Unequal Erasure Protection Techniques for Scalable Multi-Streams

UNEQUAL ERASURE PROTECTION TECHNIQUES FOR SCALABLE MULTI-STREAMS

BY

GEOFFREY KEITH RIVERS, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2009) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Unequal Erasure Protection Techniques for Scalable Multi-Streams
AUTHOR:	Geoffrey Keith Rivers B.Sc., (Computer Engineering) University of Waterloo, Waterloo, Ontario Canada
SUPERVISOR:	Dr. Sorina Dumitrescu Dr. Shahram Shirani
NUMBER OF PAGES:	xvi, 98

Dedications

With much love, I dedicate this to my wife Kourtney, and my parent's Sue and Keith. Their love, support and sacrifices have provided me with the opportunity to pursue a graduate studies degree, and I am forever thankful.

Abstract

This thesis presents a novel unequal erasure protection (UEP) strategy for the transmission of scalable data, formed by interleaving independently decodable and scalable streams, over packet erasure networks. The technique, termed multi-stream unequal erasure protection (M-UEP) differs from UEP by placing separate streams in separate packets to establish independence and using permuted systematic Reed-Solomon codes to enhance the distribution of message symbols amongst the packets. M-UEP improves upon UEP by ensuring that all received source symbols are decoded. The R-D optimal redundancy allocation problem for M-UEP is formulated and its globally optimal solution is shown to have a time complexity of $O(2^N N(L+1)^{N+1})$, where N is the number of packets and L is the packet length. To address the high complexity of the globally optimal solution, an efficient sub-optimal algorithm which runs in $O(N^2L^2)$ time is formulated. The additional side information necessary for M-UEP at the decoder is discussed and an upper bound on the amount of side information is derived. To mitigate the necessary side information, a technique termed (FM-UEP) is presented. A constrained optimal algorithm for generating N substreams from P primary substreams ($P > N$) is formulated, where the placement of primary substreams into N groups is constrained to a fixed-order. Four possible fixed-orders are proposed: raster scan, zig-zag scan, dispersed dot dithered and subband dispersed. Experiments

performed on SPIHT coded images (with appropriate grouping of wavelet coefficient) validate the superiority of M-UEP and FM-UEP over UEP, with peak improvements of 0.6 and 0.5 dB, respectively. Additionally, our tests reveal that M-UEP is more robust than UEP in adverse, unpredictable and varying channel conditions.

Acknowledgements

There were many individuals whose essential support led me through this journey of research over the last two years culminating in this thesis work.

Firstly, I would especially like to thank my supervisors Dr. Sorina Dumitrescu and Dr. Shahram Shirani. Their support and input was always available, and always much appreciated. Without their efforts, this most certainly would not have come into fruition. I am very thankful for their time and dedication throughout my two years at McMaster. I know they will continue to inspire and guide students with the same care and devotion they have shown me for many years to come!

Also, I would like acknowledge my other thesis committee members Dr. Sirouspour (committee chair) and Dr. Zhang. Thank you for the time dedicated to participating in my thesis defense, and providing valuable input to my work. I would also like to acknowledge all the great staff at McMaster who make it easy to focus on studies knowing that we have such a great support structure.

Lastly I would like to thank God, who humbles me each and every day by his grace. Thanks for keeping me safe during my long commutes to McMaster, for keeping me in good health, and blessing me with a wonderful wife and family. Without your grace, none of this would be possible.

List of Abbreviations

CRC : Cyclic Redundancy Check Code

dB : Decibel

DD : Dispersed Dot Dithered Scan

EBCOT : Embedded Block Coding with Optimal Truncation

FEC : Forward Error Correction

FIFO : First-in, First-out

FM-UEP : Multi-stream Erasure Protection with Fixed Redundancy Locations

IP : Internet Protocol

IPL : Independent Packet Loss

LDPC : Low Density Parity Check Code

LIP : List of Insignificant Pixels

LIS : List of Insignificant Sets

LSP : List of Significant Pixels

MSE : Mean Squared Error

M-UEP : Multi-stream Erasure Protection

NRN : Non Root Nodes

PCRD : Post Compression Rate-Distortion

PSNR : Peak Signal-to-Noise Ratio

QoS : Quality of Service

RCPC : Rate Controlled Punctured Convolution Code

RD-ORA : Rate-Distortion Optimal Redundancy Allocation

RS : Reed-Solomon Code Raster Scan

SD : Subband Dispersed

SNR : Signal-to-Noise Ratio

SPIHT : Set Partitioning in Hierarchical Trees

UEP : Unequal Erasure Protection

ZZ : Zig-zag Scan

Contents

Abstract iv

Acknowledgements vi

List of Abbreviations vii

1 Introduction 1

1.1 Transmission over IP Networks 1

1.2 Embedded Bit-streams 3

1.3 Unequal Error Protection Strategies for Embedded Bit-streams 3

1.4 Error Protection Strategies for Multi-streams 5

1.5 Contribution and Organization of Thesis 6

2 Embedded Multi-streams 10

2.1 Wavelets 11

2.1.1 Wavelet Decomposition of Images 12

2.2 Wavelet Based Image Encoders 15

2.2.1 Embedded Zerotree Wavelet (EZW) Algorithm 16

2.2.2 Set Partitioning in Hierarchical Trees (SPIHT) 18

2.2.3 Embedded Block Coding with Optimal Truncation (EBCOT) 20

3 Unequal Erasure Protection **22**

3.1 Formal Description of UEP 22

3.2 UEP Rate-Distortion Optimal Redundancy Allocation 25

3.3 UEP Side Information 27

4 Multi-stream Unequal Erasure Protection **31**

4.1 Formal Definition of M-UEP 32

4.1.1 M-UEP RD-ORA 35

4.2 Graph-Based Globally M-UEP RD-ORA Solution 37

4.2.1 Weighted Directed Acyclic Graph Formulation 37

4.2.2 Correspondence between G and M-UEP Packetization Array . 38

4.2.3 Unconstrained Maximum-Weight Path Problem Formulation . 40

4.3 Sub-optimal Solution to M-UEP RD-ORA 42

4.4 M-UEP Side Information 46

4.5 M-UEP with Fixed Redundancy Locations 49

5 Primary Sub-stream Grouping Strategies **52**

5.1 Unconstrained Optimal Primary Sub-stream Grouping 53

5.2 Constrained Optimal Primary Sub-stream Grouping 54

5.2.1 Algorithm Formulation 55

5.2.2 Primary Substream Fixed-Order Assignment 56

5.3 Fixed Size Primary Sub-stream Grouping 64

5.4 Grouping Overhead 67

6 Experimental Results **69**

6.1 Grouping Performance Evaluation 70

6.2 M-UEP Overhead 78

6.3 M-UEP under channel mismatch 81

7 Conclusions **84**

A Sufficiency of M-UEP RD-ORA Constraints (4.3), (4.4), (4.5) **86**

B Unconstrained Maximum-Weigth Path Problem Formulation for Convex R-D Curves **90**

List of Figures

2.1 Single level 2-D wavelet decomposition of an $M \times M$ input image. The decomposition is performed by a two-stage filter bank with the first stage performing 1-D filtering of the rows of the image, and the second stage performing the 1-D filtering of the columns. 13

2.2 Illustration of a 3-level wavelet decomposition of an input image. LR = low resolution approximation coefficients, HR = horizontal refinement coefficients, VR = vertical refinement coefficients, DR = diagonal refinement coefficients. 15

2.3 Illustration of an EZW tree in the wavelet coefficient matrix of a 3-level wavelet decomposition 17

2.4 SPIHT algorithm spatial orientation tree relationships of a 2×2 block in the lowest resolution approximation region $LR3$ of the wavelet coefficient matrix. The wavelet matrix corresponds to a 3-level wavelet decomposition 19

3.1 UEP packetization array for $N = 4$, $L = 8$. White boxes represent source symbols, grey boxes represent redundancy symbols. 24

4.1 M-UEP packetization array for $N = 4$, $L = 8$. White boxes represent lost source symbols, light grey boxes represent received source symbols and grey boxes represent redundancy symbols. 34

4.2 Example of the one-to-one correspondence between a packetization array $PA_{4\times8}$ and a path in $G_{4\times8}$. Gray boxes represent redundancy symbols and white boxes represent source symbols. The source symbols from each substream are assigned to a single packet ($A = P_1, B = P_2, C = P_3, D = P_4$). 39

4.3 M-UEP packetization array layers j and $j+1$ prior to swapping source symbols (a), and after swapping source symbols (b). Gray boxes represent redundancy symbols and white boxes represent source symbols. 44

4.4 M-UEP packetization array source symbol assignment pseudocode. . . 47

4.5 FM-UEP packetization array source symbol assignment pseudocode. . 50

4.6 FM-UEP packetization array for $N = 5$, $L = 10$ where $x_j = \{2, 3, 0, 4, 1\}$. White boxes represent source symbols and grey boxes represent redundancy symbols. 51

5.1 Pseudocode for optimal grouping of P primary substreams with fixed-order assignment to N packets. Solving for $\Delta D(N, P)$ 57

5.2 Lowest resolution subband (LL_0) of a wavelet decomposed image. LL_0 is composed equally of non-root nodes (NRN), horizontal, vertical, diagonal resolution root nodes as defined by the SPIHT algorithm (Said and Pearlman, 1996). 58

5.3 Block-based raster scan traversal of a matrix consisting of $2^{m-1} \times 2^{m-1}$ blocks, where m = 3. Each block contains a NRN (top-left), HR tree root node (top-right), VR tree root node (bottom-left), and DR tree root node (bottom-right). The matrix contains a total of $2^3 \times 2^3 = 64$ primary substreams. 59

5.4 Block based raster scan fixed-order assignment of $P = 2^3 \times 2^3 = 64$ primary substreams. 60

5.5 Block-based zig-zag scan traversal of a matrix consisting of $2^{m-1} \times 2^{m-1}$ blocks, where m = 3. Each block contains a NRN (top-left), HR tree root node (top-right), VR tree root node (bottom-left), and DR tree root node (bottom-right). The matrix contains a total of $2^3 \times 2^3 = 64$ primary substreams. 61

5.6 Zig-zag fixed-order assignment of $P = 2^3 \times 2^3 = 64$ primary substreams. 62

5.7 Dispersed dot-dither fixed-order assignment of $P = 64$ primary substreams. 63

5.8 Subband dispersed fixed-order assignment for $P = 64$ primary substreams ($m = 3$). 65

5.9 Fixed primary substream grouping for four groups ($N = 4$). Raster-scan fixed-order assignment (a). Zig-zag fixed-order assignment (b). The four groups are distinguished by the grayscale shading of the blocks. 66

6.1 Number of source symbols in each primary substream of the Peppers SPIHT multi-stream at $R_s = 0.1367$ bpp. 72

6.2 Number of source symbols in each group for Opt-ZZ M-UEP when $N = 10$ and $R_s = 0.1367$. 73

6.3 Number of source symbols in each group for FS-ZZ M-UEP when $N = 10$ and $R_s = 0.1367$. 74
6.4 Number of source symbols in each group for Opt-DD M-UEP when $N = 10$ and $R_s = 0.1367$. 75
6.5 Number of source symbols in each group for RS-DD M-UEP when $N = 10$ and $R_s = 0.1367$. 76
6.6 M-UEP PSNR vs. Number of Packets for peppers image at $R = 0.20$ and $\epsilon = 0.15$ when applying constrained optimal primary substream grouping. 77
6.7 M-UEP PSNR vs. Number of Packets for Peppers image at $R = 0.20$ and $\epsilon = 0.15$ when applying fixed-size primary substream grouping. . 78
6.8 FM-UEP PSNR vs. Number of Packets for Peppers image at $R = 0.20$ and $\epsilon = 0.15$ when applying constrained optimal primary substream grouping. 79
6.9 FM-UEP PSNR vs. Number of Packets for Peppers image at $R = 0.20$ and $\epsilon = 0.15$ when applying fixed-size primary substream grouping. . 80
6.10 Ratio between M-UEP side information and total bit budget vs. number of packets N, for Lena and Peppers, at $R = 0.50$ bpp. 81
6.11 M-UEP PSNR difference vs. UEP when transmitting M-UEP side information within the M-UEP packetization array for the Lena image at $R = 0.50$ bpp and $\epsilon = 0.15$. 82
6.12 M-UEP PSNR difference vs. UEP when transmitting M-UEP side information within the M-UEP packetization array for the Peppers image at $R = 0.50$ bpp and $\epsilon = 0.15$. 83

6.13 PSNR improvement of Opt-SD M-UEP over UEP vs. packet erasure rate ϵ for Lena and Peppers images at transmission rate $R = 0.50$ bpp and $N = 16$. The erasure protection is optimized for erasure rate $\epsilon = 0.15$. 83

A.1 Source symbol assignment pseudocode for layer j. 87

A.2 Example of source symbol assignment algorithm for layer $j = 4$ where $N = 6$ and $x_j = 3$. White boxes represent source symbols, and grey boxes represent redundancy symbols 88

B.3 Packetization array rows ℓ and $(\ell + 1)$ prior to (a) and after (b) the path change in Lemma 1. Gray boxes represent redundancy symbols and white boxes represent source symbols. 92

Chapter 1

Introduction

1.1 Transmission over IP Networks

Internet Protocol (IP) networks are packet-switching networks which employ the IP protocol for the transmission of datagrams (network layer packets). IP is the standard protocol for the network layer of the protocol stack. In fact, all Internet components which contain a network layer must run the IP protocol (Kurose and Ross, 2008). The Internet is the most well-known example of an IP network.

The IP protocol provides a *best-effort, egalitarian service* to all datagrams that it transmits over the network. The datagrams are routed on a first-come, first-serve (FIFO) basis and it does not provide the ability to reserve bandwidth or resources. It treats all packets equally and makes no guarantees about the performance of the transmission of a datagram from source to destination. In fact, the IP protocol does not even guarantee that the datagram will arrive at the destination (Kurose and Ross, 2008).

A *packet loss* occurs when a packet is transmitted from the source and does not

arrive successfully at the destination. One cause of packet loss is network congestion. When a router is over-congested with input packets, it must indiscriminately discard some incoming packets. Another cause of packet loss is bit errors. A packet can be received, but if it contains uncorrectable bit errors, it must be discarded at the receiver. Finally, transmission delay can also result in packet loss. If the delay from source to destination does not meet the real-time constraints of the application (i.e. image or video), the packet is essentially useless to the decoder, and thus is also regarded as a lost packet.

Since packet loss is a by-product of a best-effort service model, the challenge becomes maintaining an acceptable QoS in the presence of packet losses. The most typical approach for recovery from packet loss is re-transmission. The approach is ideal for such applications as Internet browsing, e-mail, and file transfers where additional transmission delay is acceptable, as long as the correctness of the data is maintained. However, the additional delay required for re-transmission may not be allowed in applications with strict delay constraints.

Forward error correction (FEC) coding is an approach to mitigating the effects of packet loss, which does not require re-transmission. FEC essentially appends controlled redundancy to the source bit stream to protect the source bit stream against potential bit errors and packet losses during transmission. Based on the strength of the FEC code, the decoder is able to detect and correct up to a certain number of bit errors or packet losses. There are many different methods that apply FEC. Unequal error protection is a type of FEC that is well suited for the transmission of embedded bit streams over IP networks.

1.2 Embedded Bit-streams

Embedded encoders generate *successively refinable* bit-streams. That is, bit-streams in which any prefix can reconstruct the image to a certain fidelity. The fidelity of the image increases in the length of the received prefix. Secondly, a bit in an embedded bit stream can only be decoded if all previous bits have been received and decoded. Clearly, the initial bits in the embedded bit stream are more important than those bits which proceed them. Additionally, since a symbol error renders all proceeding symbols useless to the decoder, the location of a symbol error directly affects its impact on the distortion performance at the decoder. With these facts in mind, it is intuitive to provide the strongest channel protection at the beginning of the embedded bit stream. Subsequent portions are then protected with channel codes of decreasing strength. This is the fundamental concept of unequal error protection, and is precisely why it is well suited to protect embedded bit streams against packet loss.

1.3 Unequal Error Protection Strategies for Embedded Bit-streams

Many approaches of unequal error protection for embedded image bit streams have been proposed. One such strategy was introduced by Chande and Farvardin (2000) to combat the impact of bit errors in a binary symmetric channel. Their scheme uses a concatenation of a rate compatible punctured convolutional (RCPC) code for error correction with a cyclic redundancy check (CRC) code for error detection. Concatenated codes of decreasing error correcting strength are applied to subsequent blocks of the embedded bit stream to achieve an optimal distribution of transmission

rate between the source bit stream and channel codes. This type of unequal error protection scheme produces variable length blocks, which all contain the same number of source symbols and varying amounts of channel symbols. Stankovic *et al.* (2003) proposed a similar scheme, which instead fixes the block length and varies the number of source symbols in each channel code block. Such a strategy proves to be more suitable when there is a constraint on the block size. These unequal error protection schemes can be optimized in terms of rate or distortion. Optimizing in terms of rate infers maximizing the expected number of correctly received source symbols. Distortion based optimization is focused on minimizing the expected distortion of the reconstructed bit stream at the receiver.

Unequal erasure protection has also been proposed to combat the impact of packet loss. In this thesis, we will use the acronym UEP for *unequal erasure protection*. UEP employs a collection of *strict* systematic Reed-Solomon (RS) block codes (i.e. where the source symbols are grouped at the beginning of the codeword) of the same length but decreasing strength, to protect subsequent segments of the source bit-stream. The packets to be sent through the lossy network are formed across the channel codewords. Any set of received packets can be used to reconstruct a prefix of the source stream which is decodable. Moreover, the fidelity of the reconstruction increases proportionally to the number of received packets. However, source symbols available at the decoder which do not belong to the reconstructed prefix cannot be decoded.

Mohr *et al.* (2000b) was one of the first works that applied UEP for transmission of embedded bit streams over packet lossy networks which considered the impact on distortion of each source symbol when determining the appropriate channel coding

strength. Further research of (Mohr *et al.*, 1999; Puri and Ramchandran, 1999; Mohr *et al.*, 2000b; T. Stockhammer, 2001; Stankovic *et al.*, 2004a; Thie and Taubman, 2005; Dumitrescu *et al.*, 2007) yielded optimal and sub-optimal distortion based optimization algorithms to achieve improved performance over Mohr's scheme. In this thesis we will take advantage of an inherent characteristic of embedded bit streams produced by embedded image encoders to improve upon the performance of previous unequal erasure protection techniques. For the remainder of this report, UEP will be used to refer to *unequal erasure protection.*

1.4 Error Protection Strategies for Multi-streams

A key property of scalable coders such as EZW (Shapiro, 1993), SPIHT (Said and Pearlman, 1996) and EBCOT (Taubman, 2000) is that they produce bit streams which are an interleaving of independently decodable and scalable substreams. We will refer to such bit streams as *multi-streams.* This multi-stream property of embedded image coders will be evident after the algorithms are described in Chapter 2.

Researchers have exploited the multi-stream characteristic to achieve improved resilience to errors and packet erasures. Creusere (1997) applied the multi-stream property of Shapiro's embedded zerotree algorithm (Shapiro, 1993) by dividing the wavelet coefficients into groups and encoding each group independently. The individual bit streams were then interleaved to form the multi-stream. This technique improves robustness since a bit error only affects the substream to which it belongs, while all other substreams can be fully decoded. Cho and Pearlman (2000) extended the concept of independently decodable substreams to scalable video (3D-SPIHT) and proposed equal error protection using a product-code of RCPC codes and CRC

codes to improve robustness against channel errors. Alatan *et al.* (2000) presented an unequal error protection method for SPIHT encoded bit streams which generates a multi-stream consisting of three substreams which are protected by varying strength RCPC/CRC codes. Kim *et al.* (2003) also proposed an unequal erasure protection method for transmission of video multi-streams over the internet. They generated the substreams by frequency domain partitioning of the wavelet coefficients, and determined the error protection for each substream independently. Thie and Taubman (2005) also exploited the multi-stream characteristic to create independent clusters of packets which are optimally protected using UEP.

Another method of achieving robustness was proposed by Rogers and Cosman (1998a). Error resilience is achieved by grouping the substreams into packets of fixed length. This method, which is tailored to transmission over packet erasure networks, ensurs that all packets are independently decodable. Thus errors cannot propagate beyond the packet in which they occur. Wu *et al.* (2001) proposed rate-distortion optimal packetization methods of multi-streams into packets of fixed length.

1.5 Contribution and Organization of Thesis

In the novel work of this thesis, the multi-stream characteristic of scalable code streams, along with the concept of independently decodable packets are exploited to achieve improved performance over the UEP framework. Independently decodable packets ensures that all received source symbols can be decoded, thus removing one of the disadvantages of the UEP framework. This is accomplished by assigning source symbols from a particular substream to a single packet. By enforcing such an

assignment, each packet will contain a prefix of a substream and hence it can be completely decoded, even in the absence of other packets. Additionally, we propose the use of *permuted systematic RS codes* instead of strict systematic RS codes, in order to maximize the flexibility of the erasure protection allocation. Permuted systematic RS codes are created by interleaving the source symbols with the redundancy symbols to form the channel codeword. This novel framework will be referred to as multi-stream unequal erasure protection (M-UEP). M-UEP was initially proposed in (Rivers *et al.*, 2008).

A similar idea was applied to multiple description coding in the thesis work of Zheng (2008). Termed MD-UEP, the framework can be regarded as a special case of M-UEP, applied to a multi-stream where all streams have identical RD-curves. However, the redundancy allocation problem with integer constraints was not discussed in that work.

Also, the transmission of separate substreams in separate packets, in conjunction with UEP by systematic RS codes has been proposed by Thomos *et al.* (2006a). However, the authors maintain the constraint of RS codes to be strict systematic. M-UEP eliminates this constraint, and thus allows for more flexibility in the erasure protection allocation. The same authors also present a product code technique (LDPC and RS codes) which employs symbol interleaving but does not consider independently decodable packets (Thomos *et al.*, 2006b). In their scheme, random symbol interleaving is performed within each packet to randomize the errors and improve the efficiency of the LDPC decoding. In the M-UEP framework, the symbol interleaving is performed across packets to provide flexibility in the placement of redundancy symbols.

