Error-Resilient and Error Concealment 3-D SPIHT for Multiple Description Video Coding With Added Redundancy

By

Jie Zhu

A thesis submitted to
The Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of
The requirements for the degree of
Doctor of Philosophy

Ottawa-Carleton Institute for Electrical and Computer Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada

Copyright © 2012 by Jie Zhu.
NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

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

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

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

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

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

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

Bien que ces formulaires aient inclus dans la pagination, il n’y aura aucun contenu manquant.
The undersigned hereby recommend to
the Faculty of Graduate and Postdoctoral Affairs
acceptance of the thesis:

Error-Resilient and Error Concealment 3-D SPIHT for Multiple
Description Video Coding With Added Redundancy

Submitted by
Jie Zhu, B.Eng., M.A.Sc.
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

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

Dr. Richard M. Dansereau, Thesis Supervisor
Department of Systems and Computer Engineering

Dr. Sorina Dumitrescu, External Examiner
Department of Electrical and Computer Engineering
McMaster University

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada

June 2012
Abstract

To combat the significant video quality degradation caused by transmission errors, in this work we present a multiple description coding algorithm based on error-resilient and error concealment set partitioning in hierarchical trees (ERC-SPIHT). In this proposed approach, additional redundancy is generated by a wavelet projection based method of the spatial root subband and such redundancy is then intentionally inserted into the substreams. As a result, the novelty of the proposed approach is that the missing spatial root subband coefficients can be reconstructed by exploiting both inherent redundancy and inserted redundancy. This reconstruction procedure is implemented in two steps, first by using existing two-dimensional error concealment techniques, and second with the proposed root subband recovery approach. The former step is used to estimate the missing coefficients in the spatial root and high-frequency subbands by exploiting the inherent redundancy, while the latter attempts to utilize the inserted redundancy to further improve the precision in the estimation of the missing spatial root subband coefficients. The proposed root subband recovery method can be iteratively applied and accuracy of the reconstruction can be gradually increased with each iteration. Our simulation results on different
video sequences ("Football" and "Susie") show that the proposed method maintains error-resilience with high source coding efficiency. In particular, the proposed method achieves a significant improvement on the reconstructed video quality by up to 2.58 dB in the presence of full substream loss compared to ERC-SPIHT. We then extend the proposed video coding algorithm to the partial substream loss case. In such a context, we consider the SPIHT decoding range concept. The missing bitplane values of the spatial root subband coefficients can be reconstructed from three types of resources: inherent redundancy, added redundancy, and decoding range. The reconstruction procedure of the missing bitplane values is achieved by introducing a new feature (i.e., range comparison and replacement) to existing error concealment techniques and the proposed root subband recovery approach. Experimental results show that the proposed method can significantly enhance the reconstructed video quality by up to 7.82 dB in the partial substream loss case compared to ERC-SPIHT.
Acknowledgements

Foremost, I would like to express my sincerest gratitude to my supervisor, Dr. Richard M. Dansereau, for his continuous support, encouragement and patience throughout this thesis. He has been actively interested in my work and has always been available to advise me. His knowledgeable suggestions and editorial efforts have greatly improved the quality of this thesis.

Special thanks to Dr. Aysegul Cuhadar for her assistance and guidance in getting my graduate career started on the right foot and giving her valuable suggestions during the early stages of this work.

Last but not least, I am truly grateful to my dear parents for their endless love and unconditional support, both financially and emotionally throughout my degree. I would also like to thank my parents-in-law for their understanding, patience and encouragement. In particular I thank with love to my husband Hua. Understanding me best as a Ph.D. himself, Hua has been my best friend and great companion, loved, supported, encouraged, entertained, and helped me get through this agonizing period in the most positive way. I simply would not have finished this work without him. I dedicate this thesis to Hua with all my love.
# Table of Contents

Abstract                                                   ii  
Acknowledgements                                           iv  
Table of Contents                                          viii 
List of Figures                                            xiv 
List of Tables                                             xvi 
List of Acronyms                                           xvii  

1 Introduction                                              1  
  1.1 Background                                             1  
  1.1.1 Wavelet Zerotree Coding and 3-D SPIHT               2  
  1.1.2 Error Control in Video Coding                       3  
  1.1.3 MD Video Coding                                     5  
  1.1.4 Domain-Partitioning Based MD Coding With 3-D SPIHT  6  
  1.2 Thesis Motivations                                    8  
  1.3 Thesis Statement                                      10 
  1.4 Thesis Contributions                                  10 
  1.5 Thesis Organization                                   13
2 Background Review

