over configuration 1 in their ability to reconstruct the missing coefficients in the root subband when both substreams *a* and *b* are lost or when both substreams *c* and *d* are lost due to transmission errors. One potential disadvantage of configuration 3 is that there is probably going to be less correlation between neighbouring coefficients in the diagonal directions because of the longer distance between them.

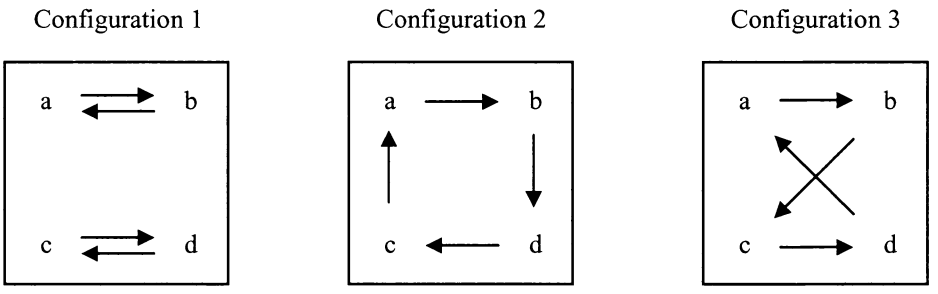


Figure 3.4: Three configurations used to determine the prediction error for the case of four independent groups ($P = 4$).

In configuration 1, the redundant data set, represented by $R_{e_1}^{C1}$, $R_{e_2}^{C1}$, $R_{e_3}^{C1}$, and $R_{e_4}^{C1}$, can be obtained by taking the difference between two adjacent groups of wavelet coefficients in the horizontal direction as indicated by the arrows:

$$\begin{aligned} R_{e_1}^{C1} &= WC_b^{RS} - WC_a^{RS} \\ R_{e_2}^{C1} &= WC_a^{RS} - WC_b^{RS} \\ R_{e_3}^{C1} &= WC_d^{RS} - WC_c^{RS} \\ R_{e_4}^{C1} &= WC_c^{RS} - WC_d^{RS} \end{aligned} \tag{3.1}$$

where characters 'a', 'b', 'c', and 'd' represent the wavelet coefficients with spatial and temporal relationships as defined in Figure 3.3.

Similarly, in configuration 2, the redundant data set can be obtained by taking the difference between two adjacent groups of wavelet coefficients in the horizontal or vertical direction:

$$\begin{aligned} R_{e_1}^{C2} &= WC_b^{RS} - WC_a^{RS} \\ R_{e_2}^{C2} &= WC_d^{RS} - WC_b^{RS} \\ R_{e_3}^{C2} &= WC_a^{RS} - WC_c^{RS} \\ R_{e_4}^{C2} &= WC_c^{RS} - WC_d^{RS} \end{aligned} \tag{3.2}$$

Finally, in configuration 3, the redundant data set can be obtained by taking the difference between two adjacent groups of wavelet coefficients in the horizontal or diagonal direction:

$$R_{e_1}^{C3} = WC_b^{RS} - WC_a^{RS} \tag{3.3}$$

$$R_{e_2}^{C3} = WC_c^{RS} - WC_b^{RS}$$

$$R_{e_3}^{C3} = WC_d^{RS} - WC_c^{RS}$$

$$R_{e_4}^{C3} = WC_a^{RS} - WC_d^{RS}$$

3.3.3 Redundancy Insertion using Predictive Coding with Sub-Pixel Correction Shift

We can further exploit the spatial correlation between neighbouring wavelet coefficients in each frame by making a modification to the calculation of prediction error, as stated above in section 3.3.2. This can be accomplished by performing a sub-pixel shift, in the direction of the arrows illustrated for the different configurations in Figure 3.4, on one group of coefficients prior to calculating the difference between two independent groups. For example, in configuration 1, before we take the difference between groups 'b' and 'a' in the horizontal direction, we first perform a horizontal half-pixel shift on group 'a'. Since the four independent groups 'a', 'b', 'c', and 'd' are sub-sampled versions of the approximation subband, illustrated in Figure 3.5, a horizontal full-pixel shift in the approximation subband is equivalent to a horizontal half-pixel shift in the context of sub-sampled results. Similarly, in configuration 2, we can perform a vertical half-pixel shift when taking the difference between two different groups in the vertical direction. In configuration 3, the correction shift will be larger than half since the length in the diagonal direction is longer than that of the horizontal and vertical directions. Hence, we obtain another set of redundant data, represented by *S*. Furthermore, we will

call the sub-pixel shifted versions of configurations 1, 2, and 3 as configurations 4, 5, and 6, respectively.

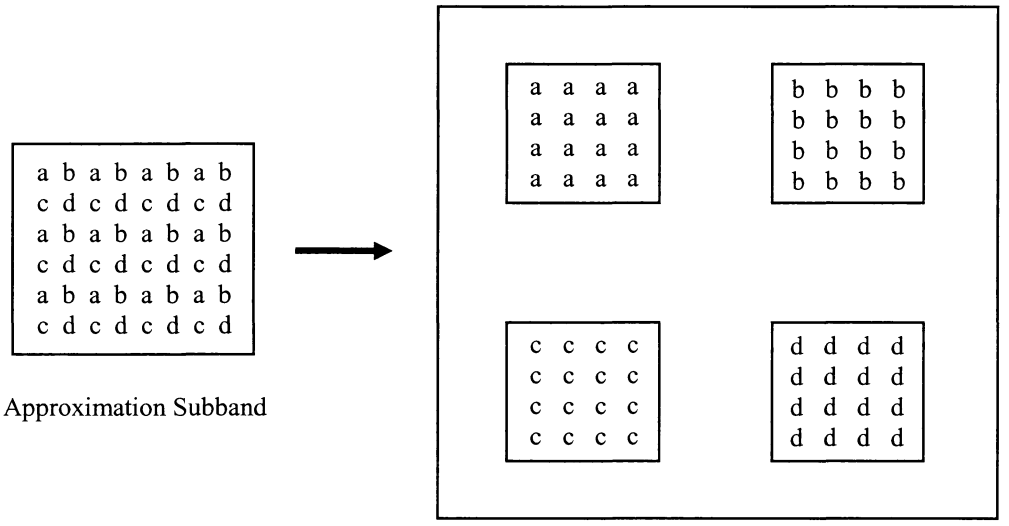


Figure 3.5: Sub-sampling of the approximation subband.

In configuration 4, the new redundant data set, denoted by $S_{e_1}^{C4}$, $S_{e_2}^{C4}$, $S_{e_3}^{C4}$, and $S_{e_4}^{C4}$, can be calculated by taking the difference between one group of wavelet coefficients and its sub-pixel shifted neighbouring group of wavelet coefficients in the horizontal direction:

$$S_{e_1}^{C4} = WC_b^{RS} - horizontal\_half\_pixel\_shift(WC_a^{RS}) \tag{3.4}$$

$$S_{e_2}^{C4} = WC_a^{RS} - horizontal\_half\_pixel\_shift(WC_b^{RS})$$

$$S_{e_3}^{C4} = WC_d^{RS} - horizontal\_half\_pixel\_shift(WC_c^{RS})$$

$$S_{e_4}^{C4} = WC_c^{RS} - horizontal\_half\_pixel\_shift(WC_d^{RS})$$

where the characters 'a', 'b', 'c', and 'd' represent the wavelet coefficients with spatial and temporal relationships as defined in Figure 3.3.

Similarly, in configuration 5, the new redundant data set S can be calculated by taking the difference between a group of wavelet coefficients and its sub-pixel shifted neighbouring wavelet coefficients in the horizontal or vertical direction:

$$S_{e_1}^{C5} = WC_b^{RS} - horizontal\_half\_pixel\_shift(WC_a^{RS}) \tag{3.5}$$

$$S_{e_2}^{C5} = WC_d^{RS} - vertical\_half\_pixel\_shift(WC_b^{RS})$$

$$S_{e_3}^{C5} = WC_a^{RS} - vertical\_half\_pixel\_shift(WC_c^{RS})$$

$$S_{e_4}^{C5} = WC_c^{RS} - horizontal\_half\_pixel\_shift(WC_d^{RS})$$

Finally, in configuration 6, the new redundant data set S can be calculated by taking the difference between a group of wavelet coefficients and its sub-pixel shifted neighbouring wavelet coefficients in the horizontal or vertical direction:

$$S_{e_1}^{C6} = WC_b^{RS} - horizontal\_half\_pixel\_shift(WC_a^{RS}) \tag{3.6}$$

$$S_{e_2}^{C6} = WC_c^{RS} - diagonal\_sub\_pixel\_shift(WC_b^{RS})$$

$$S_{e_3}^{C6} = WC_d^{RS} - horizontal\_half\_pixel\_shift(WC_c^{RS})$$

$$S_{e_4}^{C6} = WC_a^{RS} - diagonal\_sub\_pixel\_shift(WC_d^{RS})$$

In both redundancy insertion methods described above, the prediction residual along with the multiple substreams are sent to the decoder for reconstruction of the video sequence. In the next chapter, we will perform a comparative analysis of the experimental results obtained by utilizing the two different redundant data sets R and S in the noiseless and noisy environments. Figure 3.6 demonstrates the basic structure of our proposed domain-partitioning based MD video coding algorithm. As we can see from Figure 3.6, the wavelet transform coefficients inside the redundant data set have no descendants along the spatial or temporal directions, since these wavelet coefficients are generated by the coefficients inside the root subband. Due to the additional redundancy inserted into the different substreams, we must modify the conventional 3-D SPIHT encoder and decoder to accommodate this change. In the next section, we will describe the implementation details of our modified 3-D SPIHT coder.

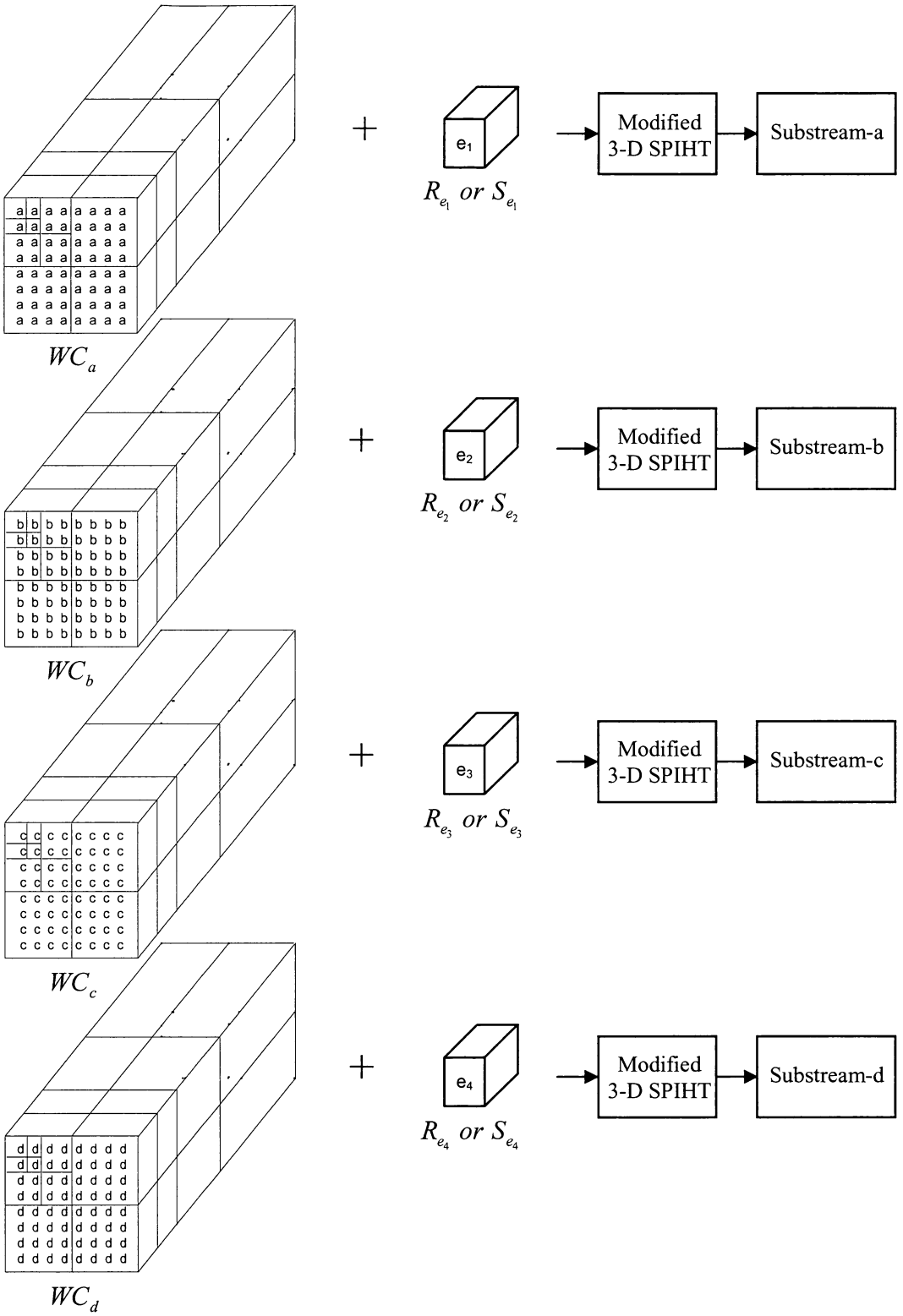


Figure 3.6: Structure of the proposed MD video coder.