In this thesis, The M-UEP rate-distortion optimal redundancy allocation (RD-ORA) problem is formulated as a constrained non-linear optimization problem. A globally optimal solution to the M-UEP RD-ORA problem is obtained by casting the problem into that of finding the maximum-weight path in a suitably constructed weighted directed acyclic graph. The globally optimal solution is shown to have a time complexity of $O(2^N N(L+1)^{N+1})$, where N is the number of packets and L is the packet length.

The globally optimal solution is intractable except for the cases when N is small. Therefore, a faster, sub-optimal solution to the M-UEP RD-ORA problem is formulated based on solution algorithms developed for UEP.

The necessary side information for the M-UEP strategy is presented, and an upper bound is derived. A few simple methods of transmitting this data are discussed. The M-UEP side information can become substantially large as N increases, and can reduce the benefit of M-UEP. A variant method of M-UEP is proposed, which requires no additional side information with respect to UEP. Fixed M-UEP (FM-MUEP) applies a structured source symbol placement given the number of rows in each layer.

To apply the M-UEP framework when N is smaller than the number of substreams P in the multi-stream, the substreams must be placed into N groups. To achieve the best M-UEP performance, the N groups must be well balanced in rate-distortion characteristics. An optimal algorithm for generating the N groups under the constraint of a fixed-ordering is presented. The fixed-ordering constraint enforces a strict order to the P substreams, and only consecutive substreams may be assigned to any of the N groups. The algorithm is derived from the the work of Wu *et al.* (2001). A number of

different substream fixed-orderings are presented, and their performance is evaluated for both M-UEP and FM-UEP and is compared to UEP performance. Experimental results are reported for two images (Lena, Peppers) at bit rates $R = 0.20, 0.50$ bpp, channel erasure rates $\epsilon = 0.05, 0.15$ for number of packets $N = 2, 4, \ldots, 40$. The performance of M-UEP under channel mismatch conditions is also evaluated. Experimental results reveal that M-UEP outperforms UEP with peak improvements exceeding 0.50 dB. M-UEP also demonstrates additional robustness under channel mismatch conditions, achieving improved performance over UEP when either fewer or more packets are lost than was estimated.

The remainder of this thesis is structured as follows. Chapter 2 provides a description of embedded bit streams in the context of embedded coding of images. Three embedded image encoders are discussed: EZW, SPIHT, and EBCOT. Chapter 3 provides a brief discussion of the traditional UEP strategy. It includes a formal description of UEP and the formulation of the optimal redundancy allocation problem in an rate-distortion sense. Chapter 4 focuses on the M-UEP framework. It formally describes the M-UEP framework and the corresponding rate-distortion optimal redundancy allocation (RD-ORA) problem. A globally optimal solution, and a sub-optimal solution are developed. Lastly, FM-UEP is introduced. Chapter 5 presents a number of grouping strategies for generating N grouped substreams from P primary substreams ($N < P$). Chapter 6 reports the experimental results of the performance evaluations of M-UEP and FM-UEP with respect to UEP. Conclusions and future research directions conclude the thesis.

Chapter 2

Embedded Multi-streams

An embedded coder is defined in Hamzaoui *et al.* (2005) as a source coder such that for any positive integers R_1 and R_2 where $R_1 < R_2$, the output bit-stream of the source coder for rate R_1 is a prefix of the output bit-stream of the source coder for rate R_2. Any source coder which satisfies this property produces embedded bit-streams. A key advantage of such an encoder is that given a set of known encoding rates $R_1, R_2, R_3, \ldots, R_N$ such that $R_1 < R_2 < R_3 < \ldots < R_N$ the encoder can generate a single output bit-stream for rate R_N and all other bit-streams can be produced by taking the appropriate prefix of the embedded bit-stream of rate R_N.

Efficient embedded image coders are EZW (Shapiro, 1993), SPIHT (Said and Pearlman, 1996), and EBCOT (Taubman, 2000). All these algorithms produce multi-streams. This aspect will be emphasized later when each algorithm is presented. The algorithms are different in nature, but all rely on wavelet based image decomposition. Each algorithm attempts to exploit the characteristics of the wavelet decomposition to efficiently represent the wavelet coefficients. Consequently, we begin this chapter with a brief overview of the concept of wavelets, and the process of wavelet decomposition

of an image. The second half of the chapter presents three wavelet based image encoders: EZW, SPIHT and EBCOT.

2.1 Wavelets

Though the concepts of wavelets can be traced back to the 1930s, Mallat (1987) was the first to apply wavelets as an approach to multiresolution analysis. *Multiresolution analysis*, also known as multiresolution theory, is based on the representation of signals (in our case images) at more than one resolution (Gonzalez and Woods, 2002). By applying multiresolution theory, an image can be decomposed into J resolutions consisting of a lowest resolution description (level 0) and a series of refinement layers (level 1 to level J-1). Wavelets are an essential component in performing such an image decomposition.

In the textbook of Gonzalez and Woods (2002), *wavelets* are defined as small waves of varying frequency and limited duration. A group of wavelet functions is described by its corresponding *mother wavelet* $\psi(x)$. Any discrete wavelet function $\psi_{m,n}(x)$ belonging to that group can be generated by a scaled and translated version of the mother wavelet $\psi(x)$

$$\psi_{m,n}(x) = \frac{1}{\sqrt{a_o^m}} \psi \left(a_o^{-m} x - n b_o \right) \tag{2.1}$$

where $m, n \in \mathbb{Z}$ and $a_o > 1$ and $b_o > 0$ are fixed (Antonini *et al.*, 1992).

Using the parameters m (scaling/frequency) and n (translation/location), discrete wavelets can be generated for specific positions and frequencies. This fundamental characteristic of wavelets proves to be one of its key advantages over the well-known

Fourier transform. The Fourier transform provides excellent frequency localization of the input signal, however, it provides no information about the instant in time (location in the signal) where the frequencies exist. By adjusting the parameters m and n, the wavelet transform provides the capacity to obtain both spatial (location) and frequency information (Gonzalez and Woods, 2002).

The most popular approach when performing a discrete wavelet decomposition of an image is to set $a_o = 2$ and $b_o = 1$ which yields the wavelet functions $\psi_{m,n}(x)$ in terms of the mother wavelet $\psi(x)$ as

$$\psi_{m,n}(x) = 2^{-m/2}\psi\left(2^{-m}x - n\right). \tag{2.2}$$

In this form, the wavelet functions $\psi_{m,n}(x)$ exist at twice the resolution of the wavelet functions $\psi_{m-1,n}(x)$ for all translations n. Therefore, with respect to each other, $\psi_{m,n}(x)$ would describe higher frequency (more detailed) information, while $\psi_{m-1,n}(x)$ would describe lower frequency (more general) information.

2.1.1 Wavelet Decomposition of Images

An image can undergo wavelet decomposition in a similar fashion to subband image decomposition. The 2-D image is decomposed using a two-stage filter bank where one stage performs 1-D filtering on the rows, while the other stage performs 1-D filtering on the columns. Figure 2.1 illustrates the two-stage filter bank for wavelet decomposition. Given an $M \times M$ input image, the resulting output wavelet coefficients represent a $M/2 \times M/2$ lower resolution approximation of the input image along with vertical, horizontal and diagonal higher resolution refinement information (each of size $M/2 \times M/2$). Note that by downsampling after each stage, the overall number

of resulting wavelet coefficients matches the number of original input pixels. In order to regenerate the input image, the wavelet coefficients are passed through a reverse filter bank with a similar structure to Figure 2.1 but the operations are performed in reverse order.

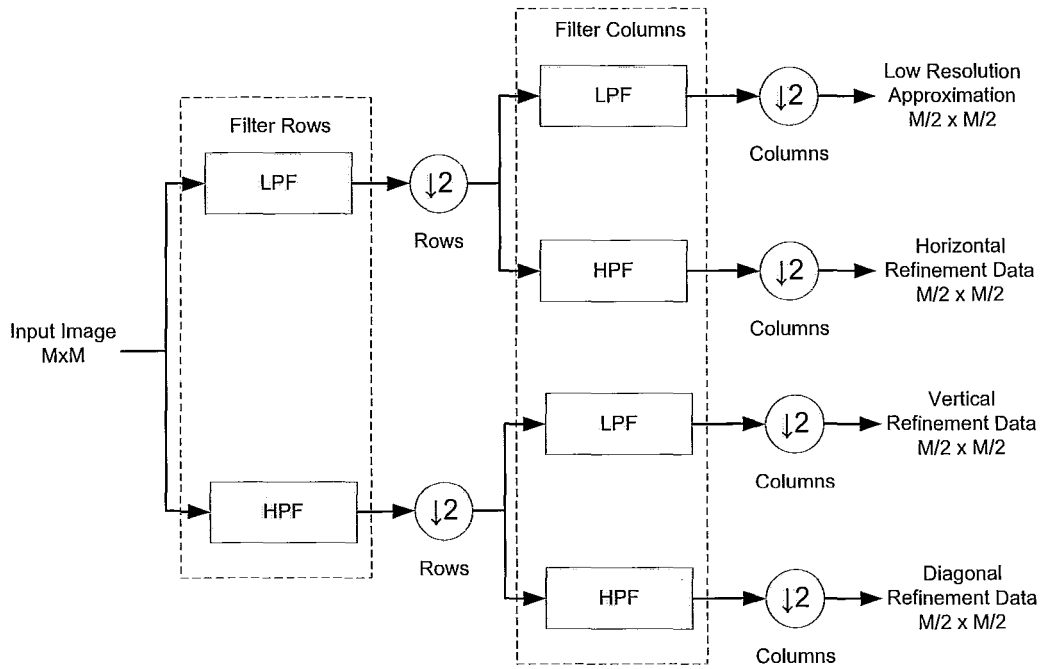

Figure 2.1: Single level 2-D wavelet decomposition of an $M \times M$ input image. The decomposition is performed by a two-stage filter bank with the first stage performing 1-D filtering of the rows of the image, and the second stage performing the 1-D filtering of the columns.

In most cases, the lower resolution approximation of the input image undergoes the decomposition process multiple times. This is accomplished by passing the lower resolution approximation wavelet coefficients through another identical two-stage filter bank. The result after passing an input image through Q filter banks is known as the Q-level wavelet decomposition. Figure 2.2 illustrates the resulting wavelet

coefficients after Q-level decompositions for $N = 1, 2, 3$. Note that only the lower resolution approximation coefficients at each level are further decomposed. The coefficients belonging to the regions HR_1 (level-1 horizontal refinement coefficients), VR_1 (level-1 vertical refinement coefficients)and DR_1 (level-1 diagonal refinement coefficients) remain the same in the resulting wavelet coefficient matrix after each of the 3 levels of wavelet decomposition. Each successive refinement of the lower approximation yields another level of refinement to the wavelet coefficient representation of the input image. When decoding the wavelet coefficients, each level provides additional refinement (higher frequency) information for the decoded image. Therefore, the wavelet decomposition strategy inherently produces multiple resolution representations of the input image, where each additional decomposition level improves the distortion characteristics of the resulting decoded image.

Lastly, it is important to note that wavelet coefficients at different decomposition levels are also spatially related. Since the wavelet decomposition provides both frequency and spatial information, there exists coefficients at each decomposition level that refer to the same spatial region in the image. Furthermore, when a multiple level wavelet decomposition is performed as illustrated in Figure 2.2, the number of coefficients at each decomposition level varies. Considering the 2-level decomposition, a single coefficient in the DR_2 region will be spatially related to 4 wavelet coefficients in the DR_1 region. This spatial relation between wavelet coefficients is exploited by efficient wavelet based image coders, as will be further discussed in the next section.

Such favourable characteristics as generating both spatial and frequency information, inherent multiresolution decomposition, and spatial relationships between wavelet coefficients at different refinement levels are all key benefits of applying

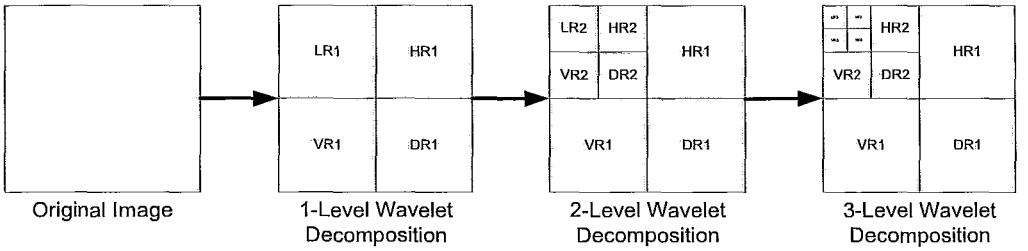

Figure 2.2: Illustration of a 3-level wavelet decomposition of an input image. LR = low resolution approximation coefficients, HR = horizontal refinement coefficients, VR = vertical refinement coefficients, DR = diagonal refinement coefficients.

wavelets to represent images (Sayood, 2006).

2.2 Wavelet Based Image Encoders

The characteristics of the wavelet decomposition of an image provide a means of producing highly efficient encodings of the wavelet coefficient matrix. More specifically, the spatial relation between wavelet coefficients located in different decomposition levels can be exploited to produce efficient embedded bit-streams representing the wavelet coefficients. The most well-known wavelet based image encoders are the Embedded Zerotree Wavelet (EZW) algorithm (Shapiro, 1993), Set Partitioning In Hierarchical Trees algorithm (SPIHT) (Said and Pearlman, 1996), and Embedded Block Coding with Optimized Truncation algorithm (EBCOT) (Taubman, 2000).

A key property of these wavelet based encoders is that they all produce bit-streams that consist of an interleaving of independently decodable and embedded substreams. Such bit-streams will be referred to as *multi-streams*.

2.2.1 Embedded Zerotree Wavelet (EZW) Algorithm

Shapiro (1993) was the first to propose an embedded algorithm for the encoding of a wavelet coefficient matrix of an image. The algorithm, entitled Embedded Zerotree Wavelet (EZW), exploits the temporal correlation between wavelet coefficients in different decomposition levels to form a tree structure connecting a root node in the lowest resolution approximation region (LR_Q for a Q-level decomposition) to all wavelet coefficients in the subsequent refinement levels (HR_Q to HR_1, VR_Q to $VR1$, and DR_Q to DR_1). The encoding scheme is based on the assumption that the wavelet coefficients closer to the root node have higher magnitudes, and thus contribute more to the overall distortion reduction of the image. Therefore, by encoding the coefficients with the largest magnitude first, the bit-stream effectively encodes the most relevant information first. Figure 2.3 illustrates the wavelet coefficients belonging to one tree in a 3-level wavelet decomposition. Note that the root node has three children, while all other nodes have four children.

The EZW algorithm classifies a wavelet coefficient in one of four categories: significant, insignificant, zero-tree root or isolated zero. The classification is based on a magnitude threshold value which is initialized to be $T_1 = 2^{\lfloor \log_2 c_{max} \rfloor}$ where c_{max} is the magnitude of the largest coefficient in the wavelet coefficient matrix. At the n^{th} iteration through the algorithm, the threshold is $T_n = T_1/2^{n-1}$. A coefficient is classified as *significant* if it has a magnitude greater than the current threshold T_n. A coefficient is *insignificant* if it has a magnitude less than the current threshold T_n. A coefficient is considered a *zero-tree root* if it along with all its descendants have magnitudes less than the current threshold T_n. Lastly, a coefficient is considered to be an *isolated zero* if it has a magnitude less than the current threshold T_n, but at

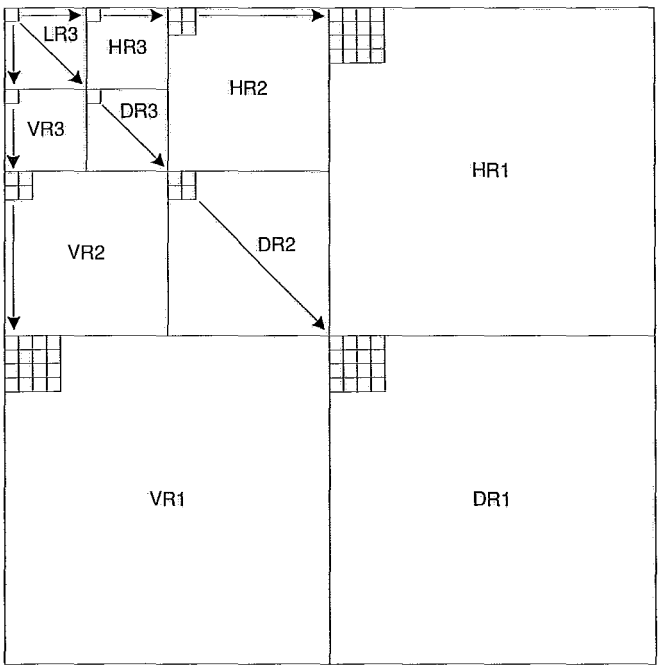

Figure 2.3: Illustration of an EZW tree in the wavelet coefficient matrix of a 3-level wavelet decomposition

least one of its descendants has a magnitude greater than T_n. The process of classifying wavelet coefficients into one of the four aforementioned categories is known as *significance mapping* (Sayood, 2006).

The EZW algorithm is an iterative algorithm, where each iteration consists of two steps. The first step is to determine the new significance mapping of all previously insignificant, zero-tree root and isolated zero coefficients with respect to the new threshold. Once this has been completed, the approximate value of all previously significant coefficients is adjusted based on their magnitude relative to the new threshold. This is known as the *refinement stage* (Sayood, 2006). Each significance mapping

requires two bits, while each refinement requires a single bit. At each iteration, the algorithm reduces the threshold by half, and repeats the process until a desired bit-rate or distortion performance has been achieved. The embedded bit-stream produced by the EZW algorithm is then passed through an arithmetic coder.

The key benefits of the EZW algorithm are that it is relatively simple conceptually, it attempts to achieve the best rate-distortion performance by encoding the coefficients with the largest magnitude first and it produces an embedded bit-stream. Additionally, it is clear that the produced embedded bit-stream is also a multi-stream since it is an interleaving of the encoding of the coefficients of each tree in the wavelet coefficient matrix.

2.2.2 Set Partitioning in Hierarchical Trees (SPIHT)

Said and Pearlman (1996) proposed an embedded coder, known as *Set Partitioning In Hierarchical Trees* (SPIHT), for the encoding of a wavelet coefficient matrix. SPIHT employs similar principles to Shapiro's EZW algorithm, but achieves improved performance. The algorithm improvements were obtained using a set partitioning scheme applied to the trees of the wavelet matrix. The tree relationships in SPIHT are different than those in EZW, and are referred to as *spatial orientation trees.* Figure 2.4 illustrates three spatial orientation trees and a non-root node that stem from a 2×2 block in the lowest resolution approximation region. All spatial orientation trees are generated in a similar fashion. The lowest resolution approximation region is divided into 2×2 blocks, with each block containing the root nodes for three spatial orientation trees and a single non-root node (top left coefficient in the 2×2 block). Generating the spatial orientation trees in the manner shown in Figure 2.4 tends to

better keep insignificant coefficients in larger subsets than the approach of EZW. Additionally, the partitioning decisions in SPIHT are binary decisions and only require a single bit, as opposed to EZW, which requires two bits for the significance mapping process (Sayood, 2006).

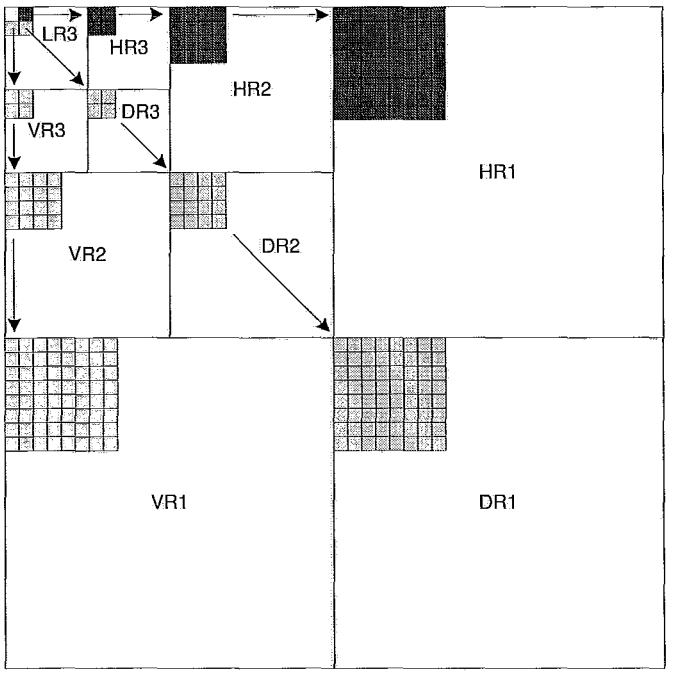

Figure 2.4: SPIHT algorithm spatial orientation tree relationships of a 2×2 block in the lowest resolution approximation region $LR3$ of the wavelet coefficient matrix. The wavelet matrix corresponds to a 3-level wavelet decomposition

SPIHT partitions the spatial orientation trees into four sets: $O(i,j)$, $D(i,j)$, $L(i,j)$, and H. The set $O(i,j)$ consists of the children of the wavelet coefficient at location (i,j). In the SPIHT scheme, a wavelet can have either four children or none. The set $D(i,j)$ contains all descendants of the coefficient at location (i,j). The set $L(i,j)$ consists of all descendants of the coefficient at location (i,j) excluding its children. Lastly, the set H consists of all root nodes.

The SPIHT algorithm is also an iterative algorithm where each iteration consists of a significance mapping stage and a refinement stage. The algorithm begins with an initial threshold determined in the same fashion as EZW to be $T_1 = 2^{\lfloor \log_2 c_{max} \rfloor}$. In SPIHT a set is considered *significant* if any coefficient in the set has a magnitude greater than the current threshold T_N. During the encoding process the coefficients of each tree are divided into three lists: *List of insignificant pixels (LIP), list of significant pixels (LSP)*, and *list of insignificant sets (LIS)*. The significance mapping stage considers all members of LIP and LIS, while the refinement stage provides a single bit refinement to each member of LSP. The SPIHT algorithm is described in its entirety in (Said and Pearlman, 1996).