2.1 Wavelet Transforms ................................................................. 16
   2.1.1 1-D Wavelet Transforms .................................................. 16
   2.1.2 2-D Wavelet Transforms .................................................. 17
   2.1.3 3-D Wavelet Transforms .................................................. 18

2.2 Wavelet-Based Coding Procedures and Zerotrees .......................... 20

2.3 2-D SPIHT ................................................................. 23

2.4 3-D SPIHT ................................................................. 25

2.5 STTP-SPIHT and ERC-SPIHT ............................................... 28

2.6 Error Concealment Techniques ............................................... 33
   2.6.1 Bilinear Interpolation ...................................................... 33
   2.6.2 Locally Adaptive Passive Error Concealment ...................... 33
   2.6.3 Interband Interpolation .................................................... 35

2.7 Video Quality Assessment ....................................................... 36
   2.7.1 Peak Signal-to-Noise Ratio .............................................. 37
   2.7.2 Structural Similarity Index .............................................. 38

3 Proposed Domain-Partitioning Based Multiple Description Video Coding Algorithm using 3-D SPIHT: A Simple Case Study 39

3.1 Spatial Root Subband ............................................................ 40

3.2 Proposed Algorithm with Some Design Issues ............................... 43
   3.2.1 System Overview ............................................................ 43
   3.2.2 Additional Redundancy Generation with an Averaging Approach 45
   3.2.3 Domain Partitioning and Encoding ..................................... 47
   3.2.4 Decoding over Noiseless Channels ..................................... 52

3.3 Reconstruction of Lost Wavelet Coefficients ................................ 53
   3.3.1 2-D Error Concealment .................................................. 54
   3.3.2 Root Subband Recovery .................................................. 55
3.4 Simulation Results .......................................................... 57
  3.4.1 Experimental Data ......................................................... 57
  3.4.2 Source Coding Efficiency ............................................... 58
  3.4.3 Error Concealment Performance .................................... 61
3.5 Conclusions ......................................................................... 68

4 Wavelet Projection Based Additional Redundancy Generation and
  Error Concealment Techniques: Full Substream Loss ............. 72
  4.1 Additional Redundancy Generation using 2-D Wavelet Decomposition 73
  4.2 Proposed Error Concealment Technique for Full Substream Loss ...... 78
    4.2.1 2-D Error Concealment ................................................. 79
    4.2.2 Iterative Root Subband Recovery .................................... 80
    4.2.3 Root Subband Recovery with Haar Wavelet Transform .......... 84
  4.3 Simulation Results .......................................................... 90
    4.3.1 Experimental Data ......................................................... 90
    4.3.2 Source Coding Efficiency ............................................... 91
    4.3.3 Error Concealment Performance .................................... 91
      4.3.3.1 Performance Evaluation with Different Missing Patterns of Coefficients 94
      4.3.3.2 Performance Evaluation at Different Coding Rates ......... 100
      4.3.3.3 Performance Evaluation with Different Number of Sub-streams Lost .... 101
      4.3.3.4 Visual Comparison .................................................... 104
  4.4 Conclusions ....................................................................... 107

5 Revised Domain-Partitioning Based Multiple Description Video Coding Algorithm in Conjunction with 3-D SPIHT Decoding Range:
  Partial Substream Loss ...................................................... 111
List of Figures

2.1 Procedure of 1-D discrete wavelet transform with two-levels of decomposition. ............................................ 18
2.2 Subband structure of 2-D dyadic wavelet transform with two levels of decomposition. ........................................ 19
2.3 Subband structures of 3-D wavelet transforms with two levels of decomposition. (a) 3-D dyadic wavelet transform. (b) 3-D wavelet-packet transform with 2-D dyadic plus independent 1-D dyadic. .......... 20
2.4 Spatial orientation tree structure in 2-D SPIHT. ................. 24
2.5 Comparison of spatio-temporal orientation tree structures. (a) 3-D symmetric tree structure in 3-D SPIHT. (b) 3-D asymmetric tree structure in AT-SPIHT. (From Fig. 3 in [41]) ......................... 27
2.6 Structure of the STTP-SPIHT decomposition using contiguous grouping. 30
2.7 Structure of the ERC-SPIHT decomposition using dispersive grouping. 31
2.8 Block diagram of domain-partitioning based MD video coding using 3-D SPIHT in conjunction with error concealment. ................. 32
3.1 The structure of the spatial root subband defined in our proposed algorithm. (a) 2-D wavelet domain. (b) 3-D wavelet domain. .......... 41
3.2 Framework of the proposed video coding algorithm. ......................... 44
3.3 Relationship between original video sequence and additional redundancy generated by an averaging approach. ......................... 46
3.4 Procedure of domain partitioning and SPIHT encoding in the proposed algorithm.