3.3.4 Modified 3-D SPIHT Coder Implementation

The 3-D SPIHT algorithm is applied to the different groups of wavelet transform coefficients so that P independent embedded 3-D SPIHT substreams are created. Since we have inserted additional redundant data to protect the root subband in each substream, we must modify the 3-D SPIHT encoder to accommodate this change. Conventionally, the 3-D SPIHT algorithm is applied only to those wavelet coefficients from the original video sequence. However, in our proposed MD coding algorithm, we modified the 3-D SPIHT encoder so that the 3-D SPIHT algorithm is applied to not only the wavelet coefficients in the original video sequence WC, but also to the wavelet coefficients in the redundant data, either R or S, as shown in Figure 3.6. This additional redundant information is intentionally inserted into the multiple substreams to improve the resilience against transmission errors.

As we stated earlier in section 2.5.2, the 3-D SPIHT encoder maintains three ordered lists for the wavelet coefficients in the original video sequence WC_x where $x \in \{a,b,c,d\}$: a LIS, a LIP, and a LSP. We now modify the 3-D SPIHT encoder so that it maintains three additional ordered lists specifically designated for the wavelet coefficients in the redundant data R_x or S_x where $x \in \{e_1,e_2,e_3,e_4\}$: a LIS, a LIP, and a LSP. Similar to the conventional 3-D SPIHT algorithm, the modified version of the 3-D SPIHT encoder also consists of the following three stages: initialization, sorting pass, and refinement pass. At the initialization stage, the three ordered lists for the wavelet coefficients in WC_x and the extra three ordered lists for the wavelet coefficients in R_x or S_x are all initialized in the same manner as the 3-D SPIHT algorithm, described in

section 2.5.2. In addition, the initial threshold n of the modified 3-D SPIHT encoder is now defined as follows for the redundant data set R_τ:

$$n = \max(n_{WC_\tau}, n_{R_\tau}) \tag{3.7}$$

or for the redundant data set S_x

$$n = \max(n_{WC_\tau}, n_{S_\tau}) \tag{3.8}$$

where n_{WC_τ}, n_{R_τ}, and n_{S_τ} represent the initial thresholds for groups WC_τ, R_x, and S_x, respectively. The threshold n is simply the larger of n_{WC_τ} and n_{R_τ}. In the next step, we apply the sorting and refinement passes to the groups WC_x and R_τ separately so that the multiple groups are encoded independently at the same bit rate. After each pass, the bitstreams generated for the coefficients in R_x is concatenated to the end of the bitstreams generated for the coefficients in WC_τ. The refinement pass is performed in the same fashion where it refines the entries found in the LSP, except for those coefficients added in the most recent sorting pass, with one additional bit of precision. Similarly, once the refinement pass has completed, the current threshold is divided by two (n is decremented by one), and the sorting and refinement passes are applied again at the lowered threshold. The process continues through successive halving of the threshold until a specified rate constraint or a distortion requirement is reached.

3.3.5 Root Subband Recovery

In this section, we will describe our root subband recovery algorithm at the decoder side. Due to transmission errors, we will sometimes lose a part or all of a substream. For example, let us consider a scenario where we lose an entire substream due to an early decoding error. As a result, all the wavelet transform coefficients in this particular substream are rendered useless, and hence they will all be set to zero. To restore most or all of the missing coefficients in the root subband of this one lost substream, we must use the other three correctly received substreams along with the redundant information, either R or S as discussed above in sub-sections 3.3.2 and 3.3.3. Mathematically, for configuration 1, we can recover the missing coefficients in the root subband of any corrupted substream by performing the following calculations:

$$\begin{aligned} WC_a^{RS'} &= R_{e_2}^{C1'} + WC_b^{RS'} \\ WC_b^{RS'} &= R_{e_1}^{C1'} + WC_a^{RS'} \\ WC_c^{RS'} &= R_{e_4}^{C1'} + WC_d^{RS'} \\ WC_d^{RS'} &= R_{e_3}^{C1'} + WC_c^{RS'} \end{aligned} \tag{3.9}$$

where $WC^{RS'} = \left\{ WC_a^{RS'}, WC_b^{RS'}, WC_c^{RS'}, WC_d^{RS'} \right\}$ and $R_e^{C1'} = \left\{ R_{e_1}^{C1'}, R_{e_2}^{C1'}, R_{e_3}^{C1'}, R_{e_4}^{C1'} \right\}$ represent the four different groups and the redundant data set reconstructed from the received substreams, respectively.

Similarly, the missing coefficients in the root subband of lost substream can be recovered by the following calculations for configuration 2:

$$WC_a^{RS'} = R_{e_3}^{C2'} + WC_c^{RS'}$$
$$WC_b^{RS'} = R_{e_1}^{C2'} + WC_a^{RS'}$$
$$WC_c^{RS'} = R_{e_4}^{C2'} + WC_d^{RS'}$$
$$WC_d^{RS'} = R_{e_2}^{C2'} + WC_b^{RS'} \quad (3.10)$$

For configuration 3:

$$WC_a^{RS'} = R_{e_4}^{C3'} + WC_d^{RS'}$$
$$WC_b^{RS'} = R_{e_1}^{C3'} + WC_a^{RS'}$$
$$WC_c^{RS'} = R_{e_2}^{C3'} + WC_b^{RS'}$$
$$WC_d^{RS'} = R_{e_3}^{C3'} + WC_c^{RS'} \quad (3.11)$$

In addition, in the case where we use the redundant data set S (the sub-pixel shifted version), we would then recover the missing coefficients in the root subband of the lost substream for configuration 4 with the following calculations:

$$WC_a^{RS'} = S_{e_2}^{C4'} + horizontal\_half\_pixel\_shift(WC_b^{RS'})$$
$$WC_b^{RS'} = S_{e_1}^{C4'} + horizontal\_half\_pixel\_shift(WC_a^{RS'})$$
$$WC_c^{RS'} = S_{e_4}^{C4'} + horizontal\_half\_pixel\_shift(WC_d^{RS'})$$
$$WC_d^{RS'} = S_{e_3}^{C4'} + horizontal\_half\_pixel\_shift(WC_c^{RS'}) \quad (3.12)$$

where $WC^{RS'} = \left\{ WC_a^{RS'}, WC_b^{RS'}, WC_c^{RS'}, WC_d^{RS'} \right\}$ and $S_e' = \left\{ S_{e_1}^{C4'}, S_{e_2}^{C4'}, S_{e_3}^{C4'}, S_{e_4}^{C4'} \right\}$

represent the four different groups and the redundant data set reconstructed from the received substreams, respectively.

Similarly, the missing coefficients in the root subband of lost substream can be recovered by the following calculations for configuration 5:

$$\begin{aligned} WC_a^{RS'} &= S_{e_3}^{C5'} + vertical\_half\_pixel\_shift(WC_c^{RS'}) \\ WC_b^{RS'} &= S_{e_1}^{C5'} + horizontal\_half\_pixel\_shift(WC_a^{RS'}) \\ WC_c^{RS'} &= S_{e_4}^{C5'} + horizontal\_half\_pixel\_shift(WC_d^{RS'}) \\ WC_d^{RS'} &= S_{e_2}^{C5'} + vertical\_half\_pixel\_shift(WC_b^{RS'}) \end{aligned} \tag{3.13}$$

For configuration 6:

$$\begin{aligned} WC_a^{RS'} &= S_{e_4}^{C6'} + diagonal\_sub\_pixel\_shift(WC_d^{RS'}) \\ WC_b^{RS'} &= S_{e_1}^{C6'} + horizontal\_half\_pixel\_shift(WC_a^{RS'}) \\ WC_c^{RS'} &= S_{e_2}^{C6'} + diagonal\_sub\_pixel\_shift(WC_b^{RS'}) \\ WC_d^{RS'} &= S_{e_3}^{C6'} + horizontal\_half\_pixel\_shift(WC_c^{RS'}) \end{aligned} \tag{3.14}$$

3.4 Summary

In this chapter, we introduced our proposed domain-partitioning based MD video coding algorithm to improve the error resilience of the 3-D SPIHT encoded bitstreams against transmission errors. The proposed algorithm is an extension to Cho and Pearlman's earlier work on the STTP-SPIHT algorithm [18],[19]. Furthermore, we presented the details around how we performed domain partitioning, redundancy insertion, modified 3-D SPIHT coder implementation, as well as root subband recovery. In the next chapter, we will conduct a series of experiments to investigate the performance of our proposed MD video coding algorithm under noiseless and noisy conditions. This is accomplished by performing a comparative analysis between the proposed algorithm and STTP-SPIHT.

Chapter 4: Simulation Results

4.1 Introduction

In this chapter, we conduct a series of experiments to examine the effectiveness of our proposed domain-partitioning based MD video coding algorithm. The simulation results obtained in each experiment of the proposed algorithm are then compared with those of the STTP-SPIHT video coder. We begin this chapter by describing the simulation parameters used in the various experiments. Next, we present a comparison of the simulation results in noiseless and noisy channels between our proposed algorithm and the STTP-SPIHT method, with respect to source coding efficiency and error resilience performance.

4.2 Experimental Setup

The 352 x 240 x 48 monochrome "Football" (frame number 0 – 47) and "Susie" (frame number 16 – 63) video sequences are specifically selected in the experiments, sampled at $F = 30$ frames per second (fps). The "Football" sequence has a relatively high level of motion, whereas the "Susie" sequence has a relatively low level of motion. We specifically chose these two monochrome video sequences because they are also the same sample video sequences used in Cho and Pearlman's earlier work on the STTP-SPIHT

algorithm [18],[19]. This way, we can make direct comparisons between our proposed algorithm and STTP-SPIHT based on the experimental results.

In our test of error resilience, we assume that the channel is a binary symmetric channel (BSC) with transition error probability p, as shown in Figure 4.1. In addition, we used 16 frames in each GOP, and applied a three-level wavelet transform using the popular Cohen-Daubechies-Feauveau 9/7-tap (CDF-9/7) biorthogonal wavelet filters [42] in both spatial and temporal domains with reflection extensions both at each image boundary and at the boundary of each GOP [36]. As we mentioned in section 2.5.2, a GOP of 16 frames provides a reasonable choice [35], since it results in better R-D performance than smaller GOP sizes, and at the same time it does not require the longer coding delay and more memory of the larger GOP sizes. In each of our test case scenarios, the 3-D wavelet transform coefficients from the original video sequence are partitioned into P different groups, and each group is independently encoded using the modified 3-D SPIHT algorithm at the same coding rate r, which we select as 1.0 bit per pixel (bpp) for our experiments. As a result, P independent embedded 3-D SPIHT substreams are created. In each of our experiments, the wavelet coefficients are partitioned into four or ten groups ($P = 4$ or $P = 10$). We note that it is certainly possible to choose higher values of P as well.

In order to calculate the total transmission bit rate of each substream, denoted by BR, we need to know the number of wavelet coefficients in the original video sequence WC of each frame. For instance, if the spatial dimensions of group WC_a is $X_a^{WC} \times Y_a^{WC}$, the total transmission bit rate of substream a is defined as follows:

$$BR_a = X_a^{WC} \times Y_a^{WC} \times F \times r \tag{4.1}$$

Equation 4.1 can be generalized to calculate the total transmission bit rate for any other group. Hence, the total transmission bit rate of the video sequence is then simply defined as follows:

$$BR = X^{WC} \times Y^{WC} \times F \times r \tag{4.2}$$

where $X^{WC} \times Y^{WC}$ denote the spatial dimensions of the original video sequence WC.