SPIHT achieves performance improvements over EZW by utilizing a more effective partitioning of the wavelet coefficients into spatial orientation trees, and performing a more efficient significance mapping procedure (only a single bit). In fact, the SPIHT encoding achieves only a small performance degradation when eliminating the arithmetic encoding stage which proceeds the SPIHT encoding (Said and Pearlman, 1996). Lastly, note that the embedded bit-stream produced by the SPIHT algorithm also classifies as a multi-stream since it is formed by an interleaving of substreams. Each substream of the SPIHT multi-stream corresponds to a single non-root-node or a spatial orientation tree.

2.2.3 Embedded Block Coding with Optimal Truncation (EBCOT)

Embedded Block Coding with Optimal Truncation (EBCOT) (Taubman, 2000) is an embedded wavelet image coding scheme that is the basis for the JPEG 2000 image compression standard. It differs significantly from EZW and SPIHT because

it does not encode the wavelet coefficient matrix in its entirety, but instead breaks the wavelet coefficients into non-overlapping blocks and encodes each block independently. In the case of EBCOT, the compression performance is achieved using an entropy coder instead of exploiting the zero-tree relationship amongst the wavelet coefficients (Hamzaoui *et al.*, 2005).

EBCOT employs a two-tier coding strategy. The first stage involves independently encoding each individual block to produce an embedded bit-stream for each block. The block coding is based on a bit-plane encoding procedure followed by a context-adaptive arithmetic encoder. EBCOT also employs the concept of *significance mapping* during the bit-plane encoding procedure. Each block is sub-divided into *sub-blocks*, and each sub-block is classified as either significant or insignificant based on the current stage of the bit-plane encoding process. The second stage of EBCOT optimally interleaves sequences from each block bit-stream to generate the desired quality layers. The interleaving of independently encoded bitstreams is achieved using *post-compression rate-distortion (PCRD) optimization.* Each quality layer achieves the optimal distortion performance under the rate constraints. The EBCOT algorithm is discussed in its entirety in (Taubman, 2000).

The rich set of features (resolution scalability, SNR scalability and random access), exceptional compression performance, low memory requirements, and improved robustness against error propagation are all key reasons for the adoption of EBCOT for the JPEG 2000 image compression standard (Taubman, 2000; Hamzaoui *et al.*, 2005). Additionally, it is evident that EBCOT also produces a multi-stream since the embedded bitstream is composed of an interleaving of independently decodable substreams, one for each non-overlapping block in the wavelet coefficient matrix.

Chapter 3

Unequal Erasure Protection

This chapter discusses the traditional UEP strategy for transmission of a scalable bit-stream over a packet erasure network, using N packets of L symbols each. This structure can be envisioned as an array of packets which we will refer to as a *packetization array*. In this context, a *symbol* is a sequence of a fixed number of bits (usually 8 bits). This section commences with a formal description of the UEP framework. The problem of determining the optimal redundancy assignment for UEP is derived, and proposed solutions from previously published research are discussed. In this presentation we will assume that the embedded bit-stream is a multi-stream.

3.1 Formal Description of UEP

To generate the UEP packetization array, the embedded bit stream is partitioned into $L+1$ consecutive segments, where only the first L segments will be transmitted. The segments must have non-decreasing lengths $m_1 \leq m_2 \leq \ldots \leq m_L$ where m_i denotes the length of segment i. For each i, $1 \leq i \leq L$, the i-th segment is protected by a

systematic (N, m_i) RS code. By applying an (N, m_i) RS channel code, if at least m_i channel symbols are received at the decoder and the locations of the missing symbols are known, then all the N symbols of the channel codeword can be recovered. Note that since packet headers in packet-switched networks contain a sequence number, the positions of lost packets in the transmitted sequence can be determined at the decoder. The i-th channel codeword consisting of m_i source symbols, followed by $N - m_i$ redundancy symbols constitutes the i-th row in the packetization array. Each column of the packetization array forms a packet, and all rows with equivalent strength RS channel codewords form a layer. More formally, *layer* j $(1 \leq j \leq N)$ is the set of rows with (N, j) RS channel codewords and x_j represents the number of rows belonging to layer j. Figure 3.1 illustrates a UEP packetization array for $N = 4$, $L = 8$. Note that each row is composed of source symbols followed by redundancy symbols, and the number of redundancy symbols (i.e. the strength of the channel code) is non-increasing as the row index increases.

In the UEP framework, when only a subset k of the total number of transmitted packets N $(k \leq N)$ is received at the destination, all source symbols in the first k layers can be recovered due to the erasure protection. Since they form a prefix of the embedded bit stream, the source data can be reconstructed to a certain fidelity. The fidelity increases in the number of received packets. However, UEP does not guarantee that all received source symbols can be decoded. In fact, all source symbols in the received packets that do not occur in the first k layers cannot be decoded, and must be discarded at the receiver. We emphasize this disadvantage of UEP by means of an example.

Example 1: Consider the source multi-stream ($a1$, $b1$, $c1$, $d1$, $a2$, $b2$, $c2$, $d2$, $a3$, $b3$, $c3$, $d3$, $a4$, $b4$, $c4$, $d4$, $a5$, $b5$, $c5$, $d5$, ...), formed by interleaving four scalable streams A, B, C, D, where $x1, x2, x3, \ldots$, denote the symbols of stream X, $X \in \{A, B, C, D\}$. Consider now the UEP packetization with (m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8) = (1,1,2,2,3,3,4,4), which is illustrated in Figure 3.1. Let us analyze the case when only packets P_2 and P_4 arrive at the receiver. Due to the erasure protection, only the lost symbols in layers 1 and 2 can be recovered, i.e., $a1$, $b1$, $c1$, $a2$. Together with the source symbols from the received packets they form the prefix $(a1, a2)$ of stream A, prefix $(b1, b2)$ of B, prefix $(c1)$ of C and prefix $(d1, d2)$ of D, which are decoded. Note that the symbols $c3, b4, b5, d4, d5$ are also available at the decoder, but they cannot be decoded ($c3$ cannot be decoded because $c2$ is missing, $b4, b5$ cannot be decoded because $b3$ is missing, etc.). Therefore, message symbols $c3, b4, b5, d4, d5$ are essentially wasted at the decoder.

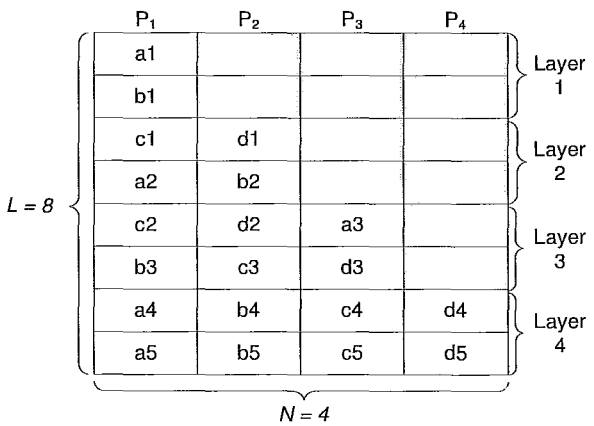

Figure 3.1: UEP packetization array for $N = 4$, $L = 8$. White boxes represent source symbols, grey boxes represent redundancy symbols.

Note that in order for the UEP strategy to achieve optimal performance in a distortion sense, the total transmission rate must be optimally allocated between

source symbols and channel protection such that the optimal channel protection for each transmitted source symbol is achieved subject to the rate constraints. The optimal channel protection for a particular source symbol is dependent on the channel conditions (i.e. packet loss rate) and the relative importance of the source symbol to the reconstructed fidelity of the source data (i.e. distortion reduction achieved by decoding the source symbol).

3.2 UEP Rate-Distortion Optimal Redundancy Allocation

Given a fixed transmission budget of $N \times L$ symbols (source and redundancy), the redundancy allocation that minimizes the expected distortion at the receiver must be determined. We will refer to this problem as the R-D optimal redundancy allocation (RD-ORA) problem.

This thesis considers the transmission over packet lossy networks modeled as an independent packet loss channel with erasure rate ϵ. Formally, let $P_N(k)$ denote the probability of losing k packets out of the N packets transmitted. Therefore $P_N(k)$ can be expressed as

$$P_N(k) = \binom{N}{k} \epsilon^k (1-\epsilon)^{N-k}, 0 \leq k \leq N. \tag{3.1}$$

The mean-squared error (MSE) is used as the distortion metric. Moreover, assume that the decoded symbols contribute additively to the total distortion reduction. This assumption holds true when MSE is used as the distortion metric and no error concealment is applied at the decoder. Furthermore, the distortion reduction contribution

of each source symbol in the multi-stream is known. Formally, for each substream i, let $\Delta D_i(r_i)$ denote the decrease in distortion due to decoding the r_i-th source symbol from sub-stream i. For the multi-stream, let $\Delta D(r)$ denote the decrease in distortion due to decoding the r-th source symbol from the multi-stream. When all packets are lost during transmission, let D_{max} be the maximum distortion at the receiver.

The UEP RD-ORA problem is a well-known problem that is reviewed here for completeness. Recall that when applying UEP, a source symbol situated in layer j will be recovered and decoded at the receiver if at least j packets are received. Therefore, the probability that a source symbol in layer j will be decoded, denoted as $C_U(j)$, in terms of $P_N(k)$ is expressed as

$$C_U(j) = \sum_{k=0}^{N-j} P_N(k)\,, 1 \leq j \leq N. \tag{3.2}$$

Let x_j denote the size of layer j (i.e., the number of rows allocated to layer j), $1 \leq j \leq N$. Then the number of source symbols in any layer ℓ is ℓx_ℓ. Consequently, the symbols situated in some layer j are all the symbols in the multi-stream between positions $(\sum_{\ell=1}^{j-1} \ell x_\ell) + 1$ and $\sum_{\ell=1}^{j} \ell x_\ell$, inclusive. Then the expected distortion of the reconstructed source at the receiver is

$$\bar{D}_U = D_{max} - \sum_{j=1}^{N} \left(\sum_{r=1+\sum_{\ell=1}^{j-1} \ell x_\ell}^{\sum_{\ell=1}^{j} \ell x_\ell} \Delta D(r) \right) \times C_U(j), \tag{3.3}$$

where D_{max} denotes the distortion when no source symbols are decoded.

The objective of the UEP R-D ORA problem is to find the non-negative integers

$x_1, x_2, \cdots, x_N$ which minimize $\bar{D}_U$ subject to the constraint

$$\sum_{j=1}^{N} x_j = L. \tag{3.4}$$

Note that the total number of feasible solutions to the RD-ORA problem is $\binom{L+N-1}{L}$, thus exhaustive search is impractical (Hamzaoui *et al.*, 2005). Many researchers have proposed methods for finding an optimal solution to the RD-ORA UEP problem. Dumitrescu *et al.* (2004) proposed a globally optimal solution with a time complexity of $O(N^2L^2)$. Assuming convexity of the R-D curve and some properties of the channel, they showed that the time complexity of the globally optimal solution reduces to $O(NL^2)$. Mohr *et al.* (2000a) proposed a $O(NL + hNlogN)$ time solution algorithm where h is the number of vertices on the convex hull of the R-D curve. Puri *et al.* (2001) presented a solution algorithm of $O(NL+kN)$ time, where k is the number of bisectional searches required for the Lagrangian minimization. Both algorithms are globally optimal under the convexity and fractional bit allocation assumptions. A locally optimal solution of time complexity $O(NL)$ was presented by Stankovic *et al.* (2004b). Thie and Taubman (2005) presented solutions which considered both linear and tree-structured dependencies. A more detailed discussion and comparison of proposed solutions to the RD-ORA UEP problem can be found in Hamzaoui *et al.* (2005).

3.3 UEP Side Information

In order to facilitate the successful decoding of a packetization array formed using UEP, additional side information is necessary at the decoder. Without the additional

side information, the decoder will not be able to appropriately interpret the contents of the packetization array as either source symbols or channel symbols. Therefore, the side information should be transmitted to the receiver via a secure channel. To completely characterize a packetization array generated by applying UEP, knowledge of the following is necessary:

1. The number of packets (N).
2. The packet length (L).
3. The number of source symbols in each row $m_\ell, 1 \leq \ell \leq L$.

Note that the number of source symbols can also be inferred by instead transmitting the number of rows in each layer j denoted by x_j. Applying the latter technique, information regarding layer N (the last layer) need not be transmitted.

The side information necessary to transmit the number of packet N is given by $\lceil \log_2(N_{max}) \rceil$ bits, where N_{max} is the fixed maximum allowable number of packets in a packetization array. Likewise, the side information necessary to transmit the packet length L is given by $\lceil \log_2(L_{max}) \rceil$ bits, where L_{max} is the fixed maximum allowable packet length. Both N_{max} and L_{max} are fixed quantities known by both the encoder and decoder.

There are numerous ways of transmitting the number of source symbols in each row $m_\ell, 1 \leq \ell \leq L$. Three such methods are presented here. *Method* 1 simply transmits each m_ℓ value ($1 \leq \ell \leq L$). This method requires $L \times \lceil \log_2(N) \rceil$ bits of side information.

The second and third methods involve transmitting the number of rows in each layer $x_j, 1 \leq j \leq N$. Since there are L rows in the packetization array, the following

equality must be satisfied:

$$\sum_{j=0}^{N} x_j = L. \tag{3.5}$$

As a result, it is sufficient to transmit the number of rows in each layer $x_j, 1 \leq j \leq N-1$ and the number of rows in layer N can be determined using equation (3.5). *Method* 2 transmits each x_j for $1 \leq j \leq N-1$. Since x_j is bounded by $0 \leq x_j \leq L$ the side information required is $\lceil \log_2(L+1) \rceil \times (N-1)$. However, when the number of packets N increases, many values of $x_j, 1 \leq j \leq N-1$ are zero, rendering this method inefficient. *Method* 3 addresses such scenarios by only transmitting non-zero values of $x_j, 1 \leq j \leq N-1$. This is accomplished by first identifying the non-zero layers j, and then transmitting the corresponding x_j values. The side information necessary when applying Method 3 is $\lceil log_2(N) \rceil \times (n_{x_j \neq 0} + 1) + n_{x_j \neq 0} \times \lceil log_2(L) \rceil$, where $n_{x_j \neq 0}$ denotes the number of non-zero layers.

Table 3.1: Summary of UEP channel side information schemes

Method	Transmit	Overhead (*in bits*)
1	$m_1, m_2, \ldots, m_L$	$L \times \lceil \log_2(N) \rceil$
2	$x_j,\ 1 \leq j \leq N-1$	$(N-1) \times \lceil \log_2(L) \rceil$
3	$x_j \neq 0, 1 \leq j \leq N-1$	$(n_{x_j \neq 0} + 1) \times \lceil log_2(N) \rceil + n_{x_j \neq 0} \times \lceil log_2(L) \rceil$

Table 3.1 summarizes the proposed UEP channel side information representation schemes. It is clear that Method 1 is the best method to apply when the packets length L is smaller than the number of packets N $(N > L)$. When $(N < L)$, the value of $n_{x_j \neq 0}$ determines if either Method 2 or Method 3 is the most efficient representation of the UEP channel side information data. To easily accommodate all three methods, a 2 bit marker could be added after transmitting the packet length L which can be used to identify which technique will be applied to encode the header information.

Note that the above techniques for transmitting the side information do not encompass all possible methods of transmitting the necessary UEP side information to the decoder. Compression techniques could be applied to minimize the necessary UEP side information, however this is not the main focus of this thesis work.

Chapter 4

Multi-stream Unequal Erasure Protection

This chapter describes the multi-stream unequal erasure protection (M-UEP) strategy for transmission over packet erasure networks. The M-UEP packetization array consists of N packets of L symbols each, giving a total transmission budget of $N \times L$ symbols. The chapter begins with a formal description of the M-UEP strategy and provides an example to illustrate its benefit over UEP for the transmission of multi-streams. Next, the RD-ORA problem for M-UEP is formulated and the additional complexity in solving the problem is emphasized. The chapter continues by presenting both a globally optimal graph-based solution, as well as a sub-optimal solution to the M-UEP RD-ORA problem. The necessary side information to completely describe the M-UEP packetization array is discussed and an upper bound on the M-UEP side information is derived. Lastly, an M-UEP variant which imposes fixed redundancy locations is presented. Termed FM-UEP, this framework eliminates the M-UEP side information by using a consistent source symbol assignment algorithm

given the number of rows in each layer $x_j, 1 \leq j \leq N$.

4.1 Formal Definition of M-UEP

Assume that the source multi-stream is composed of N interleaved, independently decodable embedded substreams. Therefore, each packet in the packetization array corresponds to a single substream. In the M-UEP framework, the source symbols in packet i constitute a prefix of substream i. Each row of the packetization array is formed by a *permuted* systematic RS channel codeword, which is obtained from a systematic RS codeword, by applying a permutation to the channel symbols of the codeword. This causes the source symbols to be interleaved with the redundancy symbols. Clearly, the erasure protection capabilities are not affected. In other words, an (N, k) *permuted* systematic RS code is able to correct up to $N - k$ erasures. As is the case in the UEP framework, the strengths of the RS channel codewords in the M-UEP framework must be non-increasing as the row index increases.

Let x_j denote the number of rows in layer j, and let $x_j^{(i)}$ be the number of source symbols from substream i (packet i), situated in layer j, $1 \leq i, j \leq N$. Due to the constraint of non-increasing strengths of the RS codes, the rows belonging to any layer are consecutive. Figure 4.1 illustrates an M-UEP packetization array for $N = 4$, $L = 8$. Note that each row is composed of a permuted systematic RS codeword, and the strength of the RS channel codes is non-increasing as the row index increases. It is also important to note that all source symbols in a packet belong to the same substream.

The structure of M-UEP ensures that all received source symbols can be completely decoded at the destination. When only a subset k of the N transmitted

packets N $(k \leq N)$ is received at the destination, all source symbols in the first k layers can be recovered and decoded due to the erasure protection. Additionally, all source symbols in layers $k+1$ through N from the received packets can also be decoded. By enforcing message symbol assignment from substream i $(1 \leq i \leq N)$ exclusively to packet i, all prefix symbols necessary to decode the message symbols in layers $k+1$ through N from the received packets are present at the decoder. This key benefit of M-UEP over UEP for the transmission of multi-streams is illustrated by the following example.

Example 2: Consider the same source multi-stream as in Example 1 of Chapter 3. Figure 4.1 illustrates an M-UEP packetization array where the streams A, B, C, D are assigned to packets P_1, P_2, P_3, P_4, respectively. Furthermore, each source symbol has the same erasure protection as in the UEP scenario of Example 1. Let us revisit the case when only packets P_2 and P_4 are received. In the context of M-UEP, source symbols in the first two layers from the lost packets $(a1, a2, c1)$ can be recovered. Thus, prefixes $(a1, a2)$ of stream A and $(c1)$ of C are formed and decoded. Moreover, the source symbols from the received packets form prefixes $(b1, b2, b3, b4, b5)$ of B and $(d1, d2, d3, d4, d5)$ of D which are also decoded. Thus, the reconstruction is strictly better than in the UEP case.

The advantage of M-UEP over UEP is clear in Example 2. Notice that in this example the redundancy assignment to source symbols is identical to the redundancy assignment achieved under the UEP framework. When this happens, in other words when any M-UEP layer j contains the same source symbols as the UEP layer j, the M-UEP technique ensures a performance improvement in any situation when only a

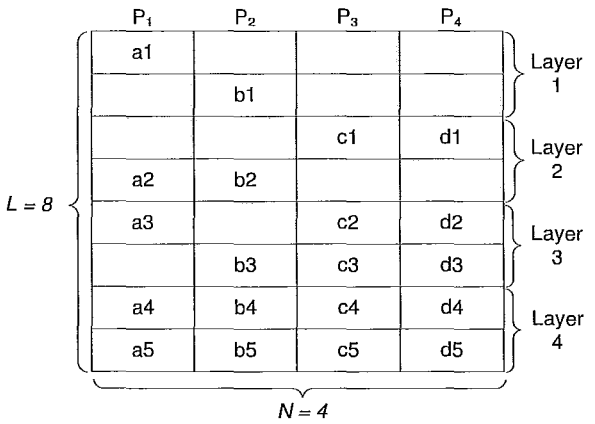

Figure 4.1: M-UEP packetization array for $N = 4$, $L = 8$. White boxes represent lost source symbols, light grey boxes represent received source symbols and grey boxes represent redundancy symbols.

subset of $k, k < N$ packets is received. This is because the M-UEP guarantees the decoding of all source symbols that would have been decoded under the UEP scenario (i.e., all source symbols in layers 1 through k), and additionally decodes all source symbols in layers $k+1$ through N of the received packets.

The ability of the M-UEP framework to ensure the same redundancy allocation as the UEP is influenced by the nature of the multi-stream. A sufficient condition for this property to hold is the multi-stream to be balanced. A multi-stream is balanced when symbols of the same importance in R-D sense are evenly distributed amongst the substreams. Assuming R-D optimal interleaving of substreams in the multi-stream, a balanced multi-stream ensures that each UEP layer j contains approximately equal number of symbols from each substream. If x_j denotes the number of rows in UEP layer j, then approximately jx_j/N source symbols from each substream are contained in this layer. Since $jx_j/N \leq x_j$, these source symbols can be arranged to fill x_j rows of layer j in the M-UEP packetization array, such that each column contains only

symbols from the corresponding substream (a rigorous proof of this claim is provided in Appendix A).