3.5 An example of substream-\(i\) generated by the proposed algorithm with reverse order concatenation of the additional redundancy.

3.6 Comparison of positions of the missing coefficients that are estimated using 2-D error concealment in the spatial root subband of size 8 \(\times\) 8. (a) ERC-SPIHT. (b) Our proposed MD video coding algorithm.

3.7 Comparison of frame by frame PSNR (dB) of “Football” video sequence in noiseless channels at different coding rates \(c\) (\(S = 4\) substreams).

3.8 Comparison of frame by frame PSNR (dB) of “Susie” video sequence in noiseless channels at different coding rates \(c\) (\(S = 4\) substreams).

3.9 Comparison of frame by frame PSNR (dB) of “Football” video sequence in noisy channels with bilinear interpolation at different coding rates \(c\) (\(S = 4\) substreams and one substream is lost). (a) Substream-2 is lost. (b) Substream-4 is lost.

3.10 Comparison of frame by frame PSNR (dB) of “Susie” video sequence in noisy channels with bilinear interpolation at different coding rates \(c\) (\(S = 4\) substreams and one substream is lost). (a) Substream-2 is lost. (b) Substream-4 is lost.

3.11 Comparison of frame by frame PSNR (dB) of “Football” sequence in noisy channels with LAPEC at different coding rates \(c\) (\(S = 4\) substreams and one substream is lost). (a) Substream-2 is lost. (b) Substream-4 is lost.

3.12 Comparison of frame by frame PSNR (dB) of “Susie” sequence in noisy channels with LAPEC at different coding rates \(c\) (\(S = 4\) substreams and one substream is lost). (a) Substream-2 is lost. (b) Substream-4 is lost.
3.13 352 × 240 “Football” sequence ($S = 4$ substreams and $c = 1.0$ bpp). (a) Original sequence (frame 47). (b) Corrupted sequence when substream 4 is missing: $PSNR = 11.93$ dB. (c) ERC-SPIHT with bilinear interpolation: $PSNR = 23.96$ dB. (d) ERC-SPIHT with LAPEC: $PSNR = 24.07$ dB. (e) Proposed algorithm with bilinear interpolation: $PSNR = 24.61$ dB, (f) Proposed algorithm with LAPEC: $PSNR = 24.63$ dB. 69

3.14 352 × 240 “Susie” sequence ($S = 4$ substreams and $c = 1.0$ bpp). (a) Original sequence (frame 27). (b) Corrupted sequence when substream 2 is missing: $PSNR = 13.47$ dB. (c) ERC-SPIHT with bilinear interpolation: $PSNR = 29.14$ dB. (d) ERC-SPIHT with LAPEC: $PSNR = 29.48$ dB. (e) Proposed algorithm with bilinear interpolation: $PSNR = 30.10$ dB. (f) Proposed algorithm with LAPEC: $PSNR = 30.22$ dB. 70

4.1 Relationship between original video sequence and additional redundancy generated by a 2-D wavelet decomposition. .................... 74

4.2 Illustration of the proposed iterative root subband recovery method for the full substream loss case. ................................. 83

4.3 Comparison of PSNR (dB) and MSSIM of “Football video sequence in noiseless channels at different coding rates $c$ ($S = 16$ substreams). (a) PSNR (dB). (b) MSSIM. ......................................................... 92

4.4 Comparison of PSNR (dB) and MSSIM of “Susie” video sequence in noiseless channels at different coding rates $c$ ($S = 16$ substreams). (a) PSNR (dB). (b) MSSIM. ......................................................... 93

4.5 Patterns randomly chosen from the five categories in Sec. 4.3.3.1 when considering Haar $2 \times 2$ subblock based missing coefficient cases in $4 \times 4$ blocks ($S = 16$ substreams and four substreams are lost with dispersive grouping). (a) Pattern 1. (b) Pattern 2. (c) Pattern 3. (d) Pattern 4. (e) Pattern 5. ......................................................... 96
4.6 Comparison of final PSNR (dB) and MSSIM of “Football” sequence at different coding rates $c$ ($S = 16$ substreams and four substreams are lost). (a) PSNR (dB) and (b) MSSIM. ...................................................... 102