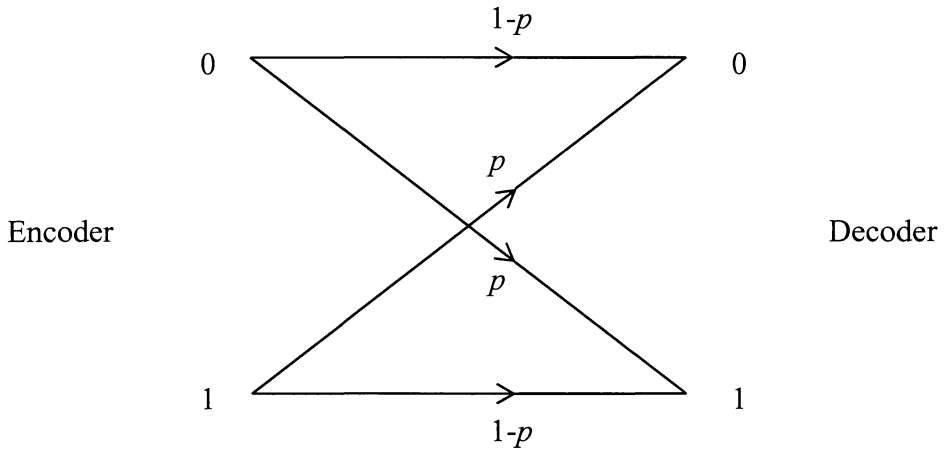


Figure 4.1: Transition probability diagram of binary symmetric channel.

As we mentioned earlier, we use one global header instead of using sub-headers for each substream, and we also assume that the SPIHT global header is not corrupted

from any bit errors. The bitstreams are partitioned into 200-bit equal length segments, and each segment is then passed through a cyclic redundancy code (CRC) parity checker to generate 16 parity bits. We use the same parameters as that of the STTP-SPIHT algorithm [19] to produce the results for our proposed coding algorithm in noiseless and noisy channels. Finally, the quality of the reconstructed video is measured based on the PSNR, which can be defined as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)dB \tag{4.3}$$

where *MSE* denotes the mean-squared error between the original and reconstructed video sequences. All PSNR values reported in the case of noisy channels are averages over 50 independent runs.

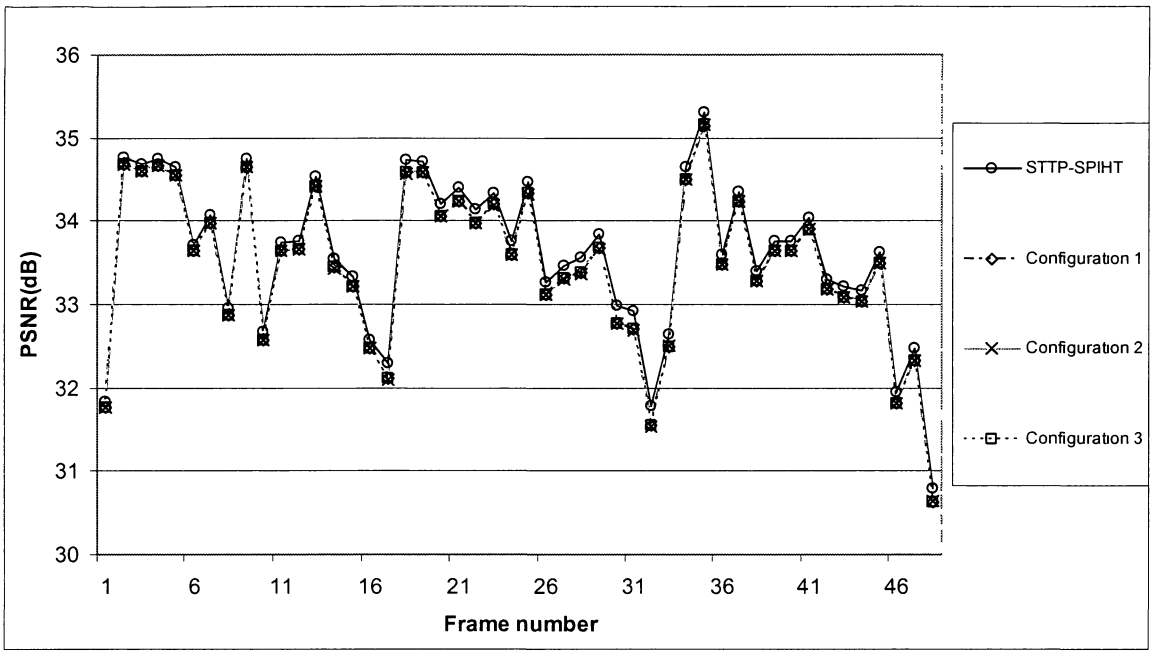
4.3 Source Coding Efficiency

In this section, we will perform a comparative analysis of our proposed domain-partitioning based MD video coding algorithm and the STTP-SPIHT algorithm in noiseless and noisy channels. The proposed algorithm consists of two ways of inserting redundant information: predictive coding and predictive coding with a sub-pixel correction shift, as discussed in detail in sub-sections 3.3.2 and 3.3.3.

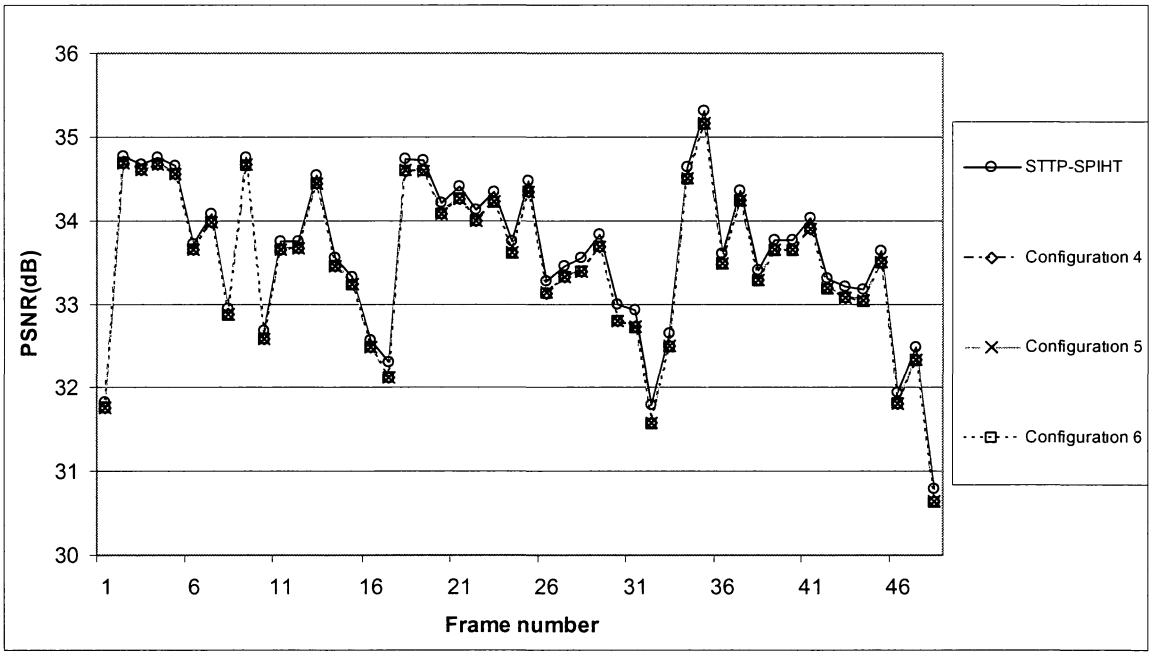
First of all, we begin by considering the case where the channel is error-free, and all P substreams are received correctly at the decoder. Figure 4.2 and Figure 4.3 illustrate the frame by frame comparison of PSNR (dB) values for the six configurations of the

proposed MD video coding algorithm and STTP-SPIHT for the 352 x 240 x 48 monochrome "Football" video sequence at a total transmission rate of 2.53 Mbps in noiseless channels for four and ten substreams ($P = 4$ and $P = 10$). Figure 4.4 and Figure 4.5 illustrate the same scenario for the "Susie" video sequence. It is clear from Figure 4.2 through Figure 4.5 that the PSNR values of the proposed coding algorithm are very close to those of STTP-SPIHT at the same coding rate, and this is true for both values of P.

In addition, we can observe that the PSNR decreases at the GOP boundaries, and in our case, this occurs at the beginning and end of each 16 frame segment. The reason for this dip is partly due to object motion and partly due to the boundary extension for temporal filtering [34],[36]. These lower PSNR values reduce the average PSNR, and can potentially cause annoying jittering artifacts in video playback [43]. Some solutions have been reported in literature which may be applied to reduce the GOP boundary artifacts and improve the overall performance of the transform. The topic of reducing GOP-boundary artifacts in wavelet-based video coding is beyond the scope of this thesis. The interested reader is referred to [43]-[45] for additional details on those methods. The average PSNR values for four and ten substreams ($P = 4$ and $P = 10$) of the proposed algorithm and STTP-SPIHT are summarized in Table 4.1(a) and (b).

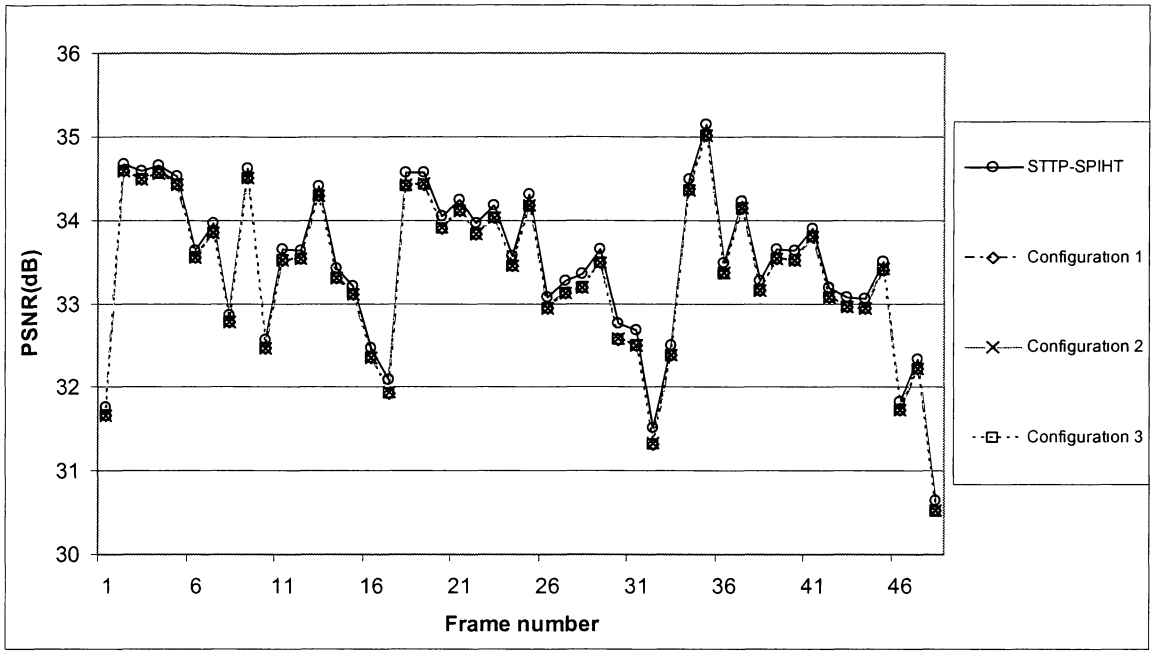


(a)

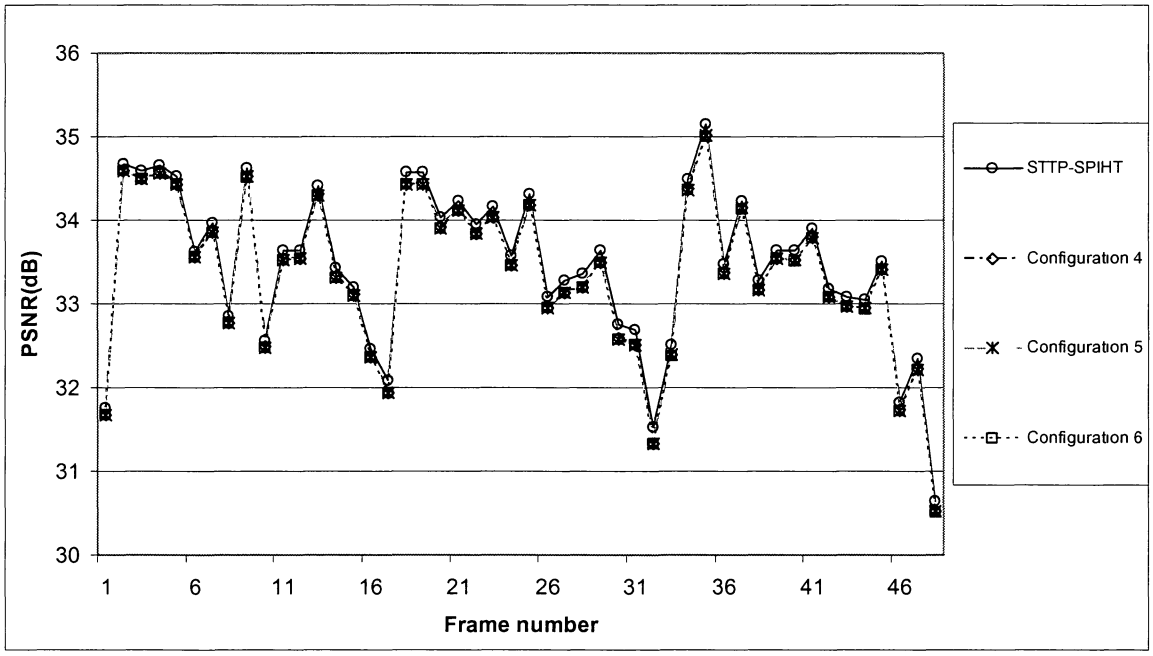


(b)

Figure 4.2: Frame by frame comparison of PSNR (dB) for "Football" video sequence in noiseless channels at a total transmission rate of 2.53 Mbps for $P = 4$. (a) Configurations 1 to 3. (b) Configurations 4 to 6.