When the multi-stream is unbalanced, the constraint of assigning source symbols from substream i only to packet i may prevent M-UEP from achieving identical redundancy protection for each source symbol as with UEP. However in the case of a mildly unbalanced multi-stream, M-UEP can still outperform UEP, especially under poor channel conditions. This is due to the fact that when only a small number of packets are received, the number of additional symbols decoded in the M-UEP framework is large enough to compensate for the unavailability of a few important source symbols which might have been available to the decoder in the UEP case. On the other hand, if the multi-stream is severely unbalanced, some important symbols might not even be transmitted with M-UEP since they do not fit into the corresponding packet. This loss might offset the contribution of the less important additional symbols decoded. However, it is difficult to provide a precise characterization of the degree of imbalance which deprives the M-UEP of its advantage over UEP.

We should also note that when the number of primary substreams P in the multi-stream is larger than the number of transmission packets N, $(P \geq N)$, then the primary substreams must be grouped to form N substreams. Numerous grouping strategies are disucussed in Chapter 5, and their performance is reported in the experimental results of Chapter 6.

4.1.1 M-UEP RD-ORA

When applying M-UEP, a source symbol situated in layer j will be recovered and decoded at the receiver if either the packet containing the symbol is received at the

destination, or this packet is lost and at least j other packets are received, ensuring that the source symbol can be recovered by the erasure protection. The probability of the former event is $1-\epsilon$, while the probability of the latter event is $\sum_{k=0}^{N-1-j} P_{N-1}(k)$. Therefore, the probability that a source symbol in layer j will be decoded is

$$C_M(j) = 1 - \epsilon + \epsilon \sum_{k=0}^{N-1-j} P_{N-1}(k), 1 \leq j \leq N. \tag{4.1}$$

Given a source symbol allocation defined by the terms $x_j^{(i)}, 1 \leq j,i \leq N$, the expected distortion of the reconstructed source at the receiver is

$$\bar{D}_M = D_{max} - \sum_{j=1}^{N} \left(C_M(j) \sum_{i=1}^{N} \left(\sum_{r_i=1+\sum_{\ell=1}^{j-1} x_\ell^{(i)}}^{\sum_{\ell=1}^{j} x_\ell^{(i)}} \Delta D_i(r_i) \right) \right). \tag{4.2}$$

To determine the M-UEP RD-ORA, we must minimize $\bar{D}_M$ over all non-negative integers $x_j, x_j^{(i)}$, $1 \leq i,j \leq N$, subject to the constraints

$$\sum_{j=1}^{N} x_j = L, \tag{4.3}$$

$$\sum_{i=1}^{N} x_j^{(i)} = j x_j \; \forall j, \; 1 \leq j \leq N, \tag{4.4}$$

$$x_j^{(i)} \leq x_j \; \forall i,j, \; 1 \leq i,j \leq N. \tag{4.5}$$

It is easy to see that the above constraints are necessary. Constraint (4.3) ensures that the total rows assigned to all layers equals the transmission budget of L rows. The constraints of (4.4) follow from the fact that any row of layer j must have exactly j source symbols. Finally, the constraints of (4.5) are due to the fact that symbols

from substream i can only be placed in packet i, thus the number of source symbols assigned to layer j from substream i cannot exceed the number of rows assigned to layer j

The sufficiency of the above constraints (4.3)-(4.5) is proven by showing that for any integers $x_j, x_j^{(i)}$, satisfying constraints (4.3)-(4.5) there is a corresponding arrangement of the source symbols in the packetization array compatible with the M-UEP framework. The complete proof is provided in Appendix A.

4.2 Graph-Based Globally M-UEP RD-ORA Solution

In this section a weighted directed acyclic graph G is derived and it is shown that the M-UEP RD-ORA problem is equivalent to the maximum-weight path problem in G subject to a constraint. The time complexity of the M-UEP RD-ORA problem for both the general case and the case when the substream R-D curves are convex are derived using the one-to-one correspondence between the M-UEP packetization arrays and paths in G with some constraints.

4.2.1 Weighted Directed Acyclic Graph Formulation

Consider the weighted directed graph $G = (V, E)$, where V is the set of vertices (or nodes) and E is the set of directed edges. The nodes of the graph are all $(N+1)$-tuples of integers $\left(v^{(1)}, v^{(2)}, \ldots, v^{(N)}, \ell\right)$, such that $0 \leq \ell \leq L$, and $0 \leq v^{(i)} \leq \ell$, for all $i \in \{1, 2, \ldots, N\}$. The edges are all ordered pairs of vertices $(\mathbf{u}, \mathbf{v})$, where $\mathbf{u} = \left(u^{(1)}, u^{(2)}, \ldots, u^{(N)}, \ell\right)$ and $\mathbf{v} = \left(v^{(1)}, v^{(2)}, \ldots, v^{(N)}, \kappa\right)$ such that the following

conditions are satisfied

$$\kappa = \ell + 1, \tag{4.6}$$

$$0 \leq v^{(i)} - u^{(i)} \leq 1 \, \forall \, i, 1 \leq i \leq N, \tag{4.7}$$

$$\sum_{i=1}^{N} \left(v^{(i)} - u^{(i)}\right) \geq 1. \tag{4.8}$$

Each edge $(\mathbf{u}, \mathbf{v})$ is assigned a weight $w(\mathbf{u}, \mathbf{v})$ defined as

$$w(\mathbf{u}, \mathbf{v}) = \left(\sum_{i=1}^{N} \left((v^{(i)} - u^{(i)}) \Delta D_i \left(v^{(i)} \right) \right) \right) \times C_M \left(\sum_{i=1}^{N} (v^{(i)} - u^{(i)}) \right). \tag{4.9}$$

The source node of the graph is $\mathbf{v}_0 = (0, 0, \ldots, 0, 0)$, and the final nodes are all vertices whose last component is L. A path in G is any sequence of nodes starting with the source node and ending with a final node, such that any two consecutive nodes are connected by an edge. The weight of the path is defined as the sum of the weights of its edges. Note that the last component of the i-th node of the path must necessarily be $i - 1$. Therefore, any path in G has exactly $L + 1$ nodes, and hence L edges.

4.2.2 Correspondence between G and M-UEP Packetization Array

Any M-UEP packetization array PA can be assigned a path P_{PA} in G as follows. For each row ℓ $(1 \leq \ell \leq L)$, and each column i $(1 \leq i \leq N)$, let $v_\ell^{(i)}$ denote the number

of source symbols in the first ℓ rows of column i. Let $\mathbf{v}_\ell = \left(v_\ell^{(1)}, v_\ell^{(2)}, \ldots, v_\ell^{(N)}, \ell\right)$, $1 \leq \ell \leq L$, be the vertex representing the source symbol assignment in the first ℓ rows of the packetization array. Then the path P_{PA} is defined as the sequence of vertices $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_L$. Note that the ℓ-th path edge $(\mathbf{v}_{\ell-1}, \mathbf{v}_\ell)$ corresponds to the ℓ-th row in the packetization array. Moreover, the weight of this edge equals the contribution of the source symbols in the ℓ-th row to the decrease of $\bar{D}_M$ in (4.2). Therefore, the weight of the path $w(P_{PA})$ satisfies

$$w(P_{PA}) = D_{max} - \bar{D}_M. \tag{4.10}$$

Figure 4.2 provides an example of an M-UEP packetization array (a) and its corresponding path in G (b).

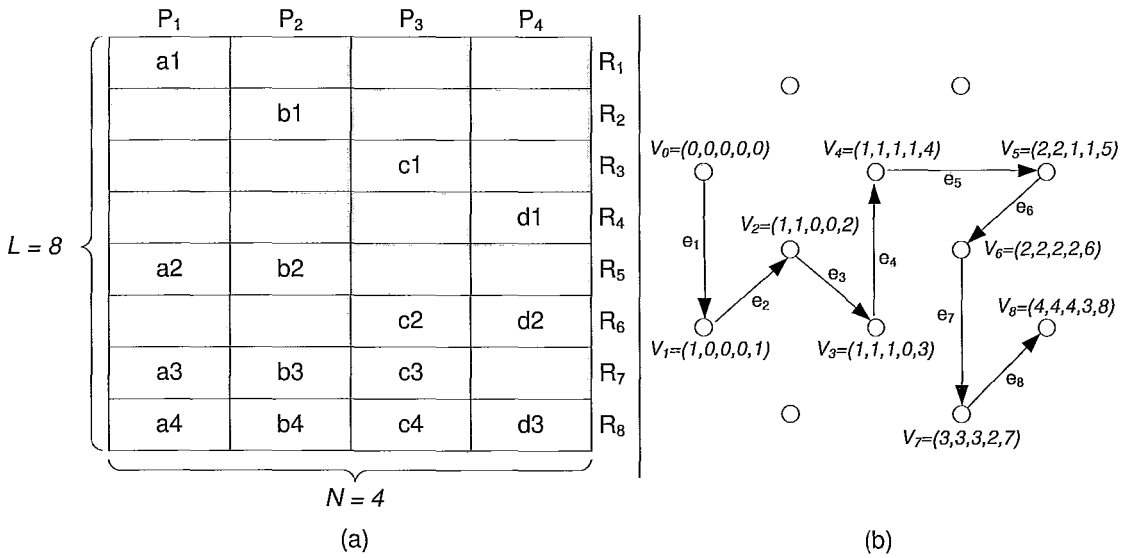

Figure 4.2: Example of the one-to-one correspondence between a packetization array $PA_{4\times8}$ and a path in $G_{4\times8}$. Gray boxes represent redundancy symbols and white boxes represent source symbols. The source symbols from each substream are assigned to a single packet ($A = P_1, B = P_2, C = P_3, D = P_4$).

Conversely, for a path in G to correspond to a packetization array it must ensure non-increasing redundancy assignment for consecutive rows. In other words, a path $P = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_L)$, corresponds to a packetization array if and only if the following relation holds true

$$\sum_{j=1}^{N} \left(v_{\ell}^{(j)} - v_{\ell-1}^{(j)} \right) \leq \sum_{j=1}^{N} \left(v_{\ell+1}^{(j)} - v_{\ell}^{(j)} \right), \tag{4.11}$$

for any $1 \leq \ell \leq L-1$. That is, the number of source symbols assigned on any row must be equal to or greater than the number of source symbols assigned on the previous row.

It is easy to see that the correspondence between packetization arrays and paths which satisfy condition (4.11) is one-to-one. Furthermore, equation (4.10) shows a clear connection between the path weight and the expected distortion. Therefore, the M-UEP RD-ORA problem can be formulated as the problem of finding the maximum-weight path among all paths in G which satisfy constraint (4.11).

4.2.3 Unconstrained Maximum-Weight Path Problem Formulation

The M-UEP RD-ORA problem can be cast as an unconstrained maximum-weight path problem based on a modification to the derived graph G. For the general case, a slightly different graph G_{new} in considered, for which the definition of a path incorporates constraint (4.11).

G is modified slightly to form G_{new} by adding an additional coordinate to each node corresponding to the layer of any incoming edge. Hence, the vertices of G_{new} are

all $(N+2)$-tuples of integers $(v^{(1)}, v^{(2)}, \ldots, v^{(N)}, \ell, n)$ such that $0 \leq \ell \leq L$, $0 \leq v^{(i)} \leq \ell$, and $0 \leq n \leq N$. The edges of G_{new} are all ordered pairs of vertices $(\mathbf{u}, \mathbf{v})$, where $\mathbf{u} = \left(u^{(1)}, u^{(2)}, \ldots, u^{(N)}, \ell, n\right)$ and $\mathbf{v} = \left(v^{(1)}, v^{(2)}, \ldots, v^{(N)}, \kappa, m\right)$ satisfying (4.6) and (4.7) with the additional constraints

$$m \geq n, \tag{4.12}$$

$$\sum_{i=1}^{N} \left(v^{(i)} - u^{(i)}\right) = m. \tag{4.13}$$

The source node of the graph is the node with all zero components. The final nodes are all nodes whose second last component is L.

There is a clear one-to-one correspondence between any path in G which satisfies constraint (4.11) and a path in G_{new}. Therefore, the M-UEP RD-ORA problem is equivalent to determining the maximum-weight path in G_{new}. The time complexity to solve the latter problem is $O(|V_{new}| + |E_{new}|)$. The number of vertices $|V_{new}|$ is clearly $O(N(L+1)^{N+1})$. Since the number of incoming edges to some node is at most 2^N, it follows that $|E_{new}| = O(|V_{new}| \times 2^N)$. We conclude that the globally optimal solution to the problem of R-D optimal M-UEP packetization can be found in $O(2^N N(L+1)^{N+1})$ time. Note that this graph-based approach for solving the optimization problem yields a drastic reduction in time complexity compared to exhaustive search $O((L+1)^{N^2+N})$. For the case when all substreams have convex R-D curves, Appendix B proves that constraint (4.11) can be safely removed because the maximum-weight path in G satisfies it. Thus the optimal solution can be found in $O(2^N(L+1)^{N+1})$ time by applying the convexity assumption.

4.3 Sub-optimal Solution to M-UEP RD-ORA

Except for small N, it is evident that the globally optimal solution to the M-UEP RD-ORA problem is intractable. Therefore, we propose a sub-optimal solution based on the solution of the UEP RD-ORA problem. To justify the sub-optimal solution method we assume convexity of the R-D curves from each substream of the multi-stream. Convexity of the R-D curves is a good approximation in practical scenarios for both SPIHT (Said and Pearlman, 1996) and EBCOT (Taubman, 2000) as both ensure near convexity of the R-D function for individual substreams.

The proposed approximate solution of the M-UEP RD-ORA problem can be split into two main steps. Step 1 is to solve the M-UEP RD-ORA problem disregarding the constraints of (4.5), yielding the redundancy assignment to each row in the packetization array. Step 2 involves keeping the redundancy assignment for each row fixed (hence fixing all x_j), and populating the packetization array while ensuring that the constraints of (4.4) and (4.5) are satisfied.

The key observation that simplifies the first step is that in the optimal solution to the problem of Step 1, each symbol in layer j must contribute a distortion reduction at least equal to the distortion reduction of any symbol in layer $j+1$. This statement holds if the R-D curves of all sub-streams are convex ($\Delta D_i(r_i) \geq \Delta D_i(r_i+1)$ for any $i = 1, \cdots, N$), and the sub-streams are interleaved in an R-D optimal fashion to form the multi-stream ($\Delta D(r) \geq \Delta D(r+1)$ for any $r \geq 1$). To prove this, we consider the scenario when it is not satisfied. Let a be the last source symbol of packet P_A in layer j and b be the first source symbol of packet P_B in layer $j+1$. Let $\Delta(a), \Delta(b)$ denote the contribution of symbol a and b respectively to the decrease in distortion. Assume that $\Delta(b) > \Delta(a)$. This scenario is illustrated in Figure 4.3(a) for $P_A = P_N - 1$ and

$P_B = P_2$. If we promote source symbol b to layer j and demote source symbol a to layer $j+1$ constraints (4.3), (4.4), and (4.5) are still satisfied. This new scenario is illustrated by Figure 4.3(b). By performing this source symbol swap between layers j and $j+1$, $\bar{D}_M$ will decrease by the amount $(C_M(j) - C_M(j+1))(\Delta(a) - \Delta(b)) > 0$, thus contradicting the optimality of the solution.

The above considerations imply that in the optimal solution to step 1, all symbols situated in a layer form a continuous segment of the multi-stream (since in the multi-stream the symbols are ordered in decreasing order of their contribution to the decrease of distortion). Then the cost function defined in (4.2) can be rewritten as

$$D_{max} - \sum_{j=1}^{N} C_M(j) \left(\sum_{r=1+\sum_{\ell=1}^{j-1} \ell x_\ell}^{\sum_{\ell=1}^{j} \ell x_\ell} \Delta D(r) \right), \tag{4.14}$$

and the problem reduces to minimizing (4.14) over all non-negative integers x_j satisfying (4.3). This optimization problem is identical to the UEP RD-ORA problem solved by (Dumitrescu *et al.* (2004), Mohr *et al.* (1999), Puri and Ramchandran (1999), Stankovic *et al.* (2004a)), where the only difference is in the calculation of the weights $C_M(j)$. Therefore any algorithm which solves the UEP RD-ORA problem can be used to complete step 1.

At Step 2 we fix the values x_j output at Step 1 and compute the values $x_j^{(i)}$ such that the constraints of (4.4) and (4.5) to be satisfied, by using the algorithm whose pseudocode is provided in Figure 4.4. The algorithm proceeds in increasing order of j. For each j, jx_j iterations are performed. At each iteration exactly one source symbol is assigned to layer j. The algorithm maintains pointers for each stream to the current

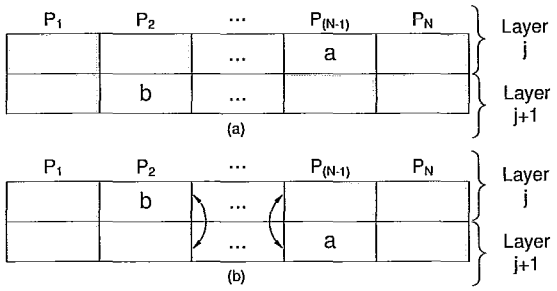

Figure 4.3: M-UEP packetization array layers j and $j+1$ prior to swapping source symbols (a), and after swapping source symbols (b). Gray boxes represent redundancy symbols and white boxes represent source symbols.

candidate symbol to be placed in layer j (i.e., the first symbol unassigned a layer yet). Also a list $\mathcal{I}$ is maintained of streams for which the number of symbols already assigned to layer j, is smaller than x_j (hence streams in $\mathcal{I}$ have not yet fully occupied their designated capacity in layer j). The symbol to be assigned at each iteration is the symbol with highest distortion reduction among all candidate symbols from streams in list $\mathcal{I}$. Clearly, this algorithm ensures that the output satisfies conditions (4.4) and (4.5). The number of operations performed is $O(\sum_{j=1}^{N} jx_j) = O(NL)$. We conclude that completing steps 1 and 2 requires $O(N^2L^2)$ time.

It is instructive to identify cases when the proposed solution to the M-UEP RD ORA problem is optimal. For the rest of this section we assume that the streams R-D curves are convex and the interleaving in the multi-stream is R-D optimal. As we have discussed, in this situation, the output of Step 1 is the optimal solution to the problem of minimizing (4.2) subject to constraints (4.3) and (4.4). Consequently, the optimal value of the cost function in (4.14), denoted by $\bar{D}_{opt}$, is a lower bound for the minimum M-UEP expected distortion. Also denote by $\bar{D}_{M,opt}$, the M-UEP expected distortion obtained by using the algorithm described in this section. Further, let

U_j denote the segment of the multi-stream between positions $(\sum_{\ell=1}^{j-1} \ell x_\ell) + 1$ and $\sum_{\ell=1}^{j} \ell x_\ell$, inclusive, and let $u_j^{(i)}$ denote the number of source symbols from substream i, situated in segment U_j, for all i, j. It is easy to see that if conditions

$$u_j^{(i)} \leq x_j \; \forall i, 1 \leq i \leq N, \tag{4.15}$$

are satisfied for all $1 \leq j \leq N$, then the algorithm of Step 2 ensures that $x_j^{(i)} = u_j^{(i)}$ for all i, j, which, in turn, implies that $\bar{D}_{M,opt} = \bar{D}_{opt}$. This guarantees that the algorithm's output is the optimal solution to the M-UEP RD ORA problem.

Note that the total number of symbols in U_j is jx_j. Hence condition (4.15) means that segment U_j contains at most $1/j$ of its total number of symbols coming from any single stream. This can be interpreted as a balance constraint imposed on U_j, which becomes stronger as j increases. Indeed, for $j = N$ this condition is satisfied if and only if all streams contribute an equal number of symbols to U_j, while as j decreases the constraint becomes more relaxed.

Note that conditions (4.15), depend on the partition of the multi-stream into segments U_j, which, in turn, depends on the problem instance (i.e., the RD curve, the channel statistics, the values of N and L). Given a multi-stream and N, L, the algorithm solution could be optimal for some channels and non-optimal for others. A sufficient condition which ensures optimality of the solution for all channels is the multi-stream to be *perfectly balanced.* We say that the multi-stream is perfectly balanced if any segment of N consecutive symbols of the multi-streams contains exactly one symbol from each stream. But clearly, given a channel, this sufficient condition is too strong. We conclude informally, that the closer the multi-stream is to a perfectly balanced one (i.e., the milder the unbalance), the higher is the range

of channels for which the algorithm in this section ensures optimal solution.

We also expect that in case of a mildly unbalanced multi-stream, if the solution is not optimal, it is still close to optimal. An intuitive explanation is the following. Violations of relations (4.15) cause excess symbols which do not fit into the layer allocated according to Step 1, to be placed into subsequent layers, thus reducing their protection level. If these violations are mild, in other words, if, for those values i, j for which $u_j^{(i)} \leq x_j$ does not hold, the excess $u_j^{(i)} - x_j$ is small, it is expected that the excess symbols will fit into the next layer. Since this leads to only a small change in the protection level (especially when N is high), it is likely that $\bar{D}_{M,opt}$ increases only slightly from the lower bound. Since $\bar{D}_{M,opt} - \bar{D}_{opt}$ is an upper bound for the gap between $\bar{D}_{M,opt}$ and the optimal solution, this guarantees only a small degradation from the optimum. In the case of severe unbalance, Step 2 will cause high changes in protection level of source symbols, pushing $\bar{D}_{M,opt}$ far away from the lower bound. In such a case the proposed redundancy allocation algorithm does not have any performance guarantee.