4.7 Comparison of final PSNR (dB) and MSSIM of “Susie” sequence at different coding rates $c$ ($S = 16$ substreams and four substreams are lost). (a) PSNR (dB) and (b) MSSIM. ....................................................... 103

4.8 Comparison of final PSNR (dB) and MSSIM of “Football” sequence with different number of substreams lost ($S = 16$ substreams and $c = 1.0$ bpp). (a) PSNR (dB). (b) MSSIM. ....................................................... 105

4.9 Comparison of final PSNR (dB) and MSSIM of “Susie” sequence with different number of substreams lost ($S = 16$ substreams and $c = 1.0$ bpp). (a) PSNR (dB). (b) MSSIM. ............................................................. 106

4.10 $352 \times 240$ “Football” sequence ($S = 16$ substreams and $c = 1.0$ bpp). (a) Original sequence (frame 47). (b) Corrupted sequence using the missing coefficient pattern in Fig. 4.5(d): $PSNR = 11.33$ dB and $MSSIM = 0.4179$. (c) Scenario 1: $PSNR = 23.25$ dB and $MSSIM = 0.7377$. (d) Scenario 2: $PSNR = 22.69$ dB and $MSSIM = 0.7199$. (e) Scenario 3 ($I = 1$): $PSNR = 23.91$ dB and $MSSIM = 0.7410$. (f) Scenario 3 ($I = 5$): $PSNR = 24.49$ dB and $MSSIM = 0.7456$. (g) Scenario 3 ($I = 10$): $PSNR = 24.65$ dB and $MSSIM = 0.7470$. (h) Scenario 3 ($I = 50$): $PSNR = 24.77$ dB and $MSSIM = 0.7479$. (i) Scenario 4: $PSNR = 24.41$ dB and $MSSIM = 0.7464$. ........................... 108
4.11 352 x 240 “Susie” sequence (S = 16 substreams and c = 1.0 bpp).

(a) Original sequence (frame 27). (b) Corrupted sequence using the
missing coefficient pattern in Fig. 4.5(a): $PSNR = 12.11$ dB and
$MSSIM = 0.4189$. (c) Scenario 1: $PSNR = 29.80$ dB and $MSSIM = 0.8923$. (d) Scenario 2: $PSNR = 26.38$ dB and $MSSIM = 0.8802$.

(e) Scenario 3 ($I = 1$): $PSNR = 30.89$ dB and $MSSIM = 0.8982$. (f) Scenario 3 ($I = 5$): $PSNR = 31.56$ dB and $MSSIM = 0.9029$. (g)
Scenario 3 ($I = 10$): $PSNR = 31.86$ dB and $MSSIM = 0.9050$. (h)
Scenario 3 ($I = 50$): $PSNR = 32.17$ dB and $MSSIM = 0.9077$. (i)
Scenario 4: $PSNR = 30.68$ dB and $MSSIM = 0.8946$.

5.1 Framework of the proposed video coding algorithm with extensions to
partial substream loss. ................................................................. 114

5.2 Binary representation of the coefficient $3D\hat{W}_{x,y,n}$ partially decoded with
some missing bitplane values. .................................................. 118

5.3 Procedure of the range comparison and replacement used in bilinear
interpolation and iterative root subband recovery. (a) $3D\hat{W}_{x,y,n}^{\text{min}} \leq
3D\hat{W}_{x,y,n} \leq 3D\hat{W}_{x,y,n}^{\text{max}}$. (b) $3D\hat{W}_{x,y,n} > 3D\hat{W}_{x,y,n}^{\text{max}}$. (c) $3D\hat{W}_{x,y,n} < 3D\hat{W}_{x,y,n}^{\text{min}}$. 121

5.4 Illustration of the proposed iterative root subband recovery method for
the partial substream loss case. .................................................. 125

5.5 Procedure of the range comparison and possible replacement used in
iterative root subband recovery. (a) $3D\hat{a}_{x,y,n}^{\text{min}} \leq 3D\hat{a}_{x,y,n} \leq 3D\hat{a}_{x,y,n}^{\text{max}}$.
(b) $3D\hat{a}_{x,y,n} > 3D\hat{a}_{x,y,n}^{\text{max}}$. (c) $3D\hat{a}_{x,y,n} < 3D\hat{a}_{x,y,n}^{\text{min}}$. 126

5.6 Comparison of PSNR (dB) and MSSIM of “Football” video sequence
in noisy channels (S = 16 substreams with 1 substream corrupted at
the end of different bitplanes d, c = 1.0 bpp and $c^R = 8.0$ bpp). (a)
PSNR (dB). (b) MSSIM. ............................................................ 130
5.7 Comparison of PSNR (dB) and MSSIM of "Susie" video sequence in noisy channels ($S = 16$ substreams with 1 substream corrupted at the end of different bitplanes $d$, $c = 1.0$ bpp and $c^R = 5.0$ bpp). (a) PSNR (dB). (b) MSSIM.