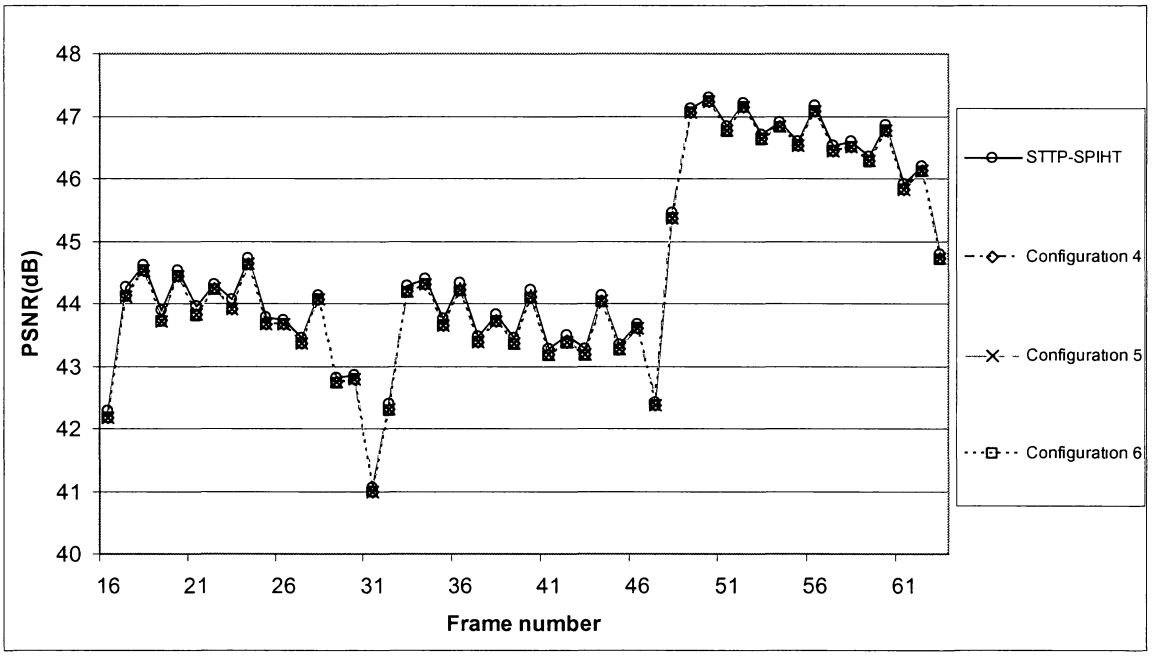
(a)



(b)

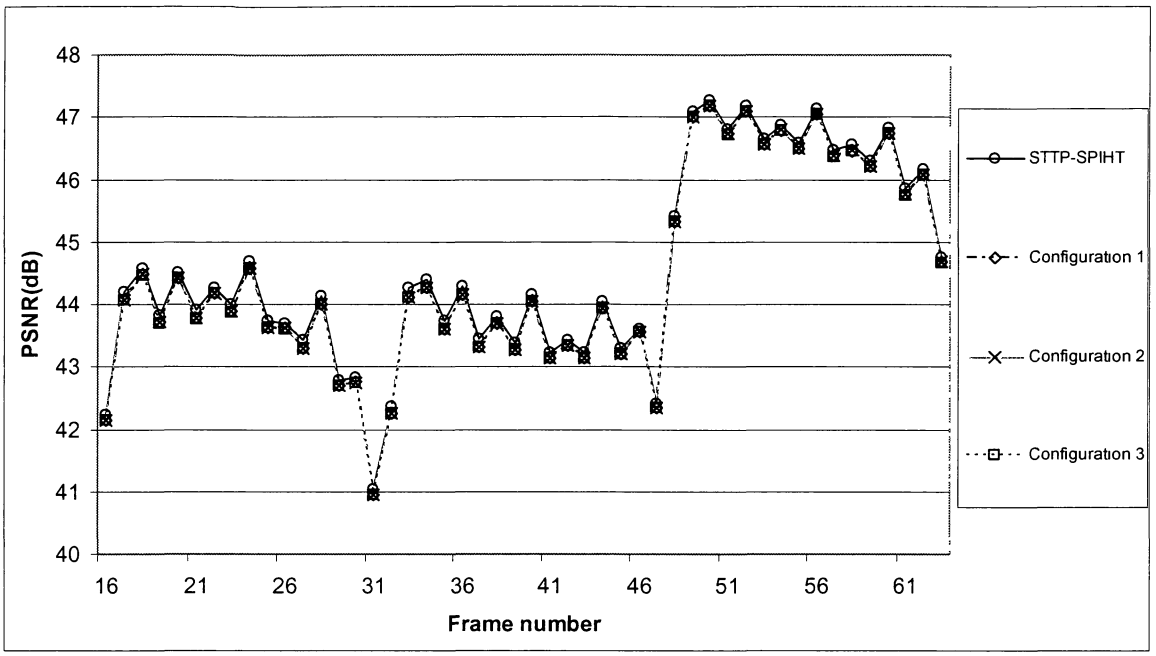
Figure 4.3: Frame by frame comparison of PSNR (dB) for "Football" video sequence in noiseless channels at a total transmission rate of 2.53 Mbps for $P = 10$. (a) Configurations 1 to 3. (b) Configurations 4 to 6.

(a)

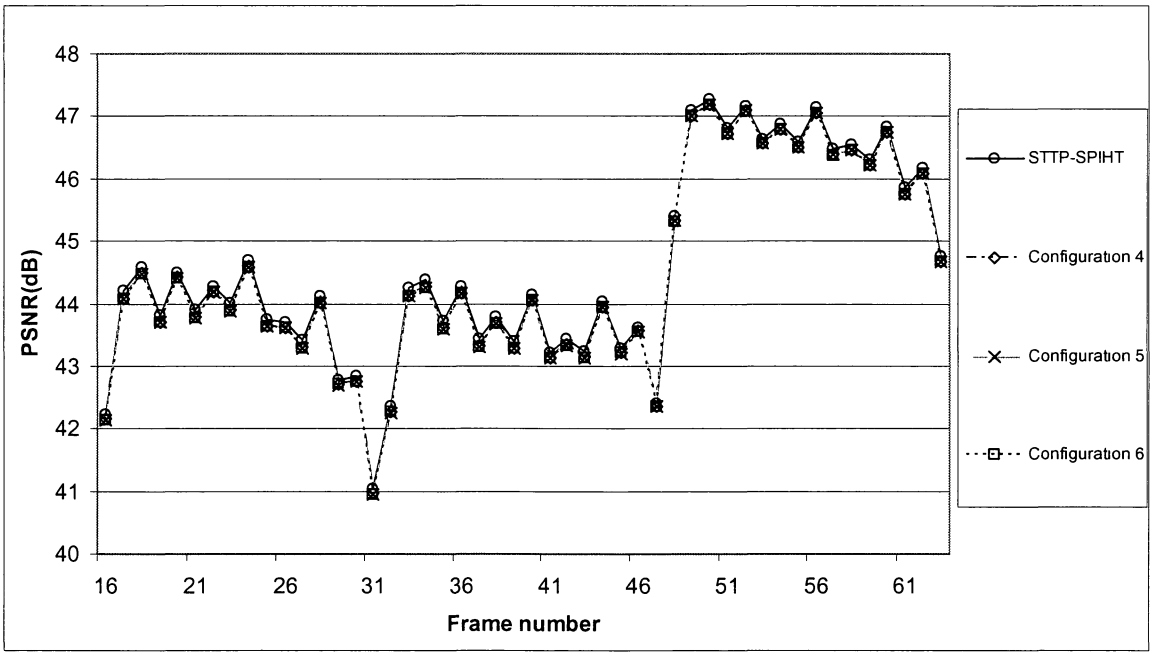


(b)

Figure 4.4: Frame by frame comparison of PSNR (dB) for "Susie" video sequence in noiseless channels at a total transmission rate of 2.53 Mbps for P = 4. (a) Configurations 1 to 3. (b) Configurations 4 to 6.



(a)



(b)

Figure 4.5: Frame by frame comparison of PSNR (dB) for "Susie" video sequence in noiseless channels at a total transmission rate of 2.53 Mbps for P = 10. (a) Configurations 1 to 3. (b) Configurations 4 to 6.

Table 4.1(a) and (b) demonstrate the average PSNR (dB) values of "Football" and "Susie" video sequences for the six configurations of the proposed MD video coding algorithm and the STTP-SPIHT algorithm, illustrated in Figure 4.2 through Figure 4.5. As we can see from the numerical results in Table 4.1, for both values of P ($P = 4$ and $P = 10$), the average PSNR values of the proposed algorithm are only slightly lower (less than 0.2 dB) than the STTP-SPIHT method. This very slight difference in the PSNR values is mainly caused by the additional redundancy that is inserted into the multiple substreams of the encoded original video sequence to protect those wavelet coefficients in the root subband. Also, we observe that the average PSNR values for the proposed algorithm using predictive coding with a sub-pixel correction shift is actually about 0.01 to 0.02 dB higher than that of the plain predictive coding. Furthermore, a larger value of P results in even slightly lower PSNR values, and this is caused by the extra overhead information bits. Hence, as the value of P increases, the PSNR values would be successively lower. In summary, with additional redundancy, the source coding efficiency of the proposed algorithm is actually reduced slightly (about 0.1 dB) in comparison to STTP-SPIHT. In the next section, we will compare the proposed MD video coding algorithm to STTP-SPIHT in noisy channels to study the error resilience performance.

Table 4.1: Comparison of average PSNR (dB) of "Football" and "Susie" video sequences for $P = 4$ and $P = 10$ in noiseless channels at a total transmission rate of 2.53 Mbps (best results shown in bold).