4.4 M-UEP Side Information

For correct decoding of a M-UEP packetization array, the decoder requires complete knowledge of the UEP side information discussed in Chapter 3, as well as the number of source symbols from each substream i ($1 \leq i \leq N$), in each layer j ($1 \leq j \leq N-1$). This is exactly the values $x_j^{(i)}$. Therefore since the non-zero layers will be known at the decoder from the UEP side information, it is only necessary to transmit the values $x_j^{(i)}$ for the non-zero layers $(x_j \neq 0), 1 \leq j \leq N-1$, for each substream i, $1 \leq i \leq N-1$. Additionally, it is not necessary to transmit $x_j^{(N)}$ since these values can be

```
current (i) = index of current candidate symbol of stream i
I = set of stream indices i ∈ {1, 2, 3, ..., N} that satisfy x_j^(i) < x_j
set current(i) = 0 and x_j^(i) = 0, ∀i, j = 1 to N
for j = 1 to N
  set I = {1, 2, 3, ..., N}
  for q = 1 to j x_j
    s = argmax_{k∈I} ΔD_k (current (k))
    increment x_j^(s) and current(s) by 1
    if x_j^(s) equals x_j
      remove s from I
    end if
  end for
end for
```

Figure 4.4: M-UEP packetization array source symbol assignment pseudocode.

derived from x_j and $x_j^{(i)}, 1 \leq i \leq N-1$ by applying constraint (4.4). Furthermore, since $x_N^{(i)} = x_N \, \forall i, 1 \leq i \leq N$, it is unnecessary to transmit the values $x_N^{(i)}$ for each substream. Therefore, the side information necessary to completely describe the M-UEP packetization array is $(N-1) \times \lceil \log_2(x_j + 1) \rceil$ bits for all $x_j \neq 0$, $1 \leq j \leq N-1$ plus the UEP side information, as derived in Chapter 3.

Clearly, the amount of M-UEP side information is dependent on the number of non-zero layers ($x_j \neq 0$), which itself depends on the channel conditions and source

rate-distortion behaviour. It is instructive to derive an upper bound for this value:

$$\begin{aligned}
\text{Side Information} &= (N-1)\sum_{j=1}^{N-1}\lceil \log_2(x_j+1)\rceil \qquad (4.16)\\
&\leq (N-1)\sum_{j=1}^{N-1}(\log_2(x_j+1)+1)\\
&\leq (N-1)\log_2\left(\Pi_{j=1}^{N-1}(x_j+1)\right)+(N-1)^2\\
&\leq (N-1)\log_2\left(\sum_{j=1}^{N-1}(x_j+1)/(N-1)\right)^{N-1}+(N-1)^2\\
&\leq (N-1)^2\left(\log_2(L/(N-1)+1)+1\right).\\
&\leq 2N\left(log_2(P)+1\right)
\end{aligned}$$

The third inequality in the above sequence of relations follows from the inequality between the geometric and arithmetic means of $N-1$ positive numbers.

It should be noted that in order for the N packets to be independently decodable, they only require to be aware of the locations of their own source symbols. That is, each packet must be able to distinguish between its redundancy symbols and its source symbols in each layer. However, in order for the decoder to be able to use the RS channel protection to reconstruct layers of lost packets, it must know the positions of source symbols and redundancy symbols in all packets, even those packets which are lost. Therefore, it is necessary that the M-UEP packetization array side information is available at the decoder in order to utilize the RS channel protection to reconstruct and decode lost source symbols.

4.5 M-UEP with Fixed Redundancy Locations

From the above derivation of the upper bound of M-UEP side information (4.17), it is clear that the amount of side information is proportional to the square of the number of packets (N). Thus, as N increases, the amount of side information necessary to support the M-UEP framework can begin to restrict the benefits achieved using M-UEP given a fixed transmission budget. To eliminate the additional side information required by M-UEP over UEP, the locations of redundancy symbols can be pre-defined, such that the encoder does not need to communicate the values $x_j^{(i)}$ to the decoder. This concept is essentially a M-UEP framework with fixed redundancy locations. For our purposes, the framework will be denoted FM-UEP.

The method which determines the redundancy locations must be known at both the encoder and decoder. However, there is no limitation on the selection of the method itself. A method which attempts to equally distribute the source symbols in each layer to each packet is presented in this work. This method is motivated by our discussion of the balanced conditions which achieve the globally optimal M-UEP solution. Given a perfectly balanced multi-stream, each layer of the M-UEP packetization array will have an equal number of source symbols from each sub-stream (i.e. $x_j^{(i)} = j \times x_j/N$ for all $i, 1 \leq i \leq N$).

The sub-optimal RD-ORA solution algorithm can be applied to FM-UEP in the same fashion as with M-UEP. Step 1 of the algorithm remains the same, while the source symbol assignment of step 2 is modified. The pseudo-code for the FM-UEP packetization array source symbol assignment is provided in Figure 4.5. The algorithm determines the values of $x_j^{(i)}$ for $1 \leq i, j \leq N$ given the number of rows in each layer $x_j, 1 \leq j \leq N$. For each layer j, there must be jx_j source symbol assignments.

```
x_j = number of rows in layer j
P_idx = packet location for next source symbol assignment
given x_j for all j, 1 ≤ j ≤ N
set P_idx = 1
for j = 1 to N
  for r = 1 to x_j
    for c = 1 to j
      increment x_j^(P_idx)
      P_idx = P_idx  mod N + 1
    end for
  end for
end for
```

Figure 4.5: FM-UEP packetization array source symbol assignment pseudocode.

These are performed on a row basis (index r), where j assignments are made in each row of layer j. The packet location of the next source symbol is determined by P_{idx} which is a counter that cycles through the packets from 1 to N (hence mod N in the pseudocode). By cycling through the packets, and performing source symbol assignment on a row-by-row basis, ensures that the source symbols are distributed as equally as possible in each layer j. Figure 4.6 illustrates a FM-UEP packetization array resulting from the proposed algorithm for $N = 5$ and $L = 10$ where $x_j = \{2, 3, 0, 4, 1\}$. The grey boxes represent redundancy symbols and the white boxes represent source symbol assignment locations. The corresponding $x_j^{(i)}$ values of the resulting FM-UEP packetization array of Figure 4.6 can be confirmed by applying the algorithm of Figure 4.5 given the x_j values as noted above.

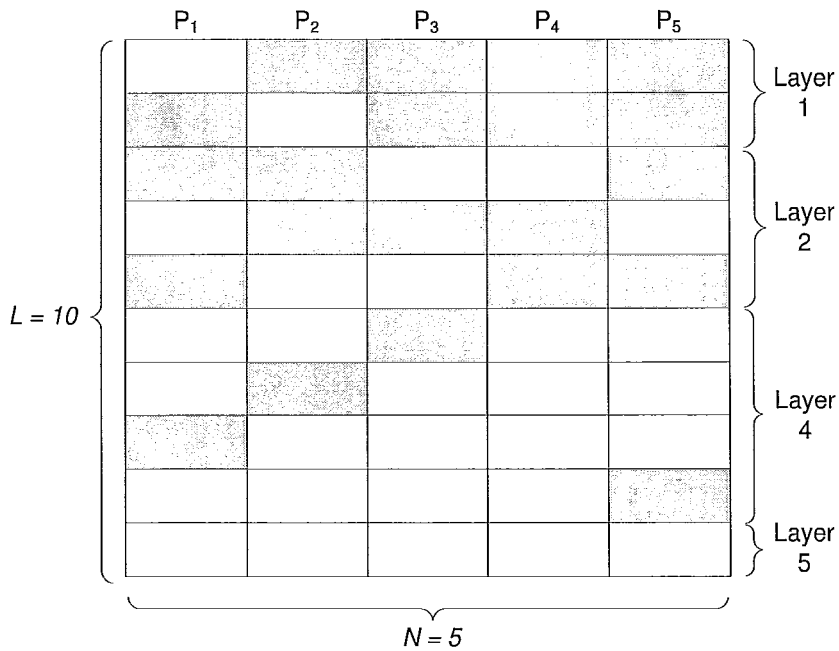

Figure 4.6: FM-UEP packetization array for $N = 5$, $L = 10$ where $x_j = \{2, 3, 0, 4, 1\}$. White boxes represent source symbols and grey boxes represent redundancy symbols.

Chapter 5

Primary Sub-stream Grouping Strategies

When formulating the RD-ORA problem, we assumed that the multi-stream was composed of exactly N independent substreams. However, in most scenarios, the number of substreams in an embedded multi-stream is dependent on the characteristics of the embedded coder. Thus, it is likely that the number of substreams (termed *primary* substreams) will not be equivalent to N. If the multi-stream contains more than N primary substreams, then it can still be transmitted using an N packet M-UEP framework by grouping the primary substreams into N independent groups. Once grouped, the source symbols from each primary substream in the group are interleaved in a R-D optimal manner, to produce a single embedded substream for the group.

5.1 Unconstrained Optimal Primary Sub-stream Grouping

Recall that the sub-optimal solution to the M-UEP RD-ORA problem is globally optimal when the N substreams are perfectly balanced. In such a scenario, any continuous segment of x symbols of the multi-stream contains approximately x/N symbols from each substream. While perfectly balanced substreams will not be attainable from most multi-streams, it is desirable to attain N groups of the P primary substreams which achieve the optimal balance over all possible groupings.

Grouping the P primary substreams into N groups is commonly formulated as the problem of partitioning a set of P elements into N non-empty sets. The number of possible solutions is given by the Stirling number of a second kind S(P,N) (Graham *et al.*, 1994), where S(P,N) is defined as

$$S(P,N) = \frac{1}{N!}\sum_{k=0}^{N}\binom{N}{k}(k)^P(-1)^{N-k} \tag{5.1}$$

We conclude from equation (5.1), an exhaustive search of all possible groupings becomes intractable for all but small P and N. Clearly, the exhaustive search approach to find the globally optimal solution is infeasible.

Additionally, without constraints on grouping the P primary substreams, substantial additional side information must be transmitted to the decoder. The decoder would need to know the group assigned to each primary substream, which would require $P\lceil\log_2(N)\rceil$ bits of side information. However, in order to maintain independently decodable packets, the side information for each primary substream P must be placed as header information in its corresponding packet. In this case, the encoder

would need to communicate the primary substreams belonging to the packet, thus requiring $P\lceil \log_2(P) \rceil$ side information bits. To obtain a tractable optimal grouping algorithm which also requires minimal side information, it is necessary to impose constraints on how the N groups are generated.

5.2 Constrained Optimal Primary Sub-stream Grouping

In order to achieve a successful grouping strategy that is tractable and requires minimal side information, we propose an algorithm that is motivated by the work of Wu *et al.* (2001) on packetization of low bit-rate embedded multimedia bit-streams. The algorithm is tractable by enforcing a fixed-order assignment of the assignment of primary substreams P to the N groups. This constraint imposes a fixed order to the substreams P, and allows only consecutive substreams to be assigned to any of the N groups. The decoder needs only to know the first and last substreams assigned to each group. This technique would require $N\lceil \log_2(P) \rceil$ bits to identify the first primary substream in each group. However, to maintain independently decodable packets, each packet must contain a header of $2\lceil \log_2(P) \rceil$ bits to fully describe the primary substreams belonging to that specific packet (first and last primary substream in group). The objective of the algorithm is to obtain the grouping assignment of the primary substreams P to the N groups, which maximizes the total distortion reduction when prefixes of size L of each substream are decoded. In other words, given the packet length L, where each group is assigned a single packet, the algorithm attempts to group the primary substreams in such a way that as many of the first $N \times L$ source

symbols of the multi-stream exist in the first L symbols of their corresponding group substream. The optimization is subject to the fixed-order constraint.

5.2.1 Algorithm Formulation

The algorithm is formulated using dynamic programming (Wu *et al.*, 2001). The algorithm modifies the low bit-rate algorithm proposed by Wu *et al.* (2001), by restricting primary substreams to be assigned to a single packet. This ensures each packet remains independently decodable. Let $\Delta D(n,p)$ denote the maximum distortion reduction achieved by optimally assigning n packets to the first p primary substreams $s_1, s_2, \ldots, s_p$, $(n \leq p)$. Let $\Delta d(u,v)$ denote the maximum distortion reduction achieved by assigning the primary substreams $s_u, s_{u+1}, \ldots, s_v$ to a single packet, $(u \leq v)$. Using the concept of dynamic programming, computing $\Delta D(n,p)$ can be accomplished by breaking it into smaller problems based on the recurrence relation

$$\Delta D(n,p) = \max_{n-1 \leq i \leq p-1} \Delta D(n-1,i) + \Delta d(i+1,p). \tag{5.2}$$

Equation (5.2) considers the problem of optimally assigning n packets to the first p primary substreams as that of optimally assigning the first i primary substreams to $n-1$ packets, and the remaining $p-i$ substreams to the n^{th} packet. The optimal grouping for $\Delta D(n,p)$ corresponds to the value of i which achieves the maximum distortion reduction over all i, $n-1 \leq i \leq p-1$. Note that the lower bound and upper bound on i ensure that there will be at least one primary substream assigned to each packet.

The base case of the recursive formula of equation (5.2) occurs when $n = 1$ for

$\Delta D(n,p)$. It is clear that in all such cases $\Delta D(1,p) = \Delta d(1,p)$ where $1 \leq p \leq P$. In order for equation (5.2) to be solved recursively, the values $\Delta d(u,v)$ must be computed for all u, v where $1 \leq u \leq v \leq P$. Given two values u and v, $\Delta d(u,v)$ is computed by interleaving the first L source symbols from the primary substreams $s_u, s_{u+1}, \ldots, s_{v-1}, s_v$ in an R-D optimal manner to produce a single embedded substream of length L. $\Delta d(u,v)$ is given to be the total distortion reduction achieved by the embedded substream. Interleaving the source symbols from the $v-u+1$ primary substreams in an R-D optimal fashion and computing $\Delta d(u,v)$ will require $O(L(v-u))$ time, thus the worst case time complexity of computing $\Delta d(u,v)$ will be $O(LP)$. Therefore, $O(LP^3)$ runtime is necessary to compute all values of $\Delta d(u,v)$.

The pseudocode for the proposed algorithm for optimal grouping of P primary substreams with fixed-order assignment to N packets of length L is provided in Figure 5.2.1. The pseudocode obtains the solution for $\Delta D(N,P)$ in a non-recursive fashion by solving the smaller sub-problems first. This ensures that $\Delta D(n-1,i)$ $, n-1 \leq i \leq p-1$ will have already been solved prior to solving for $\Delta D(n,p)$. Since all values required for each $\Delta D(n,p)$ are pre-computed, it only requires $O(p-n)$ time to find the maximum value. In the worst case, this will require $O(P)$ time. Therefore, the overall time complexity for computing $\Delta D(N,P)$ is $O(LP^3 + NP^2)$.

5.2.2 Primary Substream Fixed-Order Assignment

The proposed algorithm is tractable due to the constraint of fixed-order assignment. Additionally, this constraint further reduces the necessary side information to communicate the optimal grouping. For P primary substreams, there are $P!$ possible fixed

compute $\Delta d(u,v)\ \forall\ u,\ v$ where $1 \leq u \leq v \leq P$
initialize $\Delta D(1,i) = \Delta d(1,i)\ ,\ i = 1,2,\ldots,P$
for $n = 2$ to N
  for $p = n$ to P
  $\Delta D(n,p) = \max\limits_{n-1 \leq i \leq p-1} \Delta D(n-1,i) + \Delta d(i+1,p)$
  end for
end for

Figure 5.1: Pseudocode for optimal grouping of P primary substreams with fixed-order assignment to N packets. Solving for $\Delta D(N,P)$

orderings. Therefore, it is not practical to consider all possible orderings before determining the fixed-order assignment which maximizes the distortion reduction of the constrained optimal primary substream grouping algorithm. Furthermore, the optimal fixed-order assignment will be dependent on the R-D characteristics of P primary substreams of the multi-stream. Since each multi-stream will contain substreams with varying R-D characteristics, it is likely that there will not be a single fixed-order assignment which maximizes the distortion reduction of the optimal grouping strategy for all multi-streams.

We consider four possible fixed-orderings strategies: raster scan ordering, zig-zag ordering, dispersed dot-dithering (Kang, 1999) and subband dispersed. For each technique, we will be considering the primary streams represented as $2^m \times 2^m$ matrix where $P = 2^{2m}$. Such a representation is used since it corresponds to the organization of primary substreams of a multi-stream produced by a SPIHT embedded image coder, which was used to evaluate the UEP stratgies in Chapter 6. Figure 5.2 illustrates the lowest resolution (LL_0) subband of a wavelet decomposed image. Each location in this subband corrsponds to a unique primary substream of the multi-stream generated by a SPIHT encoder. Regardless, it is important to note that the presented

techniques can be applied to any type of multi-stream by mapping the resulting fixed-order assignment specified by the $2^m \times 2^m$ matrix to the P primary substreams of the multi-stream.

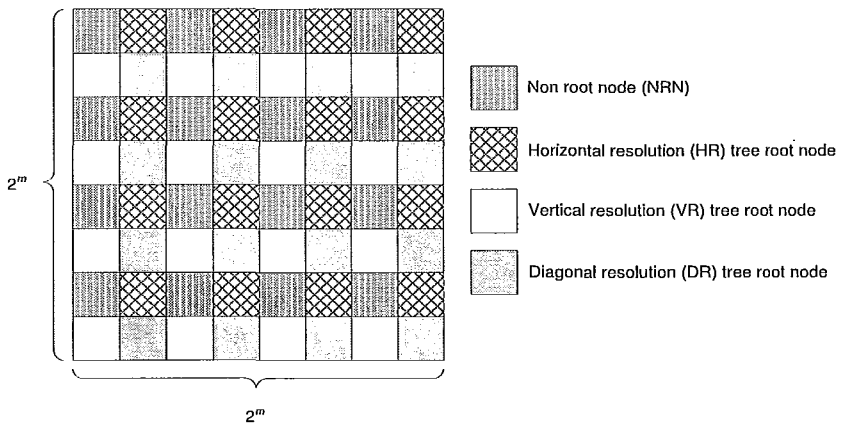

Figure 5.2: Lowest resolution subband (LL_0) of a wavelet decomposed image. LL_0 is composed equally of non-root nodes (NRN), horizontal, vertical, diagonal resolution root nodes as defined by the SPIHT algorithm (Said and Pearlman, 1996).

Raster Scan Order

Raster scan is a commonly used technique which traverses the matrix from left to right and top to bottom. The raster scan fixed-ordering is performed on a block-basis where each block has dimensions 2×2 and is composed of four primary substreams. The block itself is also traversed in raster scan order. Applying a block-based approach ensures that any four consecutive primary substreams in raster scan fixed-ordering contains a primary substream from each type of primary substream: non-root nodes, horizontal resolution trees, vertical resolution trees and diagonal resolution trees. This ensures close to uniform distribution of the substream types within each of the N groups. Figure 5.3 illustrates the block-based raster scan traversal of a $2^{m-1} \times 2^{m-1}$

matrix of blocks as well as the traversal order within each block.

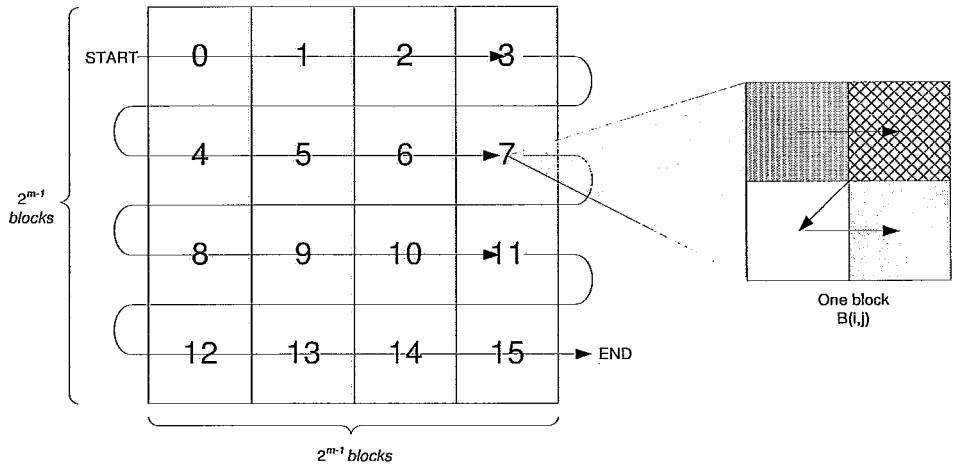

Figure 5.3: Block-based raster scan traversal of a matrix consisting of $2^{m-1} \times 2^{m-1}$ blocks, where m = 3. Each block contains a NRN (top-left), HR tree root node (top-right), VR tree root node (bottom-left), and DR tree root node (bottom-right). The matrix contains a total of $2^3 \times 2^3 = 64$ primary substreams.

The fixed-order assignment matrix using raster scan order can be generated in order to determine the position of each primary substream in the fixed-order assignment. Let $B(i,j)$ represent the code block to which the primary substream located at position (i,j) belongs, where the code blocks are numbered from 0 to $2^{m-1} \times 2^{m-1} - 1$ in raster scan order. The code block number $B(i,j)$ is given as

$$B(i,j) = \lfloor \frac{i}{2} \rfloor \times 2^{m-1} + \lfloor \frac{j}{2} \rfloor, 0 \leq i,j \leq 2^m - 1. \tag{5.3}$$

Let $FO_{RS}(i,j)$ represent the position in the fixed-order assignment of the primary stream located at (i,j). $FO_{RS}(i,j)$ is given as

$$FO_{RS}(i,j) = 4B(i,j) + 2(i \bmod 2) + (j \bmod 2), 0 \leq i,j \leq 2^m - 1. \quad (5.4)$$

Figure 5.4 illustrates a fixed-order assignment matrix using raster scan ordering for $P = 2^3 \times 2^3 = 64$ primary substreams.

$$FO_{RS}(n=3) = \begin{bmatrix} 0 & 1 & 4 & 5 & 8 & 9 & 12 & 13 \\ 2 & 3 & 6 & 7 & 10 & 11 & 14 & 15 \\ 16 & 17 & 20 & 21 & 24 & 25 & 28 & 29 \\ 18 & 19 & 22 & 23 & 26 & 27 & 30 & 31 \\ 32 & 33 & 36 & 37 & 40 & 41 & 44 & 45 \\ 34 & 35 & 38 & 39 & 42 & 43 & 46 & 47 \\ 48 & 49 & 52 & 53 & 56 & 57 & 60 & 61 \\ 50 & 51 & 54 & 55 & 58 & 59 & 62 & 63 \end{bmatrix}$$

Figure 5.4: Block based raster scan fixed-order assignment of $P = 2^3 \times 2^3 = 64$ primary substreams.