5.8 Transition states of a two-state Markov model.

5.9 $352 \times 240$ "Football" video sequence ($S = 16$ substreams with 1 substream lost after the bitplane $d = 11$, $c = 1.0$ bpp and $c^R = 8.0$ bpp). (a) Original sequence (frame 47). (b) ERC-SPIHT: $PSNR = 27.01$ dB and $MSSIM = 0.8259$. (c) Proposed algorithm: $PSNR = 28.18$ dB and $MSSIM = 0.8344$.

5.10 $352 \times 240$ "Susie" video sequence ($S = 16$ substreams with 1 substream lost after the bitplane $d = 11$, $c = 1.0$ bpp and $c^R = 5.0$ bpp). (a) Original sequence (frame 27). (b) ERC-SPIHT: $PSNR = 31.70$ dB and $MSSIM = 0.9240$. (c) Proposed algorithm: $PSNR = 38.44$ dB and $MSSIM = 0.9586$. 

xiv
List of Tables

3.1 Comparison of average PSNR (dB) of “Football” and “Susie” video sequences in noiseless channels at different coding rates $c$ ($S = 4$ substreams) .......................................................... 60

3.2 Comparison of average PSNR (dB) of “Football” and “Susie” video sequences in noisy channels with bilinear interpolation at different coding rates $c$ ($S = 4$ substreams and one substream is lost) .................... 62

3.3 Comparison of average PSNR (dB) of “Football” and “Susie” video sequences in noisy channels with LAPEC at different coding rates $c$ ($S = 4$ substreams and one substream is lost) .................. 68

4.1 Comparison of PSNR (dB) and MSSIM for “Football” video sequence under different missing patterns of Fig. 4.5 ($S = 16$ substreams and $c = 1.0$ bpp) .......................................................... 98

4.2 Comparison of PSNR (dB) and MSSIM for “Susie” video sequence under different missing patterns of Fig. 4.5 ($S = 16$ substreams and $c = 1.0$ bpp) .......................................................... 99

5.1 Comparison of PSNR (dB) and MSSIM of “Football” video sequence under different packet loss rates ($S = 16$ substreams, $c = 1.0$ bpp, $c^R = 8.0$ bpp and $L = 200$ bits) ................................. 134
5.2 Comparison of PSNR (dB) and MSSIM of “Susie” video sequence under different packet loss rates ($S = 16$ substreams, $c = 1.0$ bpp, $c^R = 5.0$ bpp and $L = 200$ bits) ................................................................. 134

5.3 Comparison of PSNR (dB) and MSSIM of “Football” video sequence under different packet loss rates ($S = 16$ substreams, $c = 1.0$ bpp, $c^R = 8.0$ bpp and $L = 2000$ bits) ..................................................... 135