(a) Football

r (bpp)	Configuration 1 (dB)	Configuration 2 (dB)	Configuration 3 (dB)	STTP-SPIHT (dB)
2.53 Mbps	33.47 ($P = 4$) 33.34 ($P = 10$)	33.47 ($P = 4$) 33.34 ($P = 10$)	33.47 ($P = 4$) 33.34 ($P = 10$)	**33.61 ($P = 4$)** **33.46 ($P = 10$)**

r (bpp)	Configuration 4 (dB)	Configuration 5 (dB)	Configuration 6 (dB)	STTP-SPIHT (dB)
2.53 Mbps	33.49 ($P = 4$) 33.35 ($P = 10$)	33.49 ($P = 4$) 33.35 ($P = 10$)	33.48 ($P = 4$) 33.35 ($P = 10$)	**33.61 ($P = 4$)** **33.46 ($P = 10$)**

(b) Susie

r (bpp)	Configuration 1 (dB)	Configuration 2 (dB)	Configuration 3 (dB)	STTP-SPIHT (dB)
2.53 Mbps	44.50 ($P = 4$) 44.46 ($P = 10$)	44.50 ($P = 4$) 44.46 ($P = 10$)	44.50 ($P = 4$) 44.46 ($P = 10$)	**44.60 ($P = 4$)** **44.56 ($P = 10$)**

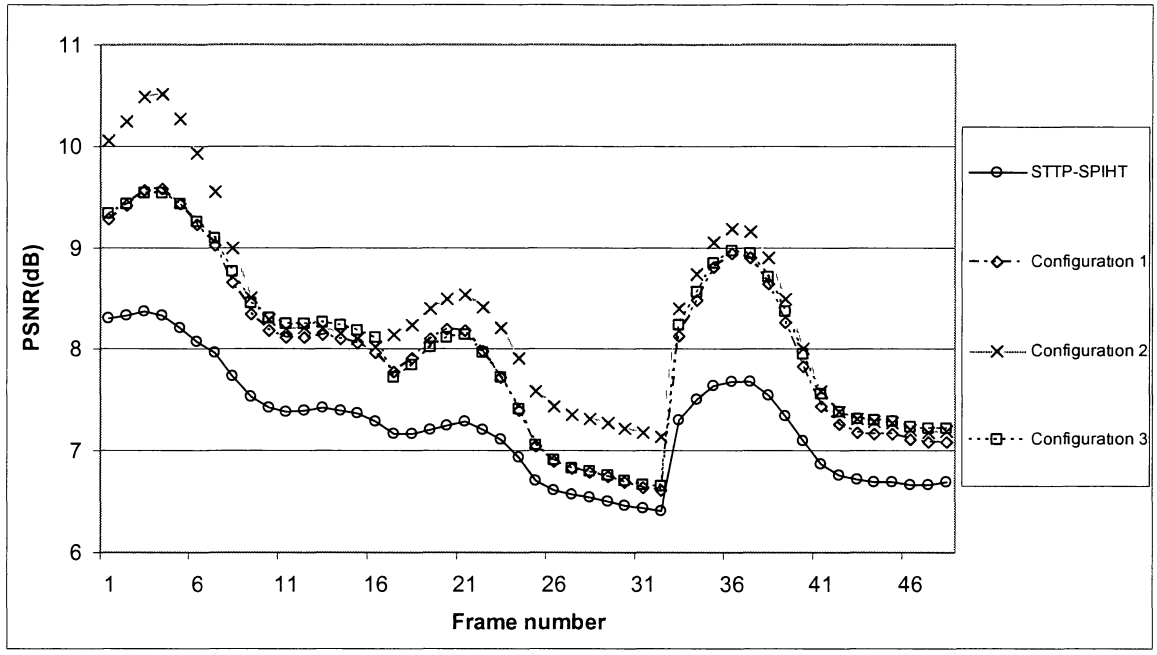
r (bpp)	Configuration 4 (dB)	Configuration 5 (dB)	Configuration 6 (dB)	STTP-SPIHT (dB)
2.53 Mbps	44.51 ($P = 4$) 44.47 ($P = 10$)	44.51 ($P = 4$) 44.47 ($P = 10$)	44.51 ($P = 4$) 44.47 ($P = 10$)	**44.60 ($P = 4$)** **44.56 ($P = 10$)**

4.4 Error Resilience Performance

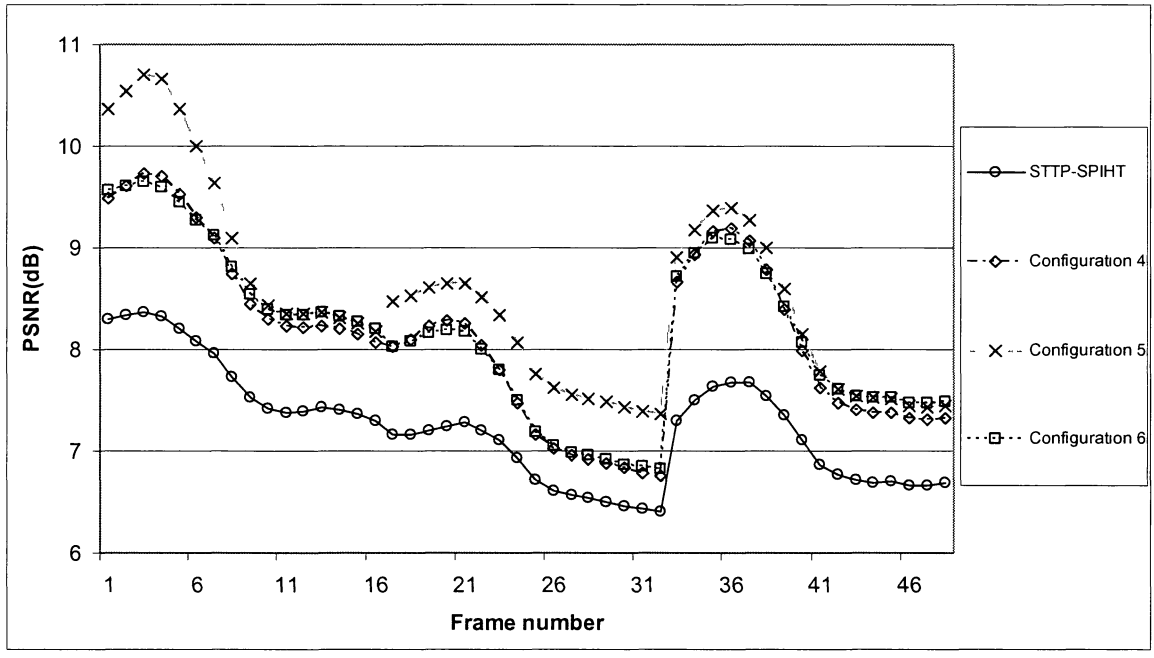
We now proceed to do a comparison of the proposed MD video coding algorithm to STTP-SPIHT in noisy channels so that we can study the error resilience performance, which is the goal of this thesis. We will only recover the missing wavelet coefficients in

the root subband, and all other missing wavelet coefficients in the higher frequency subbands will simply be set to zero. In our simulation of error-resilient video transmission, we will again perform the experiments using the six configurations with $P = 4$ and $P = 10$, and present a comparative analysis to STTP-SPIHT.

Figure 4.6 through Figure 4.11 illustrate the frame by frame comparison of PSNR values for the proposed video coding algorithm and STTP-SPIHT of the "Football" and "Susie" video sequences at a total transmission rate of 2.53 Mbps in noisy channels with bit error rates (BERs) of 10^{-3}, 10^{-4}, and 10^{-5} for four substreams ($P = 4$). Typically, wireless communication channels exhibit these BERs, as they are much more prone to bit errors than wired channels. These bit errors are generated randomly from the binomial distribution. It is clear from these figures that the PSNR values in the proposed algorithm case are much higher than those of STTP-SPIHT in the presence of channel bit errors, with configuration 5 offering the best results, and this is in accordance with what we expect in section 3.3.2. Moreover, the proposed algorithm using predictive coding with a sub-pixel correction shift (configurations 4, 5, and 6) in most cases has higher PSNR values than the proposed algorithm using plain predictive coding (configurations 1, 2, and 3). This is due to the fact that with a sub-pixel correction shift, fewer bits are required to encode the redundant information, and hence more bits can be used to encode the video data. Similar to the noiseless case, we can observe that the PSNR decreases at the GOP boundaries. The average PSNR values for both P values ($P = 4$ and $P = 10$) of the proposed algorithm and STTP-SPIHT are summarized in Table 4.2(a) and (b).

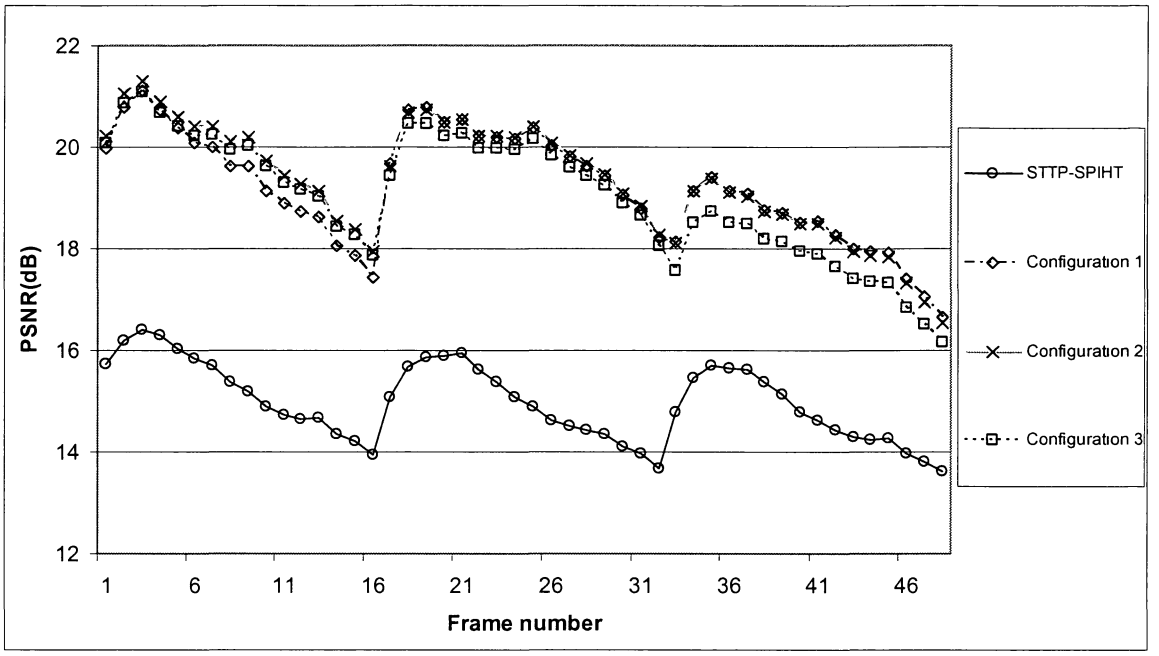


(a)

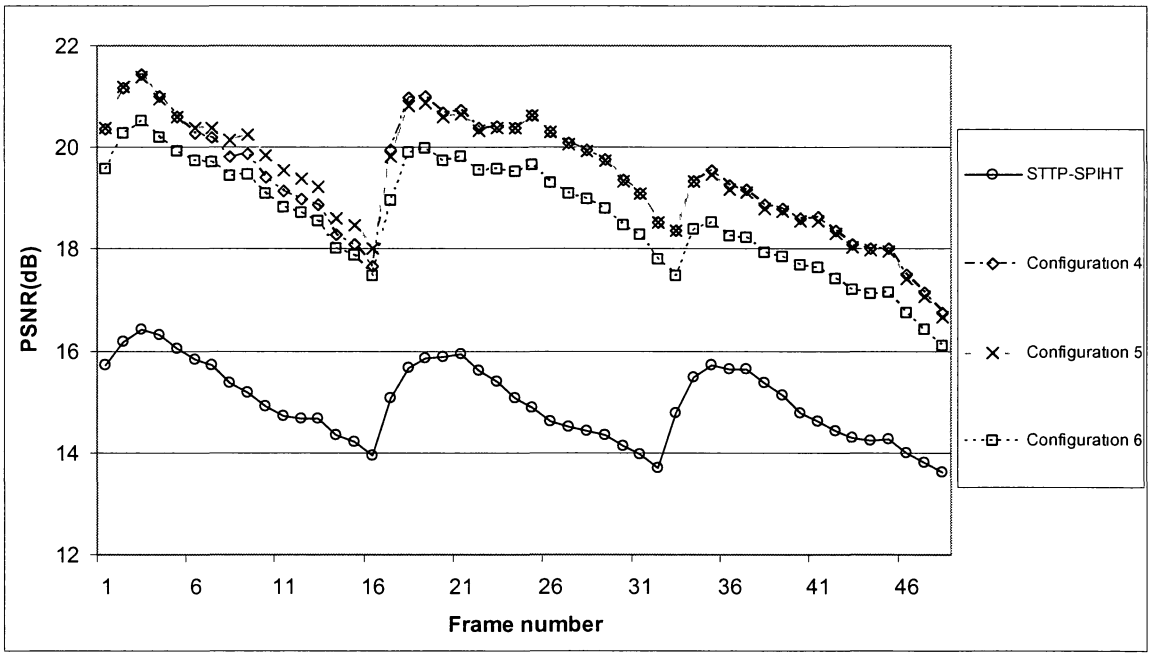


(b)

Figure 4.6: Frame by frame comparison of PSNR (dB) for "Football" video sequence with BER = 10^{-3} at a total transmission rate of 2.53 Mbps for P = 4. (a) Configurations 1 to 3. (b) Configurations 4 to 6.