Zig-zag Ordering

The zig-zag fixed-order assignment is also a block-based strategy similar to raster scan, which traverses the blocks diagonally in a zig-zag order beginning at the top left block. The equation describing zig-zag traversal is identical to equation (5.4) where in the case of zig-zag traversal $B(i,j)$ would map to the corresponding code block number based on the zig-zag traversal of the matrix as illustrated in Figure 5.5. Zig-zag ordering provides another simple traversal scheme which traverses the code blocks relative to their distance from the top-left of the matrix. This technique also keeps adjacent blocks in the traversal more spatially related than raster-scan

order since adjacent blocks in zig-zag ordering are always neighbouring blocks in the matrix. Figure 5.6 illustrates a fixed-order assignment matrix using zig-zag ordering for $P = 2^3 \times 2^3 = 64$ primary substreams.

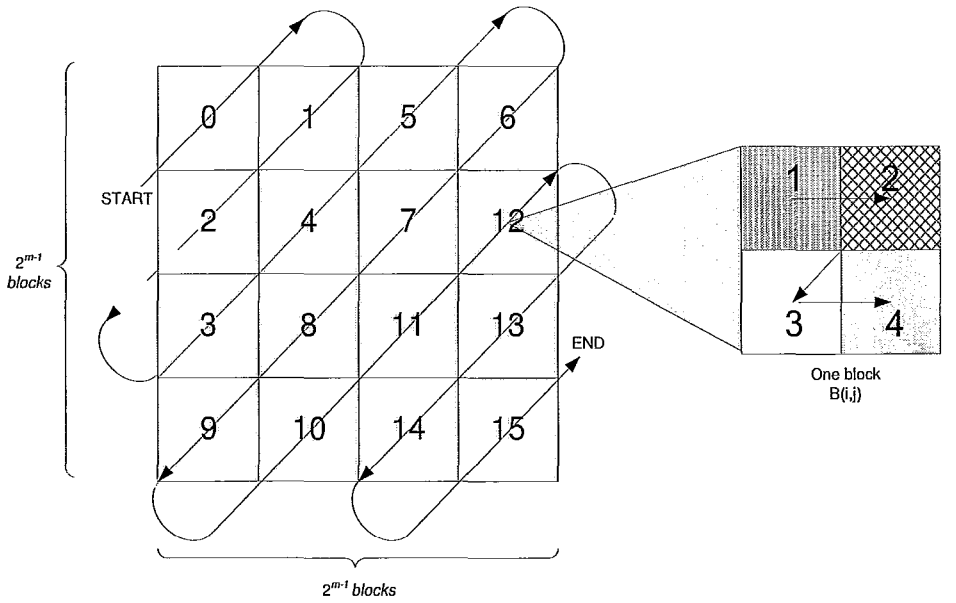

Figure 5.5: Block-based zig-zag scan traversal of a matrix consisting of $2^{m-1} \times 2^{m-1}$ blocks, where m = 3. Each block contains a NRN (top-left), HR tree root node (top-right), VR tree root node (bottom-left), and DR tree root node (bottom-right). The matrix contains a total of $2^3 \times 2^3 = 64$ primary substreams.

Dispersed Dot-dither Ordering

Dispersed dot-dither fixed-order assignment is taken from digital halftoning (Kang, 1999). It was applied in the works of Wu *et al.* (2001) and Rogers and Cosman (1998b) due to its ability to provide spatial dispersion of the primary substreams in each group. This can be beneficial in terms of error concealment if a packet is lost during transmission. It spatially disperses the error regions due to packet loss, which

$$FO_{ZZ}(n=3) = \begin{bmatrix} 0 & 1 & 4 & 5 & 20 & 21 & 24 & 25 \\ 2 & 3 & 6 & 7 & 22 & 23 & 26 & 27 \\ 8 & 9 & 16 & 17 & 28 & 29 & 48 & 49 \\ 10 & 11 & 18 & 19 & 30 & 31 & 50 & 51 \\ 12 & 13 & 32 & 33 & 44 & 45 & 52 & 53 \\ 14 & 15 & 34 & 35 & 46 & 47 & 54 & 55 \\ 36 & 37 & 40 & 41 & 56 & 57 & 60 & 61 \\ 38 & 39 & 42 & 43 & 58 & 59 & 62 & 63 \end{bmatrix}$$

Figure 5.6: Zig-zag fixed-order assignment of $P = 2^3 \times 2^3 = 64$ primary substreams.

then can be approximated by successfully received neighbouring substreams (Wu *et al.*, 2001). Additionally, it helps to disperse spatial regions of primary substreams which contain a large concentration of the R-D information. By doing so, this strategy helps to better balance the number of primary substreams assigned to each group.

The dispersed dot-dither ordering can be obtained using a recursive method which generates dispersed dot-dithering ordering matrices of size $2^m \times 2^m$ for $m = 1, 2, 3, \ldots$. The recursive method begins with a 2×2 matrix (denoted T_2) which satisfies Bayer's criterion for optimal dispersion (Kang, 1999):

$$T_2 = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix}. \tag{5.5}$$

Consistent with the notation of (Kang, 1999), T_k denotes a dispersed dot-dither matrix of size $k \times k$ which is obtained by applying the following recursive relation

$$T_k = \begin{bmatrix} 4T_{k/2} + T_2[0,0]U_{k/2} & 4T_{k/2} + T_2[0,1]U_{k/2} \\ 4T_{k/2} + T_2[1,0]U_{k/2} & 4T_{k/2} + T_2[1,1]U_{k/2} \end{bmatrix} \tag{5.6}$$

where U_k denotes a $k \times k$ matrix with all elements equal to 1. Note that dispersed dot-dither ordering is not a block based scheme, but instead generates the ordering directly for the primary substreams. Figure 5.7 illustrates a fixed-order assignment matrix using dispersed dot-dither ordering for $P = 64$ primary substreams (T_k where $k = 8$).

$$FO_{DD}(k=8) = T_8 = \begin{bmatrix} 0 & 32 & 8 & 40 & 2 & 34 & 10 & 42 \\ 48 & 16 & 56 & 24 & 50 & 18 & 58 & 26 \\ 12 & 44 & 4 & 36 & 14 & 46 & 6 & 38 \\ 60 & 28 & 52 & 20 & 62 & 30 & 54 & 22 \\ 3 & 35 & 11 & 43 & 1 & 33 & 9 & 41 \\ 51 & 19 & 59 & 27 & 49 & 17 & 57 & 25 \\ 15 & 47 & 7 & 39 & 13 & 45 & 5 & 37 \\ 63 & 31 & 55 & 23 & 61 & 29 & 53 & 21 \end{bmatrix}$$

Figure 5.7: Dispersed dot-dither fixed-order assignment of $P = 64$ primary substreams.

Subband Dispersed Ordering

The subband dispersed (SD) fixed-order assignment is inspired by the dispersed dot-dither ordering strategy, but differs in that it equally disperses the primary substreams from each of the four types: NRN, HR, VR and DR. By observing Figure 5.7, note that positions $0 - 15$ are spatially dispersed, but they are all NRN primary substreams. Similarly, positions $16-31$, $32-47$, and $48-64$ correspond to DR, HR, VR primary substreams respectively. The subband dispersed ordering ensures that any four consecutive primary substreams in subband dispersed fixed-ordering will always contain a primary substream from each of the four types: NRN, HR, VR and DR. This technique attempts to create dispersion while also maintaining near uniform

distribution of the substream types within each of the N groups.

To define the subband dispersed fixed-ordering, consider, for $k \geq 1$, the $2^k \times 2^k$-dimensional matrices MA_k, MB_k, MC_k, MD_k defined recursively as follows

$$MA_1 = \begin{bmatrix} 0 & 4 \\ 8 & 12 \end{bmatrix}, \; MB_1 = \begin{bmatrix} 13 & 1 \\ 5 & 9 \end{bmatrix}, \; MC_1 = \begin{bmatrix} 10 & 14 \\ 2 & 6 \end{bmatrix}, \; MD_1 = \begin{bmatrix} 7 & 11 \\ 15 & 3 \end{bmatrix}. \tag{5.7}$$

$$X_{k+1} = \begin{bmatrix} X_k & X_k + 1 \times 2^{2k+2} \times U_k \\ X_k + 2 \times 2^{2k+2} \times U_k & X_k + 3 \times 2^{2k+2} \times U_k \end{bmatrix} \text{ for } X \in \{MA, MB, MC, MD\}, k \leq 1 \tag{5.8}$$

where U_k denotes the $2^k \times 2^k$-dimensional matrix whose all elements are equal to 1. Finally, for $k \geq 2$, define the $2^k \times 2^k$-dimensional matrix M_k as

$$M_k = \begin{bmatrix} MA_{k-1} & MB_{k-1} \\ MC_{k-1} & MD_{k-1} \end{bmatrix} \tag{5.9}$$

If the number of rows and columns in the lowest subband is 2^m, then the matrix M_m defines the index in the SD ordering corresponding to each primary substream. Note that the indexing starts at 0. Figure 5.8 illustrates the assignment matrix for SD ordering, when $m = 3$, hence $P = 64$.

5.3 Fixed Size Primary Sub-stream Grouping

It is not practical to find the globally optimal unconstrained grouping of P primary substreams into N groups for all but small N and P. The constrained optimal

$$FO_{SD}(m=3) = T_{2^3} = \begin{bmatrix} 0 & 4 & 16 & 20 & 13 & 1 & 29 & 17 \\ 8 & 12 & 24 & 28 & 5 & 9 & 21 & 25 \\ 32 & 36 & 48 & 52 & 45 & 33 & 61 & 49 \\ 40 & 44 & 56 & 60 & 37 & 41 & 53 & 57 \\ 10 & 14 & 26 & 30 & 7 & 11 & 23 & 27 \\ 2 & 6 & 18 & 22 & 15 & 3 & 31 & 19 \\ 42 & 46 & 58 & 62 & 39 & 43 & 55 & 59 \\ 34 & 38 & 50 & 54 & 47 & 35 & 63 & 51 \end{bmatrix}$$

Figure 5.8: Subband dispersed fixed-order assignment for $P = 64$ primary substreams ($m = 3$).

primary substream grouping strategy provides a tractable solution which maximizes the distortion reduction achieved by the groupings under the constraint of a fixed-order assignment. This scheme was shown to have a time complexity of $O(LP^3 + NP^2)$, as well as requiring side information information to communicate the optimal grouping to the decoder. By applying a fixed primary substream grouping strategy, the optimization complexity can be further reduced and the side information required to specify the grouping can be eliminated completely.

The fixed-size primary substream grouping strategy essentially divides the primary substreams into equally sized groups. Any of the proposed fixed-order assignments (raster scan, zig-zag scan, dispersed dot-dither, and subband dispersed) can be applied for fixed primary substream grouping. The selected fixed-order assignment must be known by both the encoder and decoder in order to perform successful encoding and decoding of the multi-stream.

After selecting a fixed-order assignment, generating the N groups from the P

primary substreams is a trivial process. The first $R = P \mod N$ groups are assigned $\lceil P/N \rceil$ consecutive primary substreams, then the remaining groups are assigned $\lfloor P/N \rfloor$ consecutive primary substreams based on the fixed-order assignment. Figure 5.9 illustrates the creation of four groups ($N = 4$) of primary substreams based on the raster (a) and zig-zag (b) fixed-ordering assignment techniques. Figure 5.9 illustrates the groupings at the block level. The primary substreams within each block are traversed in raster-scan order as discussed previously. Note that for some values of N, the primary substreams of a single block may belong to different groups.

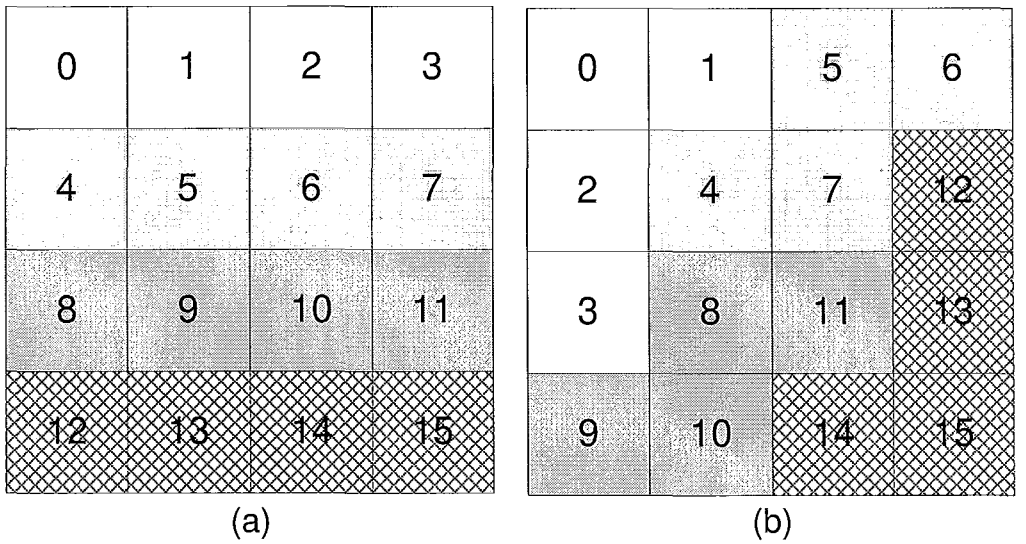

Figure 5.9: Fixed primary substream grouping for four groups ($N = 4$). Raster-scan fixed-order assignment (a). Zig-zag fixed-order assignment (b). The four groups are distinguished by the grayscale shading of the blocks.

5.4 Grouping Overhead

It is essential to consider the necessary side information for each primary substream grouping strategy. Though the side information has been discussed specifically in each section, the necessary side information for each grouping strategy is presented formally in Table 5.1 below. The minimum side information as well as the side information necessary to ensure independently decodable packets is presented. Note that the minimum side information is smaller, but would require that transmission occurs via a secure channel to ensure that it has been correctly received at the decoder prior to transmitting the individual packets.

Table 5.1: Summary of amount of side information (*in bits*) necessary for primary substream grouping strategies

Grouping Method	Minimal Overhead	Independent Packet Overhead
Unconstrained	$P\lceil\log_2(N)\rceil$	$P\lceil\log_2(P)\rceil$
Fixed order assignment	$N\lceil\log_2(P)\rceil$	$2N\lceil\log_2(P)\rceil$
Fixed size	0	0

For the unconstrained grouping scenario, the minimal side information would entail the group number for each primary stream ($P\lceil\log_2(N)\rceil$ *bits*). For independently decodable packets, each packet (group) would need to know each primary substream which belongs to that group ($P\lceil\log_2(P)\rceil$ *bits*).

For the fixed-order assignment grouping strategy, the minimal side information would contain an identfier for the first primary substream in each group ($N\lceil\log_2(P)\rceil$ *bits*). For independently decodable packets, each packet (group) would need to identify the first and last primary substream belonging to the group ($2N\lceil\log_2(P)\rceil$ *bits*).

Since the fixed-size grouping strategy requires small computational load, the constrained optimal primary substream grouping assignment strategy must provide sufficient additional distortion reduction performance in order to justify the additional complexity of implementing the optimization scheme. This trade-off will be evaluated in the experimental results to follow.

Chapter 6

Experimental Results

The main goal of our experiments was to compare the performance of the proposed M-UEP strategies versus UEP. To accomplish this, tests were performed on two 512×512 images (Lena and Peppers). In both cases a 5-level Cohen-Daubechies-Feauveau 9/7 wavelet transform (Cohen *et al.*, 1992) was applied, and the resulting wavelet coefficient matrix was SPIHT encoded (without arithmetic coding). In such circumstances, the multi-stream produced by SPIHT can be regarded as the interleaving of 256 independent *primary* substreams, with 64 substreams corresponding to non-root nodes in the lowest subband, and the remaining 192 streams representing spatial orientation trees. The experiments considered varying numbers of packets $N = 2, 4, \ldots, 38, 40$ at two transmission rates $R = 0.20, 0.50$ bpp. For all experiments, a source symbol was considered to be 8 bits (1 byte) of data. In order to obtain N independent streams, we group the 256 primary substreams into N groups by applying the fixed-order assignment strategies discussed in Chapter 5: dispersed dot dithered (DD), subband dispersed (SD), raster scan (RS) and zig-zag scan (ZZ). The fixed-order assignment strategies are applied to both the constrained optimal primary substream grouping

and fixed-size primary substream grouping scenarios. The source symbols from the primary substreams assigned to each of the N groups are then interleaved to form a new substream, such that the order of symbols in each substream is the same as their order in the SPIHT coded multi-stream.

The M-UEP and FM-UEP expected PSNR was evaluated for all possible grouping scenarios. In each case we used the sub-optimal MUEP RD-ORA algorithm presented in Chapter 4. To solve Step 1, the globally optimal algorithm for UEP of Dumitrescu *et al.* (2004) was used. The same algorithm was used to obtain the UEP RD-ORA solution.

Tests were performed by assuming an independent packet loss (IPL) channel model with erasure rates $\epsilon = 0.05, 0.15$. Independence of packet losses is a reasonable assumption when the packets are interleaved with packets from other applications during the transmission.

The reported experimental results are divided into three sections. Section 6.1 provides a performance comparison of the various grouping strategies for M-UEP and FM-UEP. All grouping experiments take the grouping side information into consideration by appending it to the beginning of each substream. Section 6.2 reports the M-UEP side information experimental data, and its impact on the overall M-UEP performance is evaluated. The chapter concludes with a presentation of M-UEP performance under channel mismatch conditions in Section 6.3.

6.1 Grouping Performance Evaluation

Recall that a perfectly balanced multi-stream ensures the globally optimal solution to the M-UEP RD-ORA problem is obtained using the proposed sub-optimal algorithm.

Perfectly balanced multi-streams were discussed in detail in Chapter 4. Since the characteristics of a SPIHT multi-stream are dependent on the encoded image, it is difficult to achieve perfectly balanced multi-streams in practice. However, it is expected that the closer a multi-stream is to being perfectly balanced, the closer the solution of the sub-optimal M-UEP RD-ORA algorithm becomes to being globally optimal. In other words, the sub-optimal M-UEP algorithm should perform better for multi-streams close to being perfectly balanced than for unbalanced multi-streams.

Recall that two types of grouping strategies were proposed in Chapter 5: constrained optimal primary substream grouping (Opt) and fixed-size primary substream grouping (FS). For each grouping strategy, any of the proposed fixed-order grouping assignment techniques can be applied. The ordering of the P primary substreams directly impacts the ability to balance the N groups.

To gain a better understanding of the impact of the primary substream ordering on the performance of the various grouping strategies, it is instructive to observe the source symbol distribution of the SPIHT multi-stream among its primary substreams, and the resulting source symbol distribution after applying the proposed grouping strategies. Figure 6.1 illustrates the number of source symbols in each primary substream of the SPIHT multi-stream generated by encoding the Peppers image at $R_s = 0.1367$ bpp. The value of R_s is the optimal M-UEP source symbol budget for a total transmission rate $R = 0.20$ bpp with channel erasure rate $\epsilon = 0.15$. The value of R_s is obtained after completion of Step 1 of the sub-optimal M-UEP RD-ORA algorithm, and thus is not dependent on the applied grouping strategy. In the case of Figure 6.1, the primary substream ordering is $[1, 64]$ = Non root nodes, $[65, 128]$ =

Horizontal resolution trees, $[129-192]$ = Vertical resolution trees, $[193-256]$ = Diagonal resolution trees. Note that the primary substreams corresponding to non root

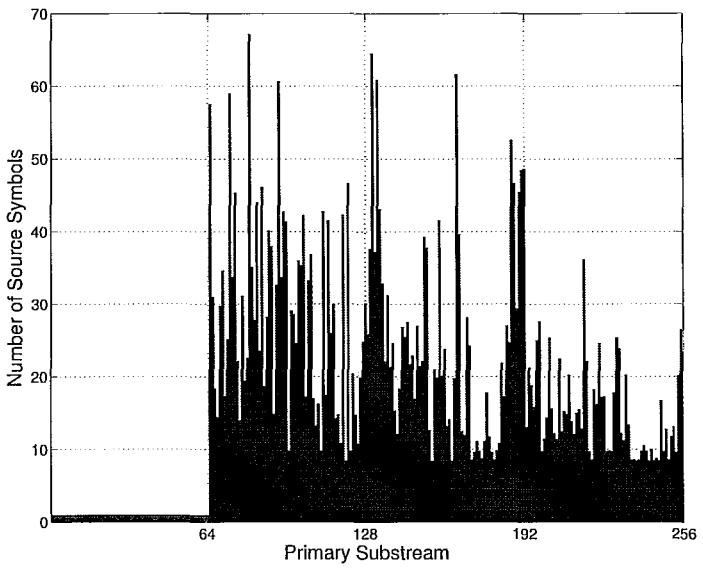

Figure 6.1: Number of source symbols in each primary substream of the Peppers SPIHT multi-stream at $R_s = 0.1367$ bpp.

nodes contain significantly less source symbols than those corresponding to spatial orientation trees. This is due to the fact that these substreams describe the encoding of only a single wavelet coefficient. Additionally, note that there is a significant variation between the number of source symbols belonging to the various primary substreams of the multi-stream.

Figures 6.2 and 6.3 illustrate the number of source symbols in each group for $N = 10$ and $R_s = 0.1367$ when applying zig-zag ordering (ZZ) with the constrained optimal primary substream grouping (Opt-ZZ M-UEP) and fixed-size primary substream grouping (FS-ZZ M-UEP) respectively. The y-axis label at position 448 represents the number of source symbols per group if the groups were equally sized (symbol

balanced). It is clear by observation that Opt-ZZ M-UEP is much closer to achieving equally sized groups than FS-ZZ M-UEP.

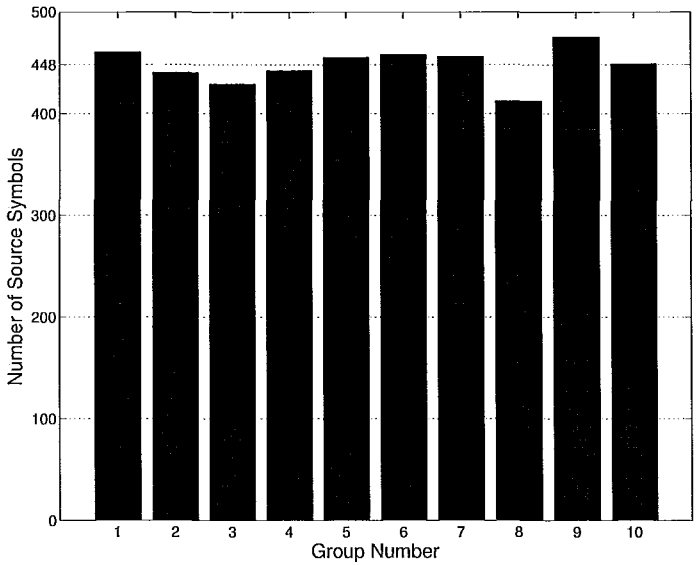

Figure 6.2: Number of source symbols in each group for Opt-ZZ M-UEP when $N = 10$ and $R_s = 0.1367$.

Figures 6.4 and 6.5 illustrate the source symbol distribution when applying dispersed dot-dither ordering (DD) for the case of Opt-DD M-UEP and FS-DD M-UEP respectively. Note that Opt-DD M-UEP is not as close to producing equally sized groups as Opt-ZZ M-UEP, and FS-DD M-UEP produced significantly unbalanced groups. These figures also illustrate the effectiveness of the constrained optimal primary substream grouping algorithm, as it is able to significantly improve the balance characteristics of the groups with respect to fixed-size primary substream grouping.

Grouping performance evaluations reveal that the grouping strategy closest to achieving equally sized groups produced the highest expected PSNR over the full

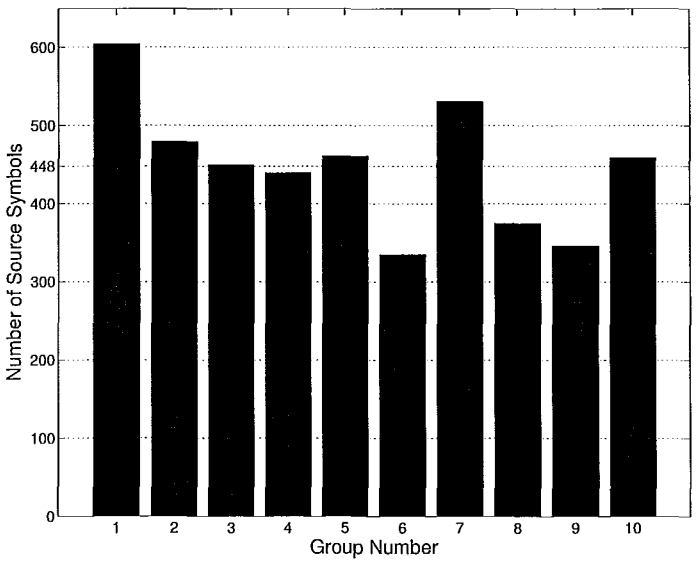

Figure 6.3: Number of source symbols in each group for FS-ZZ M-UEP when $N = 10$ and $R_s = 0.1367$.

range of packets $N = 2, 4, \ldots, 38, 40$. In the case of constrained optimal primary sub-stream grouping, Opt-ZZ M-UEP yielded the highest PSNR values for both images at all transmission rates (R) and erasure rates (ϵ). The other three fixed-order assignment strategies (RS,SD,DD) performed equally well, but less than Opt-ZZ, with peak differences greater than 0.1 dB. Note that the behaviour coincides with the observations from the source symbol distributions. Since Opt-ZZ is more balanced than Opt-DD, we expect it to achieve higher performance from the sub-optimal M-UEP RD-ORA algorithm. All figures include both the UEP performance, and the M-UEP upper bound. The M-UEP upper bound is the expected PSNR computed by Step 1 of the sub-optimal M-UEP RD-ORA solution algorithm. This PSNR upper bound is achieved if all constraints of (4.5) are met by the solution of Step 1. Figure 6.6 illustrates the PSNR for Opt-(DD,SD,RS,ZZ) M-UEP for the Peppers image at

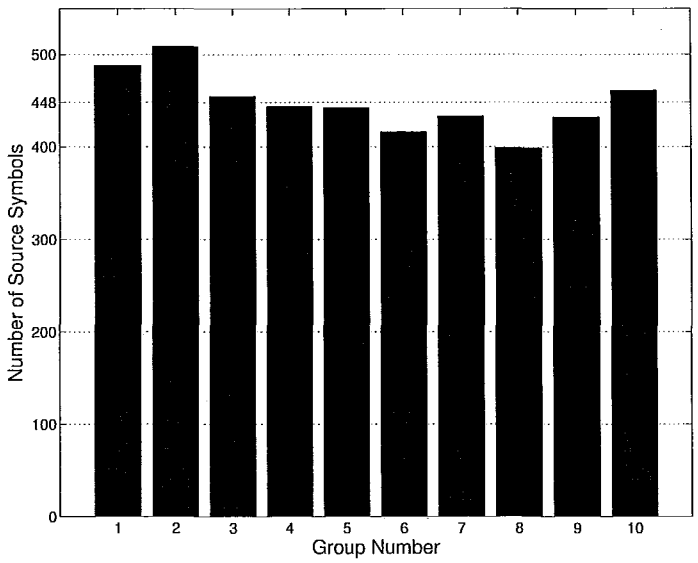

Figure 6.4: Number of source symbols in each group for Opt-DD M-UEP when $N = 10$ and $R_s = 0.1367$.

$R = 0.20$, $\epsilon = 0.15$. Note that Opt-ZZ M-UEP achieves near optimal performance (close to M-UEP upper bound), while Opt-(RS, SD, DD) achieve similar performance, approximately 0.1 dB less than Opt-ZZ M-UEP. All four grouping strategies outperform UEP with peak improvement exceeding 0.6 dB for all strategies when $N = 6$. Therefore, while all grouping strategies achieve improvement over UEP, additional PSNR improvement can be achieved by selecting Opt-ZZ M-UEP.

In the case of fixed-size primary substream grouping, no optimization of the groups is performed, and thus no side information must be appended to the substreams. FS-SD M-UEP achieves the highest expected PSNR for both images at all transmission rates (R) and erasure rates (ϵ). Figure 6.7 presents the PSNR for FS-(DD,SD,RS,ZZ) M-UEP for the Peppers image at $R = 0.20$, $\epsilon = 0.15$. All grouping strategies except FS-DD M-UEP outperform UEP for all N with peak improvements near 0.6 dB for

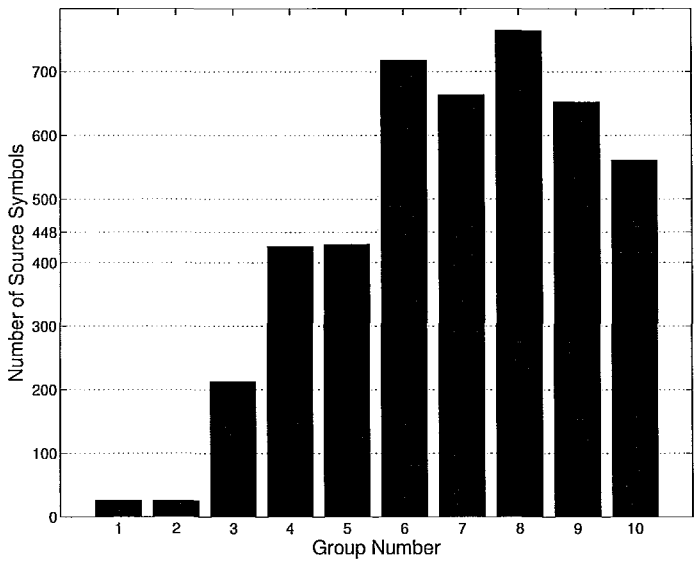

Figure 6.5: Number of source symbols in each group for RS-DD M-UEP when $N = 10$ and $R_s = 0.1367$.

$N = 4$. The poor performance of FS-DD can be attributed to its severely unbalanced substreams (as shown in Figure 6.5). This result shows the benefit of the optimal primary substream grouping strategy. Even though FS-DD M-UEP under performs UEP, by applying the optimal substream grouping algorithm, Opt-DD M-UEP is able to outperform UEP for all N.

For FM-UEP, Opt-SD and FS-SD M-UEP yielded the best PSNR for optimal and fixed-size primary substream grouping. In all our experiments, the peak improvements of SD FM-UEP over (DD, RS, ZZ) FM-UEP did not exceed 0.1 dB. Figure 6.8 presents the results for the Peppers image where $R = 0.20$ and $\epsilon = 0.15$ for constrained optimal primary substream grouping. In comparison to Figure 6.6, it is clear that FM-UEP does not achieve as much PSNR improvement over UEP as M-UEP when the side information is not considered. However, FM-UEP obtains peak PSNR improvement

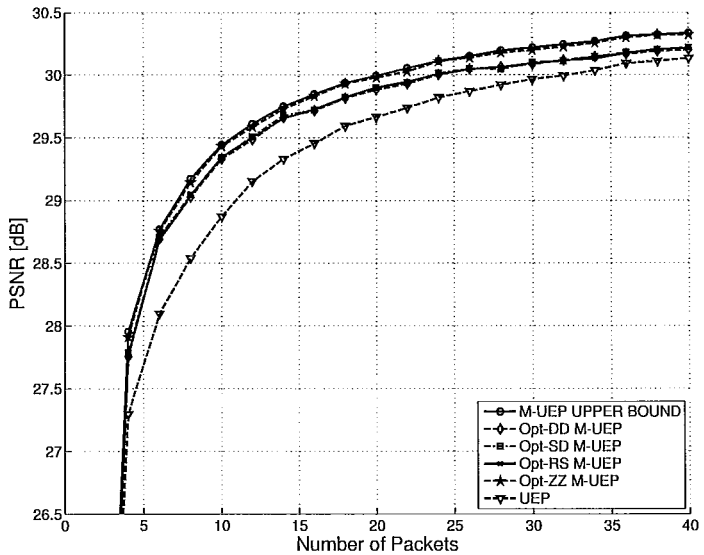

Figure 6.6: M-UEP PSNR vs. Number of Packets for peppers image at $R = 0.20$ and $\epsilon = 0.15$ when applying constrained optimal primary substream grouping.

of at least 0.4 dB over UEP. All grouping strategies excluding Opt-DD FM-UEP yield improvement over UEP for $N < 30$. Figure 6.9 illustrates the results for the Peppers image where $R = 0.20$ and $\epsilon = 0.15$ for fixed-size primary substream grouping. FS FM-UEP produces peak PSNR improvements exceeding 0.5 dB over UEP for all grouping strategies except FS-DD FM-UEP. For large N, FS FM-UEP under performs UEP for all groupings. As with FS-DD M-UEP, the poor performance of FS-DD FM-UEP can be attributed to the unbalanced substreams produced by FS-DD grouping.

From our experimental results, it is evident that significant performance improvements (upwards of 0.4 dB) can be obtained over UEP in all proposed frameworks: Opt M-UEP, FS M-UEP, Opt FM-UEP, FS FM-UEP.

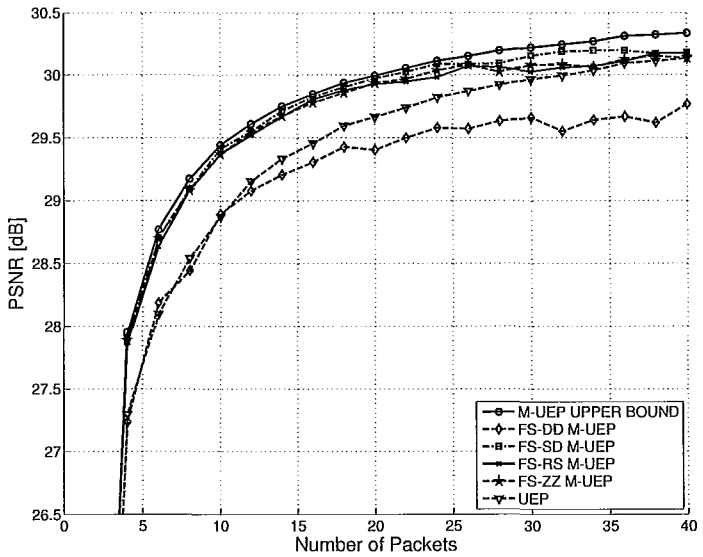

Figure 6.7: M-UEP PSNR vs. Number of Packets for Peppers image at $R = 0.20$ and $\epsilon = 0.15$ when applying fixed-size primary substream grouping.

6.2 M-UEP Overhead

Recall that the performance advantage of M-UEP over UEP comes at the cost of additional side information needed at the decoder. In Figure 6.10 we plot the ratio between the amount of M-UEP side information and R versus N, for Lena and Peppers images at $R = 0.50$, $\epsilon = 0.15$. Note that the M-UEP side information is calculated at the completion of Step 1 of the sub-optimal M-UEP solution algorithm, and thus is not dependent on the applied grouping strategy. Also, the M-UEP side information does not contain the grouping information. Figure 6.10 also includes a plot corresponding to the M-UEP side information upper bound (see equation (4.17)), and the plot corresponding to the UEP side information. As it can be observed for both images, the side information increases roughly linearly in the number of packets. Also, the difference between M-UEP side information and UEP side information

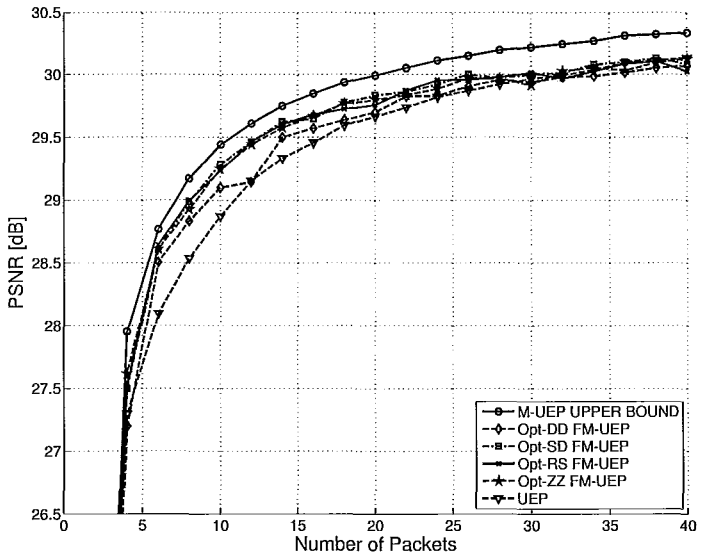

Figure 6.8: FM-UEP PSNR vs. Number of Packets for Peppers image at $R = 0.20$ and $\epsilon = 0.15$ when applying constrained optimal primary substream grouping.

increases in N. However, the rate of increase is lower than predicted by the upper bound. Also, note that the difference in side information between images is negligible.

It is clear from Figure 6.10 that the M-UEP side information is larger than the UEP side information. Next we analyze the impact of transmitting the side information within the M-UEP packets, and thus including it in the total bit budget. Since the side information is critical for decoding the remainder of the packetization array, it must be appended to the front of the multi-stream. To incorporate the side information in the total bit budget we do the following. First we apply the sub-optimal algorithm of Chapter 4 disregarding the side information, and populate the packetization array with source symbols from the substreams. Then we increase the size of the first non-zero layer j_{nz} by adding to it $\lceil S/j_{nz} \rceil$ more rows, where S is the amount of M-UEP side information (in symbols). The source symbols placed in these rows

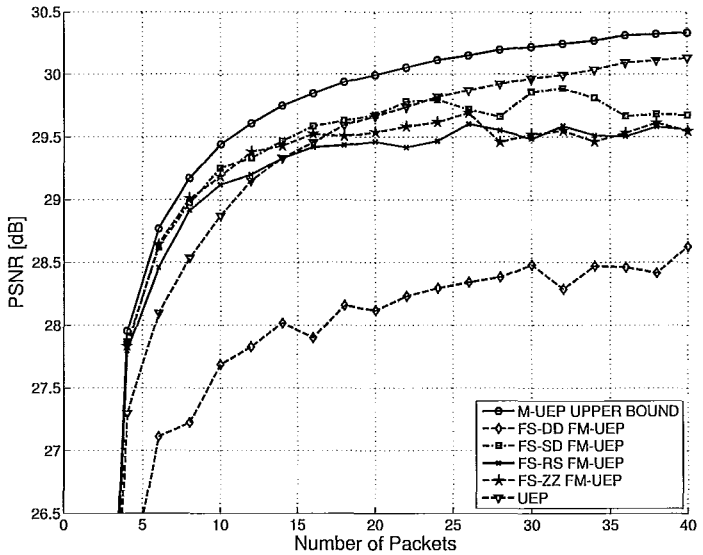

Figure 6.9: FM-UEP PSNR vs. Number of Packets for Peppers image at $R = 0.20$ and $\epsilon = 0.15$ when applying fixed-size primary substream grouping.

constitute the M-UEP side information. To keep the size of each packet equal to L then we decrement the size of the last layer by removing from it $\lceil S/j_{nz} \rceil$ rows.

Figures 6.11 and 6.12 plot the PSNR difference of M-UEP vs. UEP when transmitting the M-UEP side information within the M-UEP packetization array for Opt-ZZ M-UEP at $R = 0.50$ bpp and $\epsilon = 0.15$. Both Figures also include the upper bound on the M-UEP PSNR improvement over UEP, as well as the PSNR improvement of M-UEP without incorporating the side information into the total bit budget. Opt-ZZ M-UEP with side information achieves peak improvements over UEP of 0.44 dB and 0.32 dB for the Lena and Peppers images respectively. Opt-ZZ M-UEP with side information out performs UEP for all N, even though the side information increases in N.

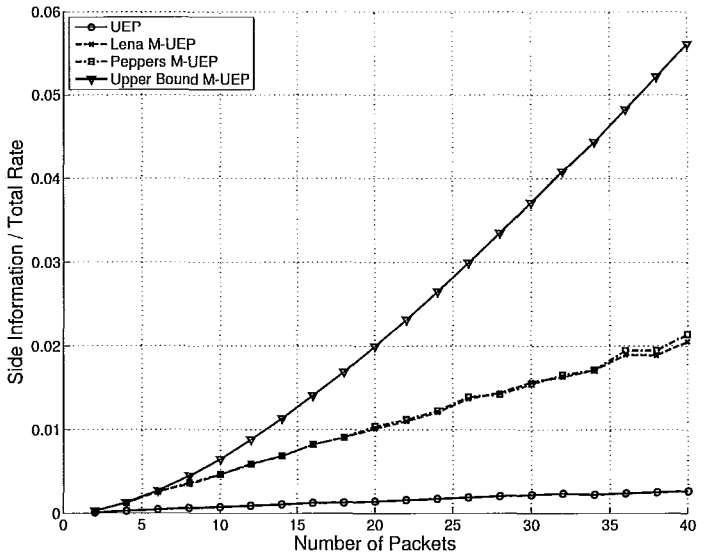

Figure 6.10: Ratio between M-UEP side information and total bit budget vs. number of packets N, for Lena and Peppers, at $R = 0.50$ bpp.

6.3 M-UEP under channel mismatch

A channel mismatch refers to the scenario when the actual packet erasure rate of the channel varies from the predicted value ϵ. Figure 6.13 plots the PSNR difference between Opt-SD M-UEP and UEP for each image under channel mismatch conditions. The actual channel loss rate varies in the range $[0.01 - 0.30]$, while the redundancy allocation is optimized for the loss rate $\epsilon = 0.15$. The number of packets is $N = 16$ and $R = 0.50$ bpp. It is clear that OptSD M-UEP maintains its superiority over UEP in all cases. Another interesting observation is that the improvement in PSNR increases as the erasure rate tends away from the predicted value. This increase in performance as the loss rate becomes smaller can be attributed to the fact that the redundancy allocation algorithm for M-UEP assigns less total redundancy than UEP. Consequently, M-UEP has higher performance than UEP when the number of

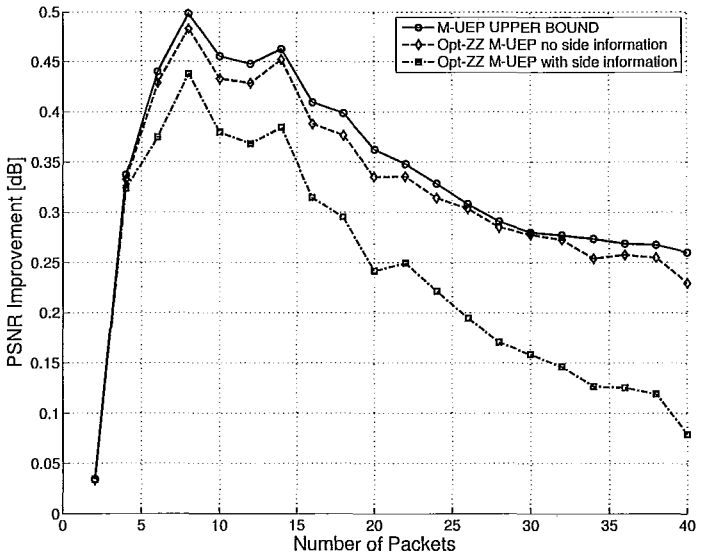

Figure 6.11: M-UEP PSNR difference vs. UEP when transmitting M-UEP side information within the M-UEP packetization array for the Lena image at $R = 0.50$ bpp and $\epsilon = 0.15$.

received packets is very high. When the loss rate exceeds the predicted value, M-UEP is able to decode any received packet, while UEP is not able to begin decoding until at least j_{nz} packets are recieved. Therefore, not only does M-UEP achieve higher PSNR performance at the optimized erasure rate (ϵ), but it is also a more robust framework under channel mismatch conditions.