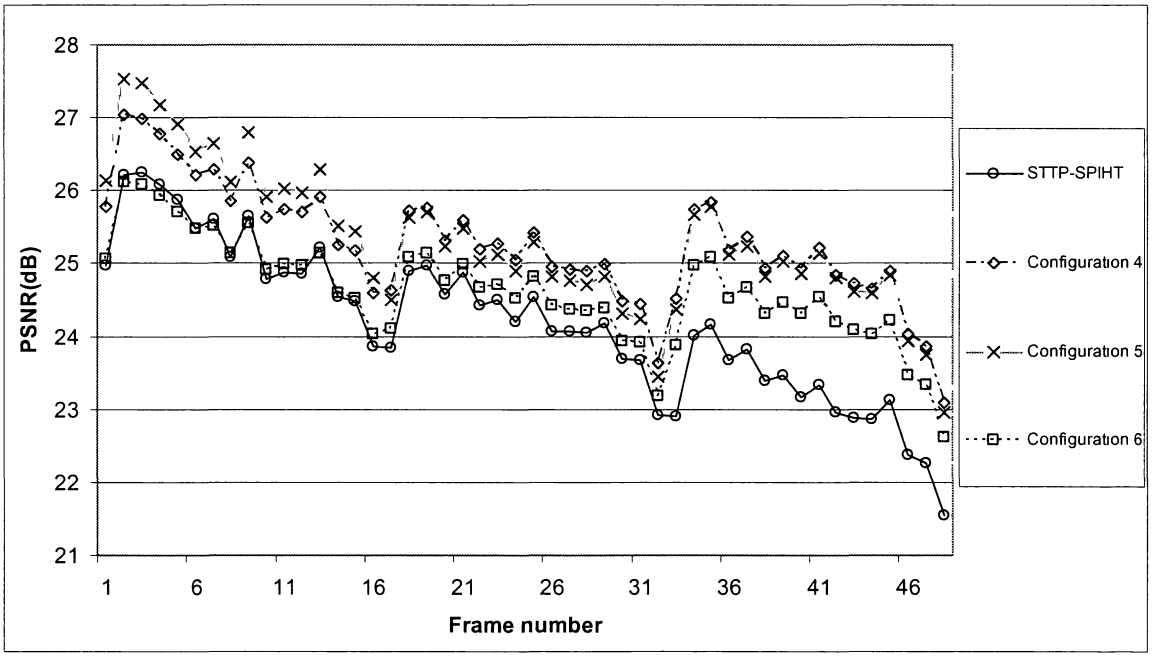
(a)



(b)

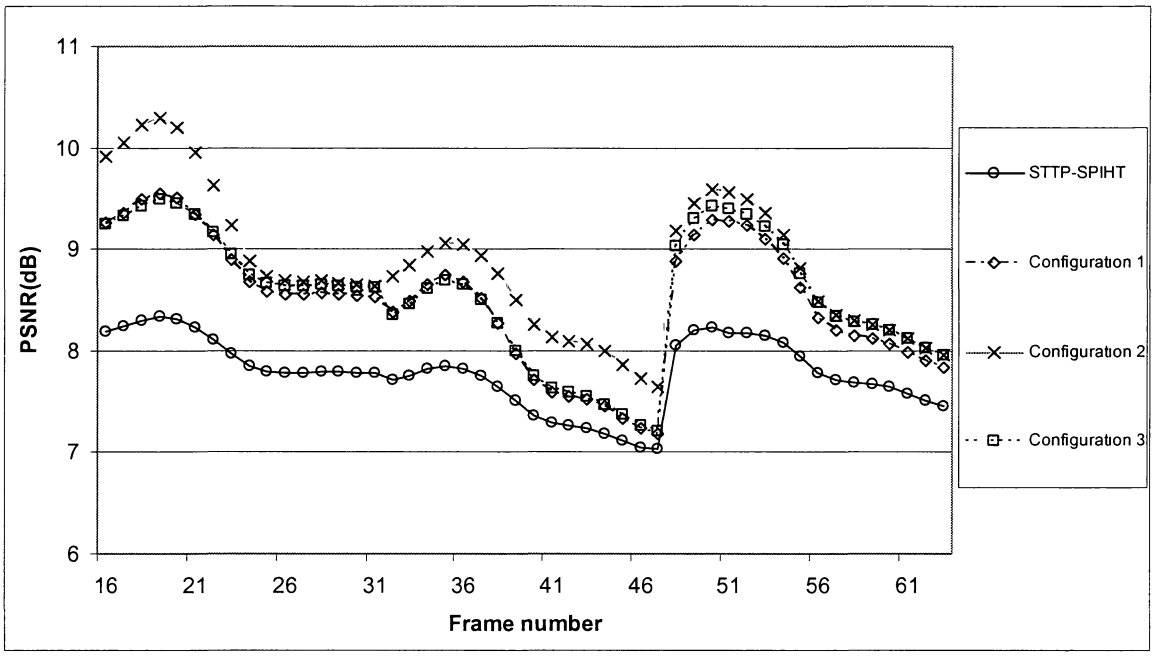
Figure 4.7: Frame by frame comparison of PSNR (dB) for "Football" video sequence with BER = 10^{-4} at a total transmission rate of 2.53 Mbps for P = 4. (a) Configurations 1 to 3. (b) Configurations 4 to 6.

(a)

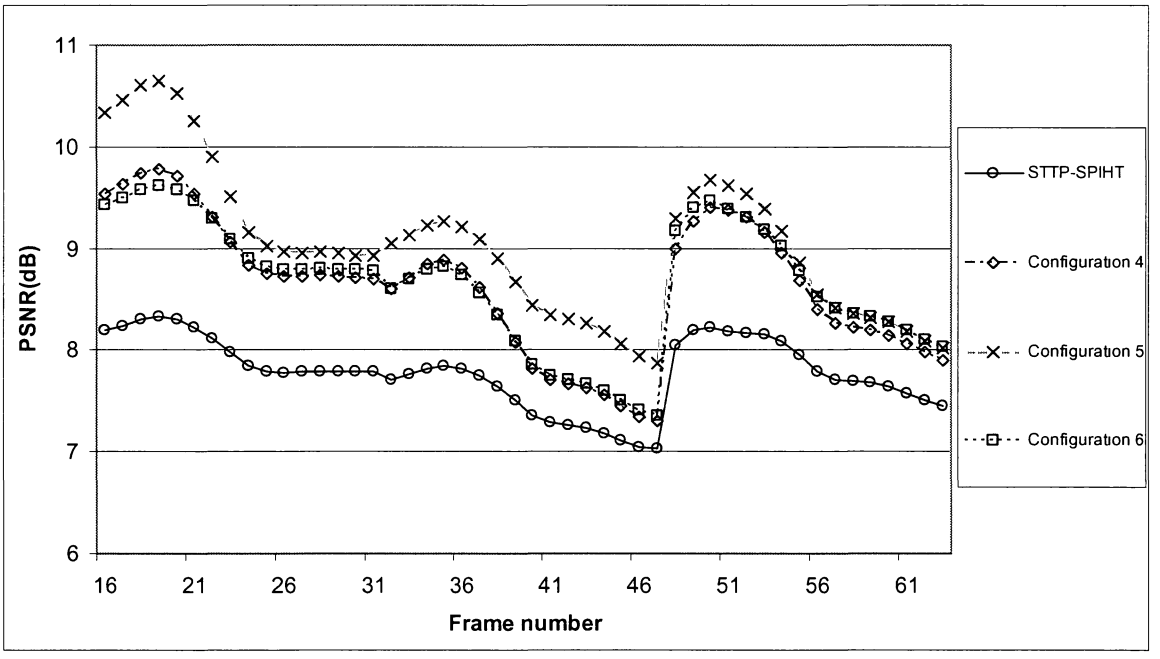


(b)

Figure 4.8: Frame by frame comparison of PSNR (dB) for "Football" video sequence with BER = 10^{-5} at a total transmission rate of 2.53 Mbps for P = 4. (a) Configurations 1 to 3. (b) Configurations 4 to 6.

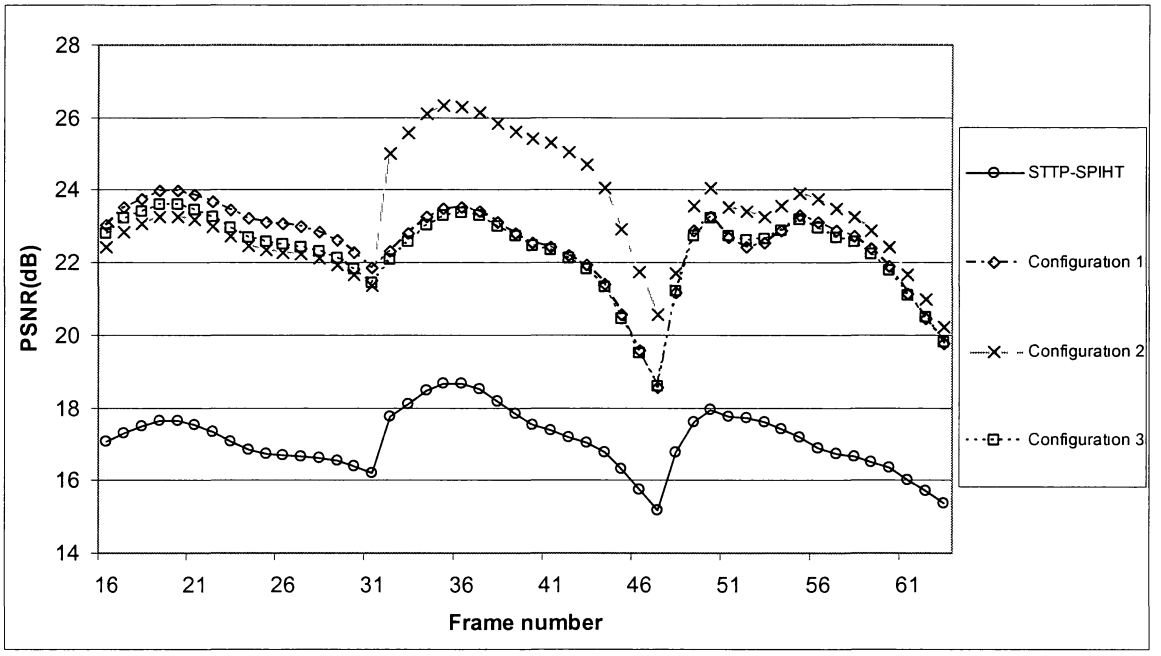


(a)

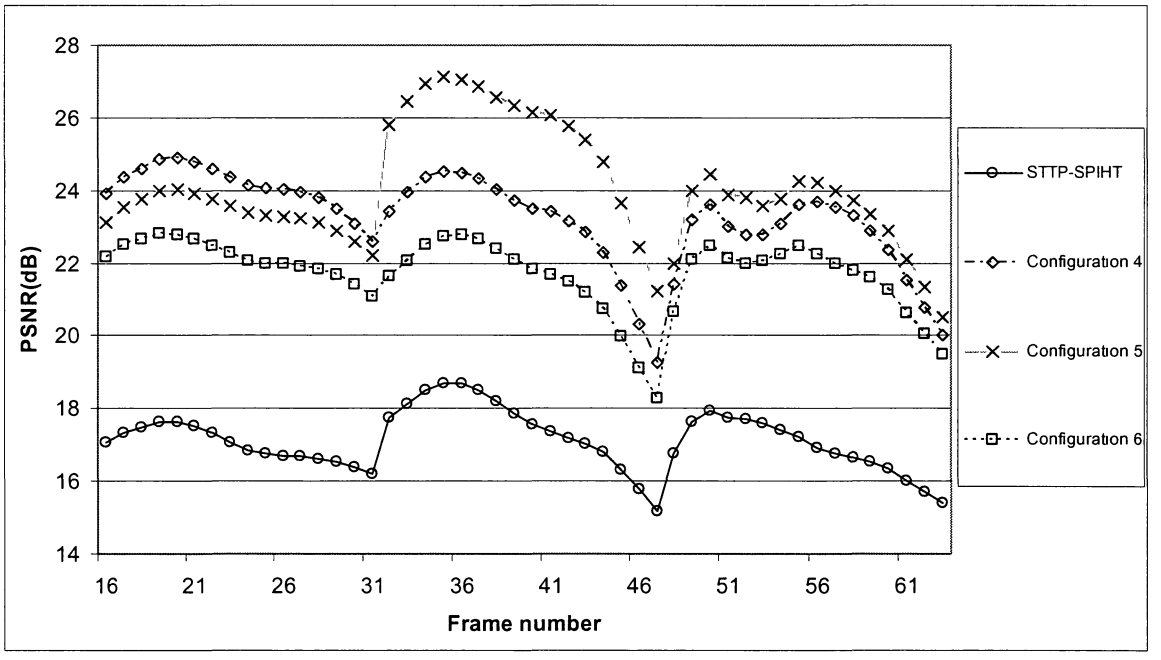


(b)

Figure 4.9: Frame by frame comparison of PSNR (dB) for "Susie" video sequence with BER = 10^{-3} at a total transmission rate of 2.53 Mbps for P = 4. (a) Configurations 1 to 3. (b) Configurations 4 to 6.

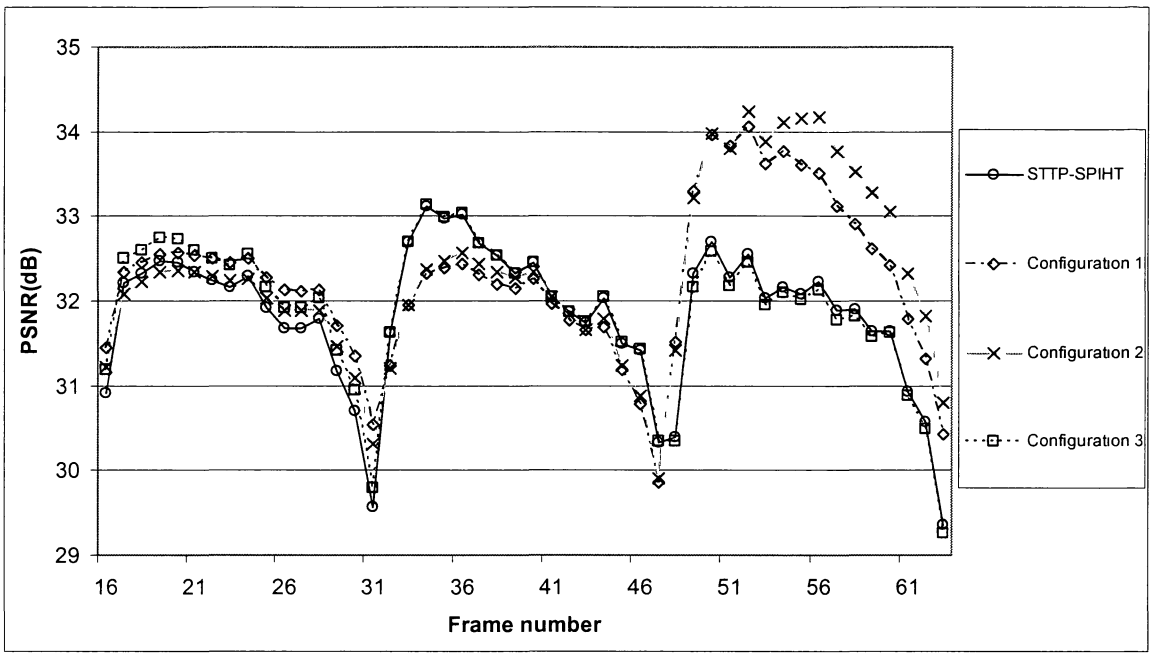


(a)

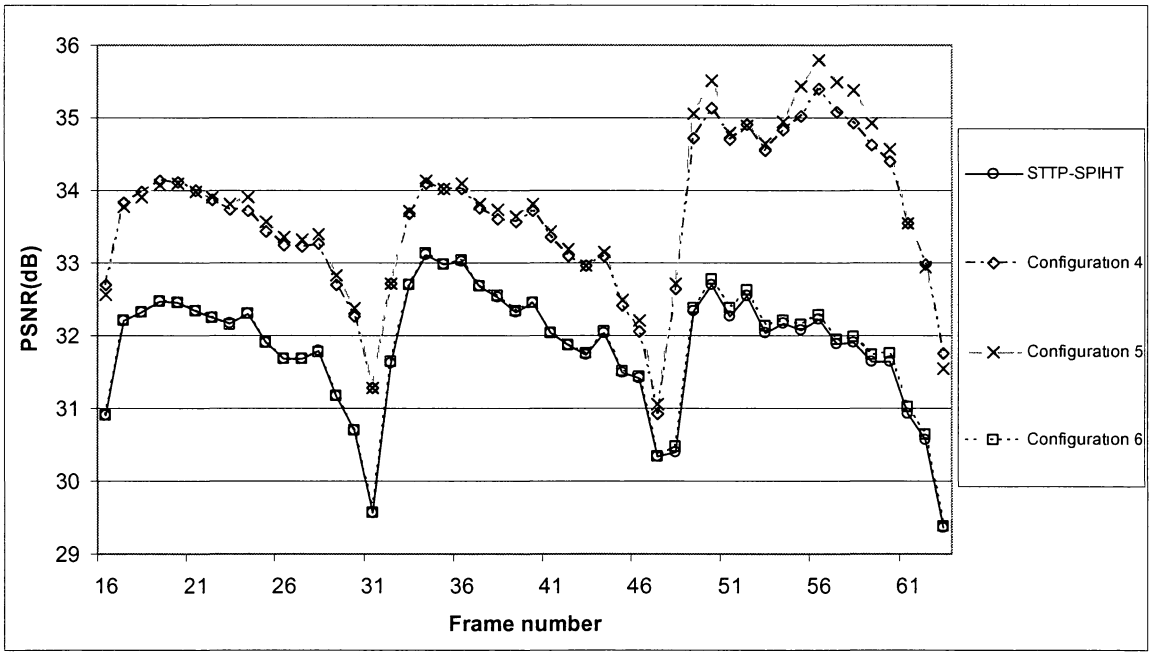


(b)

Figure 4.10: Frame by frame comparison of PSNR (dB) for "Susie" video sequence with BER = 10^{-4} at a total transmission rate of 2.53 Mbps for P = 4. (a) Configurations 1 to 3. (b) Configurations 4 to 6.



(a)



(b)

Figure 4.11: Frame by frame comparison of PSNR (dB) for "Susie" video sequence with BER = 10^{-5} at a total transmission rate of 2.53 Mbps for P = 4. (a) Configurations 1 to 3. (b) Configurations 4 to 6.

Table 4.2(a) and (b) illustrate the average PSNR (dB) values of "Football" and "Susie" video sequences for the six configurations of the proposed MD video coding algorithm and the STTP-SPIHT algorithm, at a total transmission rate of 2.53 Mbps for both four and ten substreams (P = 4 and P = 10). As we can see from Table 4.2, the average PSNR values for the proposed algorithm are much higher than those of STTP-SPIHT in the presence of channel bit errors, with configuration 5 offering the best PSNR results in each case. Configuration 5 outperforms all the other configurations mainly due to its efficient redundancy insertion structure and the sub-pixel correction shift. The improvement in the PSNR values is especially noticeable in the case for BER = 10^{-4}, and it is around 4 dB higher in the fast motion "Football" video sequence and around 7 dB higher in the slow motion "Susie" video sequence. The simulation results show that the improvement is more significant in the "Susie" video. Additionally, we also demonstrate that a larger P value results in higher PSNR values, and this is true for any BER. Therefore, as the value of P increases, the PSNR values would also improve. Furthermore, by observing the numerical results in Table 4.2, we note that the proposed algorithm using predictive coding with a sub-pixel correction shift in most cases outperforms the proposed algorithm using plain predictive coding.

Table 4.2: Comparison of average PSNR (dB) of "Football" and "Susie" video sequences for $P = 4$ and $P = 10$ with different BERs at a total transmission rate of 2.53 Mbps (best results shown in bold).

(a) Football

BER	10^{-5}	10^{-4}	10^{-3}
Configuration 1 (dB)	25.09 ($P = 4$) Std Dev: 0.99	19.20 ($P = 4$) Std Dev: 1.40	7.96 ($P = 4$) Std Dev: 0.86
	27.35 ($P = 10$) Std Dev: 0.86	21.41 ($P = 10$) Std Dev: 0.99	10.65 ($P = 10$) Std Dev: 1.13
Configuration 2 (dB)	25.23 ($P = 4$) Std Dev: 1.02	19.33 ($P = 4$) Std Dev: 1.55	8.31 ($P = 4$) Std Dev: 0.66
	27.70 ($P = 10$) Std Dev: 0.81	21.49 ($P = 10$) Std Dev: 0.96	11.07 ($P = 10$) Std Dev: 1.13
Configuration 3	25.07 ($P = 4$) Std Dev: 0.98	19.02 ($P = 4$) Std Dev: 1.69	8.02 ($P = 4$) Std Dev: 0.92
	27.22 ($P = 10$) Std Dev: 0.92	21.16 ($P = 10$) Std Dev: 0.96	10.71 ($P = 10$) Std Dev: 1.15
STTP-SPIHT (dB)	24.19 ($P = 4$) Std Dev: 2.21	14.98 ($P = 4$) Std Dev: 2.08	7.24 ($P = 4$) Std Dev: 0.46
	26.71 ($P = 10$) Std Dev: 1.64	17.91 ($P = 10$) Std Dev: 2.01	8.70 ($P = 10$) Std Dev: 0.55

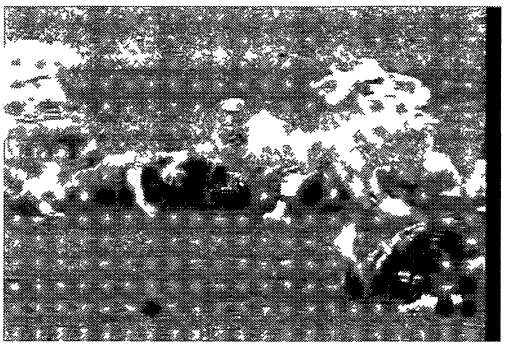
BER	10^{-5}	10^{-4}	10^{-3}
Configuration 4 (dB)	25.27 ($P = 4$) Std Dev: 0.90	19.41 ($P = 4$) Std Dev: 1.41	8.12 ($P = 4$) Std Dev: 0.90
	27.69 ($P = 10$) Std Dev: 0.82	21.44 ($P = 10$) Std Dev: 0.95	10.75 ($P = 10$) Std Dev: 1.09
Configuration 5 (dB)	**25.31 ($P = 4$)** Std Dev: 1.02	**19.45 ($P = 4$)** Std Dev: 1.51	**8.51 ($P = 4$)** Std Dev: 0.70
	27.76 ($P = 10$) Std Dev: 0.79	**21.55 ($P = 10$)** Std Dev: 0.94	**11.23 ($P = 10$)** Std Dev: 1.13
Configuration 6	24.62 ($P = 4$) Std Dev: 0.93	18.64 ($P = 4$) Std Dev: 1.63	8.16 ($P = 4$) Std Dev: 0.94
	26.78 ($P = 10$) Std Dev: 0.83	20.79 ($P = 10$) Std Dev: 0.94	10.93 ($P = 10$) Std Dev: 1.19
STTP-SPIHT (dB)	24.19 ($P = 4$) Std Dev: 2.21	14.98 ($P = 4$) Std Dev: 2.08	7.24 ($P = 4$) Std Dev: 0.46
	26.71 ($P = 10$) Std Dev: 1.64	17.91 ($P = 10$) Std Dev: 2.01	8.70 ($P = 10$) Std Dev: 0.55

(b) Susie

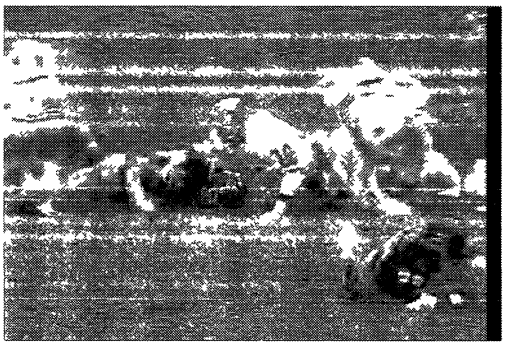
BER	10^{-5}	10^{-4}	10^{-3}
Configuration 1 (dB)	32.24 ($P = 4$) Std Dev: 1.43	22.51 ($P = 4$) Std Dev: 2.45	8.49 ($P = 4$) Std Dev: 0.84
	37.00 ($P = 10$) Std Dev: 1.38	27.18 ($P = 10$) Std Dev: 1.93	11.70 ($P = 10$) Std Dev: 1.33
Configuration 2 (dB)	32.30 ($P = 4$) Std Dev: 1.36	23.38 ($P = 4$) Std Dev: 2.25	8.83 ($P = 4$) Std Dev: 0.66
	37.20 ($P = 10$) Std Dev: 1.31	27.44 ($P = 10$) Std Dev: 1.65	12.13 ($P = 10$) Std Dev: 1.46
Configuration 3 (dB)	31.91 ($P = 4$) Std Dev: 1.36	22.31 ($P = 4$) Std Dev: 2.78	8.55 ($P = 4$) Std Dev: 0.90
	36.90 ($P = 10$) Std Dev: 1.30	26.71 ($P = 10$) Std Dev: 1.91	11.78 ($P = 10$) Std Dev: 1.29
STTP-SPIHT (dB)	31.85 ($P = 4$) Std Dev: 2.39	17.11 ($P = 4$) Std Dev: 2.82	7.78 ($P = 4$) Std Dev: 0.46
	35.12 ($P = 10$) Std Dev: 2.56	20.77 ($P = 10$) Std Dev: 2.14	9.55 ($P = 10$) Std Dev: 0.70