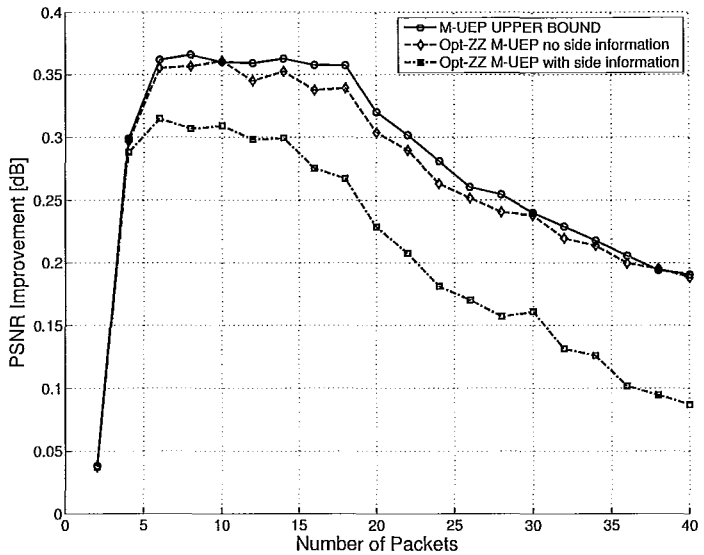

Figure 6.12: M-UEP PSNR difference vs. UEP when transmitting M-UEP side information within the M-UEP packetization array for the Peppers image at $R = 0.50$ bpp and $\epsilon = 0.15$.

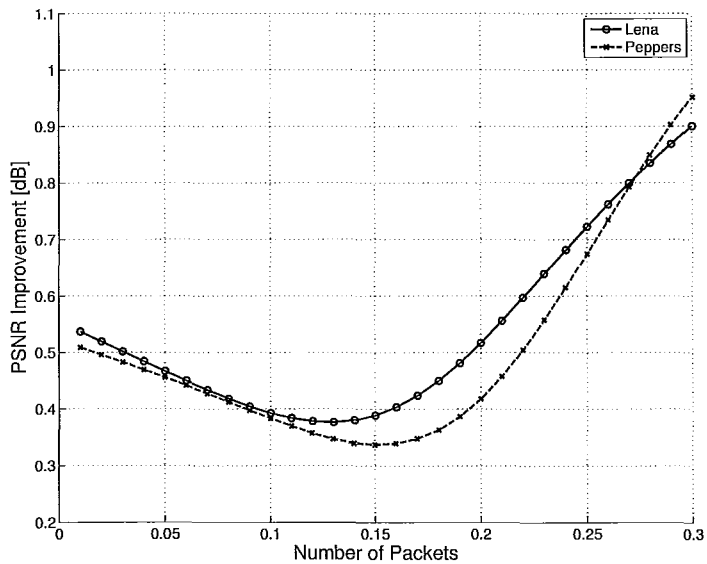

Figure 6.13: PSNR improvement of Opt-SD M-UEP over UEP vs. packet erasure rate ϵ for Lena and Peppers images at transmission rate $R = 0.50$ bpp and $N = 16$. The erasure protection is optimized for erasure rate $\epsilon = 0.15$.

Chapter 7

Conclusions

This thesis proposed a novel unequal erasure protection (UEP) strategy for the transmission of multi-streams over packet erasure networks: multi-stream unequal erasure protection (M-UEP). M-UEP creates independently decodable packets to ensure that all received source symbols are decoded and uses permuted Reed-Solomon codes to increase the flexibility of the redundancy assignment. The R-D optimal redundancy allocation (RD-ORA) problem was formulated and shown in the general case to have a complexity of $O(2^N N(L+1)^{N+1})$ time, and under the convexity assumption $O(2^N(L+1)^{N+1})$ time, where N is the number of transmitted packets and L is the packet size. To address the high complexity of the globally optimal solution an efficient sub-optimal algorithm running in $O(N^2L^2)$ time was proposed. The additional side information necessary for M-UEP at the decoder was discussed and an upper bound on the side information length was derived. Moreover, a technique for mitigating the side information (FM-UEP) was presented. Experiments performed on SPIHT coded images (with appropriate grouping of wavelet coefficient) validated the superiority of M-UEP and FM-UEP over UEP, with peak improvements of 0.6

and 0.5 dB, respectively. Additionally, our tests revealed that M-UEP is more robust than UEP in adverse, unpredictable and varying channel conditions. Future research interests include improving the sub-optimal solution to M-UEP RD-ORA problem and applying M-UEP to JPEG2000 coded images. Initial upper-bound experiments with JPEG2000 reveal the promise of similar performance improvement of M-UEP over UEP as achieved with the SPIHT coder.

Appendix A

Sufficiency of M-UEP RD-ORA Constraints (4.3), (4.4), (4.5)

The M-UEP RD-ORA problem is formulated as

$$\bar{D}_M = D_{max} - \sum_{j=1}^{N} \left(C_M(j) \sum_{i=1}^{N} \left(\sum_{r_i=1+\sum_{\ell=1}^{j-1} x_\ell^{(i)}}^{\sum_{\ell=1}^{j} x_\ell^{(i)}} \Delta D_i(r_i) \right) \right). \tag{A.1}$$

Thus, to solve the M-UEP RD-ORA, $\bar{D}_M$ must be minimized over all non-negative integers $x_j, x_j^{(i)}$, $1 \leq i, j \leq N$, subject to the constraints

$$\sum_{j=1}^{N} x_j = L, \tag{A.2}$$

$$\sum_{i=1}^{N} x_j^{(i)} = j x_j \ \forall j,\ 1 \leq j \leq N, \tag{A.3}$$

$$x_j^{(i)} \leq x_j \ \forall i, j,\ 1 \leq i, j \leq N. \tag{A.4}$$

In order to show that the constraints (A.2)-(A.4) are sufficient, we need to prove that for any non-negative integers x_j, $x_j^{(i)}$ satisfying (A.2-A.4), there is a corresponding arrangement of the source symbols in the packetization array compatible with the M-UEP framework.

Proof We begin by fixing some j and considering populating the j-th layer of the packetization array with source symbols. Consider N groups, where the i-th group contains $x_j^{(i)}$ symbols from the i-th substream. Populating the j^{th} layer is equivalent to placing these symbols in an array of x_j rows and N columns such that all the symbols of the i-th group are situated in the i-th column, and each row contains exactly j source symbols. The unoccupied locations in the array are reserved for the redundancy symbols. The pseudocode for the symbol assignment algorithm is provided in Figure A.1.

for $k = 1$ to x_j
  sort $x_j^{(i)}\ \forall i\ (1 \leq i \leq N)$ in descending order
  assign a symbol from each of the first j substreams to row k
  decrement by 1 $x_j^{(i)} \forall\, i$ with symbols in row k
end for

Figure A.1: Source symbol assignment pseudocode for layer j.

The algorithm begins with the first row of layer j. At each iteration, the values $x_j^{(i)}$ are sorted in descending order. Source symbols are assigned to their respective columns in the current row for the first j substreams in the sorted list. The corresponding $x_j^{(i)}$ value is decremented by 1 for each substream with an assigned source symbol in the current row. The algorithm repeats this process for all x_j rows of layer j.

An example of the source symbol assignment algorithm is illustrated in Figure A.2

where $j = 4, N = 6,$ and $x_j = 3$. In the first iteration, source symbols are assigned to the first row of layer 4 from substreams $1, 4, 5, 6$ since they have the four largest $x_j^{(i)}$ values. Each of these substreams now has one less source symbol to assign in layer 4. The same procedure is followed for iterations 2 and 3 Note that the procedure requires x_j iterations, and all source symbols are assigned to a location in the packetization array after the completion of the x_j-th iteration.

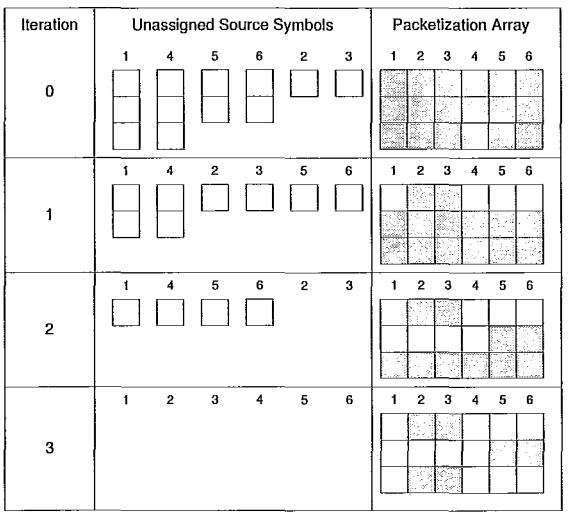

Figure A.2: Example of source symbol assignment algorithm for layer $j = 4$ where $N = 6$ and $x_j = 3$. White boxes represent source symbols, and grey boxes represent redundancy symbols

To prove that our algorithm generates a packetization array compatible with the M-UEP framework, we first show that after the ℓ-th iteration, for $0 \leq \ell \leq x_j$, the value of $x_j^{(i)}$ for any $i\,(1 \leq i \leq N)$ is at most $x_j - \ell$, and a total of $j(x_j - \ell)$ source symbols must still be assigned in layer j (i.e. $\sum_{i=1}^{N} x_j^{(i)} = j(x_j - \ell)$ after the ℓ-th iteration). This can be proved easily by induction. It is true for $\ell = 0$ (i.e., at the beginning of the algorithm) by Conditions (A.3) and (A.4). Assume now that the statement is true for

some $\ell, 0 \leq \ell < x_j$, and let us prove it for $\ell+1$. At most j of the segments may contain the maximum number of $x_j - \ell$ symbols. Assuming otherwise we obtain the total length of the segments to be at least $(j+1)(x_j - \ell)$, which contradicts the inductive hypothesis. Moreover, there are at least j non-empty segments. Again, assuming otherwise we obtain the total length of all segments to be at most $(j-1)(x_j - \ell)$, which again contradicts the inductive hypothesis. Consequently, we can proceed to the next iteration. We identify the j largest $x_j^{(i)}$ values and assign a symbol from each of the j selected substreams to the corresponding column in the $(\ell+1)$-th row. The number of source symbols in each of these segments decreases by one, hence it becomes at most $x_j - \ell - 1$. Each of the segments which have not been used in this iteration contains at most $x_j - \ell - 1$ symbols too. The total number of symbols remaining to be assigned in layer j is clearly j less than the previous iteration, hence it is $j(x_j - \ell - 1)$. Consequently, the statement is true for $\ell + 1$. Therefore, using the above reasoning, it is clear that the algorithm performs exactly x_j iterations, and after the x_j-th iteration $x_j^{(i)} = 0 \forall i \, (1 \leq i \leq N)$, thus all source symbols in layer j have been assigned to a valid location in the packetization array.

Appendix B

Unconstrained Maximum-Weigth Path Problem Formulation for Convex R-D Curves

The following lemma implies that when the R-D curves of all substreams are convex, the optimization problem reduces to finding the maximum-weight path in G, without constraint (4.11).

Lemma 1 *If the the R-D curves of all substreams are convex, then for any path P, there is some path P' satisfying constraint (4.11) such that $w(P') \geq w(P)$.*

Proof Consider a path P consisting of the following nodes $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_L$, where $\mathbf{v}_k = (v_k^{(1)}, v_k^{(2)}, \ldots, v_k^{(N)}, k)$, $0 \leq k \leq L$. Assume that condition (4.11) is violated for some ℓ. Construct the new path P' by replacing the node $\mathbf{v}_\ell$ with a new node $\mathbf{u}_\ell = (u_\ell^{(1)}, u_\ell^{(2)}, \ldots, u_\ell^{(N)}, \ell)$, where

$$u_\ell^{(i)} = v_{\ell-1}^{(i)} + v_{\ell+1}^{(i)} - v_\ell^{(i)} \, \forall i, 1 \leq i \leq N.$$

First, note that this node replacement alters only the ℓ-th and $(\ell+1)$-th edges in the path P. These are replaced, respectively, by the new edges $(\mathbf{v}_{\ell-1}, \mathbf{u}_\ell)$, and $(\mathbf{u}_\ell, \mathbf{v}_{\ell+1})$ to form path P'. It can be easily confirmed that these node pairs satisfy constraints (4.6)-(4.8), hence they are valid edges. Secondly, the definition of the new node $\mathbf{u}_\ell$ ensures that the path P' satisfies condition (4.11). Lastly, we must ensure that P' has at least the same weight as P. For this it is enough to show that

$$w(\mathbf{v}_{\ell-1}, \mathbf{u}_\ell) + w(\mathbf{u}_\ell, \mathbf{v}_{\ell+1}) \geq w(\mathbf{v}_{\ell-1}, \mathbf{v}_\ell) + w(\mathbf{v}_\ell, \mathbf{v}_{\ell+1}) \tag{B.5}$$

since all other edges are identical in paths P and P'.

In order to help visualize the proof of this last step, we note that the change from P to P' corresponds to changing only rows ℓ and $(\ell+1)$ in the packetization array. Precisely, the location of source symbols and redundancy symbols on these two rows are swapped. Figure B.3 shows rows ℓ and $(\ell+1)$ corresponding to path P (a) and path P' (b). This implies that the redundancy allocation of the two rows is swapped. Therefore, all source symbols which are swapped from one row to the other, will have the same amount of erasure protection, and their contribution to the expected distortion (hence, to the path weight) is unchanged. The only change in the path weight is due to the columns which contain source symbols on both rows ℓ and $(\ell+1)$.

Let column i contain a source symbol in both rows ℓ and $(\ell+1)$. Then the difference in the path weight between P' and P due to column i is

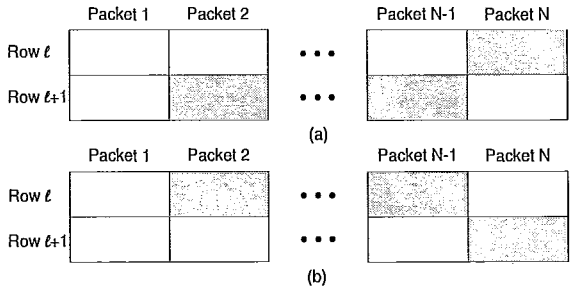

Figure B.3: Packetization array rows ℓ and $(\ell+1)$ prior to (a) and after (b) the path change in Lemma 1. Gray boxes represent redundancy symbols and white boxes represent source symbols.

$$\left[C\left(\sum_{j=1}^{N}\left(v_{\ell+1}^{(j)}-v_{\ell}^{(j)}\right)\right)-C\left(\sum_{j=1}^{N}\left(v_{\ell}^{(j)}-v_{\ell-1}^{(j)}\right)\right)\right]\times\left[\Delta D_i(v_{\ell}^{(i)})-\Delta D_i(v_{\ell+1}^{(i)})\right] \tag{B.6}$$

Note that the first factor in the product of (B.6) is >0 by our initial assumption that constraint (4.11) was not satisfied, and the second factor is ≥ 0 due to convexity of the R-D curves. Therefore the path weight difference between P' and P due to column i is ≥ 0. By extending the same proof to all columns which contain source symbols in both rows ℓ and $(\ell+1)$ we conclude that $\mathbf{w}(P')-\mathbf{w}(P)\geq 0$. In order to complete the proof we add that if path P violates condition (4.11) for several values ℓ then the above procedure for altering the path is applied iteratively until all violations are resolved.

Therefore, if the R-D curves of all substreams are convex the problem reduces to that of finding the maximum-weight path in G. Since G has $O((L+1)^{N+1})$ vertices and $O(2^N(L+1)^{N+1})$ edges (for each node there are at most 2^N-1 incoming edges), the M-UEP RD-ORA problem can be solved in $O(2^N(L+1)^{N+1})$ time under the

convexity assumption.

Bibliography

Alatan, A. A., Zhao, M., and Akansu, A. N. (2000). Unequal error protection of SPIHT encoded image bit streams. *IEEE J. Sel. Areas Commun.*, **18**, 814–818.

Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I. (1992). Image coding using wavelet transform. *IEEE Trans. Image Process.*, **1**(2), 205–220.

Chande, V. and Farvardin, N. (2000). Progressive transmission of images over memoryless channels. *IEEE J. Select. Areas Commun.*, **18**(6), 850–860.

Cho, S. and Pearlman, W. (2000). Error resilient compression and transmission of scalable video. In *Proc. SPIE*, volume 4115, pages 396–405, San Diego, CA.

Cohen, A., Daubechies, I., and Feauveau, J. (1992). Biorthogonal bases of compactly supported wavelets. *Commun. Pure and Applied Math*, **45**, 485–560.

Creusere, C. D. (1997). A new method of robust image compression based on the embedded zerotree wavelet algorithm. *IEEE Trans. Image Process.*, **6**(10), 1436–1442.

Dumitrescu, S., Wu, X., and Wang, Z. (2004). Globally optimal uneven error-protected packetization of scalable code streams. *IEEE Trans. Multimedia*, **6**(2), 230–239.

Dumitrescu, S., Wu, X., and Wang, Z. (2007). Efficient algorithms for optimal uneven protection of single and multiple scalable code streams against packet erasures. *IEEE Trans. Multimedia*, **9**(7), 1466–1474.

Gonzalez, R. and Woods, R. (2002). *Digital Image Processing.* Prentice Hall, Upper Saddle River, NJ, 2 edition.

Graham, R. L., Knuth, D. E., and Patashnik, O. (1994). *Concrete Mathematics.* Addison-Wesley, Upper Saddle River, NJ, 2 edition.

Hamzaoui, R., Stankovic, V., and Xiong, Z. (2005). Optimized error protection of scalable image bit streams : Advances in joint source-channel coding of images. *IEEE Signal Process. Mag.*, **22**(6), 91–107.

Kang, H. (1999). *Digital Color Halftoning.* SPIE Press, Bellingham, WA, 1 edition.

Kim, J., Mersereau, R. M., and Altunbasak, Y. (2003). Error-resilient image and video transmission over the internet using unequal error protection. *IEEE Trans. Image Process.*, **12**(2), 121–131.

Kurose, J. and Ross, K. (2008). *Computer Networking - A Top-Down Approach.* Addison Wesley, Boston, MA, 4 edition.

Mallat, S. (1987). A compact multiresolution representation: the wavelet model. In *Proc. IEEE Computer Society Workshop on Computer Vision*, pages 2–7, Washington, D.C.

Mohr, A., Riskin, E., and Ladner, R. (1999). Graceful degradation over packet erasure channels through forward error correction. In *Proc. DCC'99*, pages 92–101, Utah, USA.

Mohr, A., Ladner, R., and Riskin, E. (2000a). Approximately optimal assignment for unequal loss protection. In *Proc. ICIP'00*, pages 367–370, Vancouver, Canada.

Mohr, A., Riskin, E., and Ladner, R. (2000b). Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction. *IEEE J. Select. Areas Commun.*, **18**(6), 819–828.

Puri, R. and Ramchandran, K. (1999). Multiple description source coding through forward error correction codes. In *Proc. 33rd Asilomar Conference on Signals, Systems, and Computers*, volume 1, pages 342–346, California, USA.

Puri, R., Lee, K., Ramchandran, K., and Bharghavan, V. (2001). An integrated source transcoding and congestion control paradigm for video streaming in the Internet. *IEEE Trans. Multimedia*, **3**(1), 18–32.

Rivers, G., Dumitrescu, S., and Shirani, S. (2008). Novel R-D Optimized Uneven Erasure-Protection Strategy for Scalable Data Formed of Multiple Code Streams. In *Proc. MMSP'08*, pages 418–423, Cairns, Australia.

Rogers, J. and Cosman, P. (1998a). Robust wavelet zerotree image compression with fixed-length packetization. In *Proc. DCC'98*, pages 418–427, Utah, USA.

Rogers, J. and Cosman, P. (1998b). Wavelet zerotree image compression with packetization. *IEEE Signal Processing Lett.*, **5**, 105–107.

Said, A. and Pearlman, W. (1996). A new fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. Video Technol.*, **6**, 243–250.

Sayood, K. (2006). *Introduction to Data Compression.* Morgan Kaufmann, San Francisco, CA, 3 edition.

Shapiro, J. (1993). Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Process.*, **41**, 3445–3462.

Stankovic, V., Hamzaoui, R., and Saupe, D. (2003). Fast algorithm for rate-based optimal error protection of embedded codes. *IEEE Trans. Commun.*, **51**(11), 1788–1795.

Stankovic, V., Hamzaoui, R., and Xiong, Z. (2004a). Efficient channel code rate selection algorithms for forward error correction of packetized multimedia bitstreams in varying channels. *IEEE Trans. Multimedia*, **14**(2), 240–248.

Stankovic, V., Hamzaoui, R., and Xiong, Z. (2004b). Real-time error protection of embedded codes for packet erasure and fading channels. *IEEE Trans. Circuits Syst. Video Technol.*, **14**, 1064–1072.

T. Stockhammer, C. B. (2001). Progressive texture video streaming for lossy pakcet networks. In *Proc. 11th International Packet Video Workshop*, Kyongju, Korea.

Taubman, D. (2000). High performance scalable image compression with EBCOT. *IEEE Trans. Image Process.*, **9**, 1158–1170.

Thie, J. and Taubman, D. (2005). Optimal erasure protection strategy for scalably compressed data with tree-structured dependencies. *IEEE Trans. Image Process.*, **14**(12), 2002–2011.

Thomos, N., Boulgouris, N., and Strintzis, M. (2006a). Optimized transmission of

JPEG2000 streams over wireless channels. *IEEE Trans. Image Process.*, **15**(1), 54–67.

Thomos, N., Boulgouris, N., and Strintzis, M. (2006b). Product code optimization for determinate state LDPC decoding in robust image transmission. *IEEE Trans. Image Process.*, **15**(8), 2113–2119.

Wu, X., Cheng, S., and Xiong, Z. (2001). On packetization of embedded multimedia bitstreams. *IEEE Trans. Multimedia*, **3**(1), 132–140.

Zheng, T. (2008). *Two Techniques for Symmetric Multiple Description Coding with Reduced Storage Space Decoder.* Master's thesis, McMaster University, Hamilton, Ontario Canada.