BER	10^{-5}	10^{-4}	10^{-3}
Configuration 4 (dB)	33.62 ($P = 4$) Std Dev: 1.71	23.25 ($P = 4$) Std Dev: 2.35	8.62 ($P = 4$) Std Dev: 0.93
	37.07 ($P = 10$) Std Dev: 1.29	27.23 ($P = 10$) Std Dev: 1.88	11.87 ($P = 10$) Std Dev: 1.36
Configuration 5 (dB)	**33.72 ($P = 4$)** Std Dev: 1.70	**24.04 ($P = 4$)** Std Dev: 2.29	**9.03 ($P = 4$)** Std Dev: 0.72
	37.35 ($P = 10$) Std Dev: 1.26	**27.73 ($P = 10$)** Std Dev: 1.59	**12.27 ($P = 10$)** Std Dev: 1.46
Configuration 6 (dB)	31.89 ($P = 4$) Std Dev: 1.39	21.72 ($P = 4$) Std Dev: 2.68	8.65 ($P = 4$) Std Dev: 0.96
	36.85 ($P = 10$) Std Dev: 1.59	25.89 ($P = 10$) Std Dev: 1.87	12.05 ($P = 10$) Std Dev: 1.38
STTP-SPIHT (dB)	31.85 ($P = 4$) Std Dev: 2.39	17.11 ($P = 4$) Std Dev: 2.82	7.78 ($P = 4$) Std Dev: 0.46
	35.12 ($P = 10$) Std Dev: 2.56	20.77 ($P = 10$) Std Dev: 2.14	9.55 ($P = 10$) Std Dev: 0.70

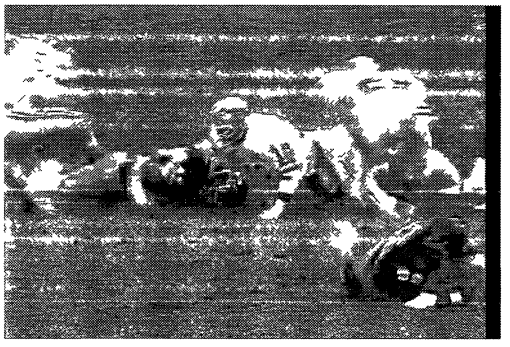
To demonstrate the visual quality of the proposed method and STTP-SPIHT, the reconstructed frames (frame 9) in "Football" video sequence are illustrated in Figure 4.12(a), (b), and (c). Figure 4.12(a) illustrates the reconstructed frame at a total transmission rate of 2.53 Mbps with channel BER of 10^{-5} using STTP-SPIHT (P = 4), and PSNR = 21.56 dB. The reconstructed frame using configuration 5 (P = 4) of our proposed algorithm is shown in Figure 4.12(b), and PSNR = 26.01 dB. Furthermore, Figure 4.12(c) illustrates the reconstructed frame using configuration 5 (P = 10), and PSNR = 27.06 dB. We can observe that the perceptual quality with the proposed algorithm is noticeably better than that of STTP-SPIHT. Additionally, the visual results improve as the number of s-t blocks P increase.



(a)



(b)



(c)

Figure 4.12: 352 x 240 "Football" sequence (frame 9) at 2.53 Mbps with BER = 10^{-5}. (a) Reconstructed frame using STTP-SPIHT (P = 4), PSNR = 21.56 dB. (b) Reconstructed frame using configuration 5 (P = 4) of proposed algorithm, PSNR = 26.01 dB. (c) Reconstructed frame using configuration 5 (P = 10) of proposed algorithm, PSNR = 27.06 dB.

4.4 Summary

In this chapter, we described the various experiments that were performed on the monochrome video sequences in the noiseless and noisy channels in order to determine the effectiveness of our proposed MD video coding approach. The comparative analysis

is performed with respect to source coding efficiency and error resilience performance. From our simulation results in the noiseless channels, we found that the source coding efficiency of the proposed algorithm is actually reduced by a small amount in comparison to STTP-SPIHT, but the reduction is very small and likely not perceivable. In the presence of bit errors, we can see that our proposed algorithm outperforms the STTP-SPIHT method, and has proved to be much more robust to random channel bit errors. Furthermore, we have also shown that the proposed algorithm using predictive coding with a sub-pixel correction shift outperforms the plain predictive coding.

Chapter 5: Conclusions and Future Work

5.1 Conclusions

In this thesis, we presented an error-resilient domain-partioning based MD video coding algorithm for robust video transmission in error-prone packet switched networks. We improved upon the STTP-SPIHT algorithm by intentionally inserting a certain amount of redundant data into the multiple substreams of the encoded original video sequence in order to protect those wavelet transform coefficients in the approximation subband. At the receiver side, we can use this redundant information along with the other correctly received substreams to predict the missing wavelet coefficients in the lost substream. Specifically, the insertion of redundancy can be achieved with six configurations: three of them use predictive coding and the other three use predictive coding with a sub-pixel correction shift.

We then demonstrate the effectiveness of our proposed MD video coding algorithm by performing extensive comparative tests with STTP-SPIHT on monochrome video sequences in noiseless and noisy channels, with respect to source coding efficiency and error resilience performance. Even though our proposed MD coding algorithm has a small loss in the noiseless channel performance in comparison to the STTP-SPIHT algorithm due to the additional redundant information, it has been shown to be more resilient to random channel bit errors than STTP-SPIHT with little increase in its

complexity. The simulation results also indicate that configuration 5 of our proposed algorithm outperforms all the other configurations due to its efficient redundancy insertion structure and the sub-pixel correction shift.

5.2 Future Work

As for future research, we would like to further investigate the following items:

- We would like to explore the possibility of inserting redundancy not only from the wavelet transform coefficients in the approximation subband, but also from the higher frequency subbands.
- To further improve the reconstructed video quality, we would like to consider integrating error concealment at the decoder side after most of the missing low frequency coefficients have been recovered. The error concealment techniques can be applied to wavelet coefficient in either low frequency or high frequency subbands.

References

[1] ITU-T Rec. H.262, "Information technology – Generic coding of moving pictures and associated audio information – Part 2: Video," ISO/IEC 13818-2 (MPEG-2), Mar. 1995.

[2] ISO/IEC, "Information technology – Coding of audio-visual objects – Part 2: Visual," ISO/IEC 14496-2 (MPEG-4), Oct. 1998.

[3] ITU-T Rec. H.263, "Video Coding for low bit rate communication," Mar. 1996.

[4] ITU-T Rec. H.264, "Advanced video coding for generic audiovisual services," May 2003.

[5] W. Y. Kung, C. S. Kim, and C. C. J. Kuo, "Spatial and temporal error concealment techniques for video transmission over noisy channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 7, pp. 789–802, Jul. 2006.

[6] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74-93, Sep. 2001.

[7] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, "Multiple description video coding for scalable and robust transmission over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 321-334, Mar. 2005.

[8] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," in *Proc. IEEE*, vol. 93, no. 1, pp. 57-70, Jan. 2005.

[9] M. Khansari, A. Zakauddin, W.-Y. Chan, E. Dubois, and P. Mermelstein, "Approaches to layered coding for dual-rate wireless video transmission," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, pp. 258-262, Nov. 1994.

[10] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Select. Areas Commun.*, vol. 7, no. 5, pp. 771-781, Jun. 1989.

[11] Y.-C. Lee, J. Kim, Y. Altunbasak, and R. M. Mersereau, "Layered coded vs. multiple description coded video over error-prone networks," *EURASIP Signal Processing-Image Communication*, vol. 18, pp. 337-356, May 2003.

[12] R. Singh, A. Ortega, L. Perret, and W. Jiang, "Comparison of multiple description coding and layered coding based on network simulations," in *Proc. SPIE Vis. Commun. Image Process.*, vol. 3974, pp. 929-939, Jan. 2000.

[13] J. Chakareski, S. Han, B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," *Springer Multimedia Syst.*, vol. 10, no. 4, pp. 275-285, Jan. 2005.

[14] C. D. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Trans. Image Process.*, vol. 6, no. 10, pp. 1436-1442, Oct. 1997.

[15] C. D. Creusere, "A family of image compression algorithms which are robust to transmission errors," *Proc. SPIE*, vol. 2825, pp. 890-900, Oct. 1996.

[16] A. A. Alatan, M. Zhao, and A. N. Akansu, "Unequal error protection of SPIHT encoded image bit streams," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 814-818, Jun. 2000.

[17] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389-400, Apr. 1988.

[18] S. Cho and W. A. Pearlman, "Error resilient compression and transmission of scalable video," *Proc. SPIE*, vol. 4115, pp. 396-405, Dec. 2000.

[19] S. Cho and W. A. Pearlman, "A full-featured, error-resilient, scalable wavelet video codec based on the set partitioning in hierarchical trees (SPIHT) algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 3, pp. 157-171, Mar. 2002.

[20] P.G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Process. Lett.*, vol. 4, no. 7, pp. 189-191, Jul. 1997.

[21] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674-693, Jul. 1989.

[22] D. S. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, 1st ed. Boston, USA: Kluwer Academic Press, 2001.

[23] S. Mallat, *A Wavelet Tour of Signal Processing*, 1st ed. San Diego, CA: Academic Press, 1998.

[24] I. Daubechies, *Ten Lectures on Wavelets*, 1st ed. Society For Industrial and Applied Mathematics (SIAM), May 1992.

[25] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal. 3*, pp. 186-200, Apr. 1996.

[26] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," *Proc. SPIE*, vol. 2569, pp. 68-79, Jul. 1995.

[27] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511-546, Mar. 1998.

[28] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 245-267, May 1998.

[29] C. He, J. Dong, Y. F. Zheng, and Z. Gao, "Optimal 3-D coefficient tree structure for 3-D wavelet video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 10, pp. 961-972, Oct. 2003.

[30] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[31] A. Said and W. A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243-250, Jun. 1996.

[32] M. Sakalli, W. A. Pearlman, and M. Farshchian, "SPIHT algorithms using depth first search algorithm with minimum memory usage," in *Proc. Conf. Inform. Sci. and Syst.*, pp. 1158-1163, Mar. 2006.

[33] Y. Chen and W. A. Pearlman, "Three-dimensional subband coding of video using the zero-tree method," *SPIE Vis. Commun. Image Process.*, vol. 2727, pp. 1302-1312, Mar. 1996.

[34] B. J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," in *Proc. Data Compression Conf.*, pp. 251-260, Mar. 1997.

[35] W. A. Pearlman, B. J. Kim, and Z. Xiong, "Embedded video subband coding with 3D SPIHT," *Springer Netherlands*, vol. 450, pp. 397-432, 2002.

[36] B. J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D Set partitioning in Hierarchical Trees (3-D SPIHT)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 1374-1387, Dec. 2000.

[37] P. Elias, "Predictive coding—Part I," *IRE Trans. Inf. Theory*, vol. 1, no. 1, pp. 16-24, Mar. 1955.

[38] P. Elias, "Predictive coding—Part II," *IRE Trans. Inf. Theory*, vol. 1, no.1, pp. 24-33, Mar. 1955.

[39] J. Xu, F. Wu, and W. Zhang, "Intra-predictive transforms for block-based image coding," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3030-3040, Aug. 2009.

[40] G. Motta, J. A. Storer, and B. Carpentieri, "Lossless image coding via adaptive linear prediction and classification," *Proc. IEEE*, vol. 88, no. 11, pp. 1790-1796, Nov. 2000.

[41] G. R. Kuduvalli, and R. M. Rangayyan, "Performance analysis of reversible image compression techniques for high-resolution digital teleradiology," *IEEE Trans. Med. Imag.*, vol. 11, no. 3, pp. 430-445, Sep. 1992.

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

[43] D. Wang, L. Zhang, and A. Vincent, "New method for reducing GOP-boundary artifacts in wavelet-based video coding," *IEEE Trans. Broadcast.*, vol. 52, no. 3, pp. 350-355, Sep. 2006.

[44] J. Liang, C. Tu, and T. D. Tran, "Optimal block boundary pre/postfiltering for wavelet-based image and video compression," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2151-2158, Dec. 2005.

[45] I. Kharitonenko, X. Zhang, and S. Twelves, "A wavelet transform with point-symmetric extension at tile boundaries," *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1357–1364, Dec. 2002.