5.4 Comparison of PSNR (dB) and MSSIM of “Susie” video sequence under different packet loss rates ($S = 16$ substreams, $c = 1.0$ bpp, $c^R = 5.0$ bpp and $L = 2000$ bits) ......................................................... 136
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-D</td>
<td>one-dimensional</td>
</tr>
<tr>
<td>2-D</td>
<td>two-dimensional</td>
</tr>
<tr>
<td>3-D</td>
<td>three-dimensional</td>
</tr>
<tr>
<td>ARQ</td>
<td>Auto retransmission request</td>
</tr>
<tr>
<td>AT-SPIHT</td>
<td>Asymmetric tree structure for 3-D set partitioning in hierarchical trees</td>
</tr>
<tr>
<td>BER</td>
<td>Bit error rate</td>
</tr>
<tr>
<td>BL</td>
<td>Basic layer</td>
</tr>
<tr>
<td>bpp</td>
<td>bit/pixel</td>
</tr>
<tr>
<td>CDF</td>
<td>Cohen-Daubechies-Feauveau</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete cosine transform</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete wavelet transform</td>
</tr>
<tr>
<td>EL</td>
<td>Enhancement layer</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ERC-SPIHT</td>
<td>Error resilient and error concealment 3-D set partitioning in hierarchical trees</td>
</tr>
<tr>
<td>EZW</td>
<td>Embedded zerotree wavelet</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward error correction</td>
</tr>
<tr>
<td>FR</td>
<td>Full reference</td>
</tr>
<tr>
<td>GOF</td>
<td>Group of frames</td>
</tr>
<tr>
<td>LAPEC</td>
<td>Locally adaptive passive error concealment</td>
</tr>
<tr>
<td>LC</td>
<td>Layered coding</td>
</tr>
<tr>
<td>LIP</td>
<td>List of insignificant pixels</td>
</tr>
<tr>
<td>LIS</td>
<td>List of insignificant sets</td>
</tr>
<tr>
<td>LLSE</td>
<td>Linear least-square error</td>
</tr>
<tr>
<td>LSP</td>
<td>List of significant pixels</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a posterior</td>
</tr>
<tr>
<td>MCP</td>
<td>Motion-compensated prediction</td>
</tr>
<tr>
<td>MD</td>
<td>Multiple description</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean-squared error</td>
</tr>
<tr>
<td>MSSIM</td>
<td>Mean structural similarity</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MV</td>
<td>Motion vector</td>
</tr>
<tr>
<td>NR</td>
<td>No-reference</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak signal-to-noise ratio</td>
</tr>
<tr>
<td>R-D</td>
<td>Rate distortion</td>
</tr>
<tr>
<td>RR</td>
<td>Reduced reference</td>
</tr>
<tr>
<td>SD</td>
<td>Squared difference</td>
</tr>
<tr>
<td>SPIHT</td>
<td>Set partitioning in hierarchical trees</td>
</tr>
<tr>
<td>SSIM</td>
<td>Structural Similarity</td>
</tr>
<tr>
<td>STTP-SPIHT</td>
<td>Spatial and temporal tree preserving 3-D set partitioning in hierarchical trees</td>
</tr>
<tr>
<td>VLC</td>
<td>Variable length coding</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

With the rapid development of packet switching networks, such as the Internet and wireless networks, many new communication applications that deliver multimedia content over networks are becoming mainstream. Some of these applications include: video conferencing, video telephony, video on demand, and television broadcasting through the Internet. However, delivery of high quality video over packet switching networks poses great technical challenges for a video communication system design. One of the challenges is that raw size of the video data is too large compared to bandwidth available for the transmission. For this reason, compression is essential in a successful video communication system and video needs to encoded within the bandwidth limit. On the other hand, the transport environment of packet networks is not always reliable. For example, random bit errors and packet losses may frequently occur. While reducing the data volume efficiently, compression also makes the coded
video bitstream highly sensitive to such transmission errors. To alleviate this problem, the compressed bitstream needs to be sufficiently error-resilient. Hence, error control techniques are also critical in a video communication system for video applications.

1.1.1 Wavelet Zerotree Coding and 3-D SPIHT

Over the last decade, wavelet zerotree coding techniques have established an important presence in the task of two-dimensional (2-D) image compression, and they are increasingly being considered for three-dimensional (3-D) video compression as well. The embedded zerotree wavelet (EZW) algorithm proposed by Shapiro [1] was the first coder to make use of the zerotree concept. It was further developed by Said and Pearlman as 2-D set partitioning in hierarchical trees (SPIHT) [2]. Later, Kim et al. extended SPIHT to 3-D for video compression [3,4]. The 3-D SPIHT algorithm has shown promise as an effective and computationally simple video coding scheme with excellent rate-distortion (R-D) performance. It has also shown potential in providing a scalable representation of video coding in spatial resolution, temporal resolution, and quality with low computational cost. However, the 3-D SPIHT algorithm, similar to all wavelet zerotree coding schemes, produces variable-length bitstreams. As a result, 3-D SPIHT coding sacrifices robustness to transmission errors [5]. A single bit error can cause an uncontrolled degradation of the reconstructed video quality. To alleviate this problem, it is important to incorporate error-resilience with the SPIHT encoding algorithm for robust video transmission over error-prone networks.
1.1.2 Error Control in Video Coding

There have been various error control mechanisms proposed to provide error-resilience for video coders [6–9]. They are categorized into three types:

- Those introduced either at the source encoder or channel encoder, such as forward error correction (FEC) and source coding techniques. FEC is designed to add additional channel redundancy to the bitstream. For example, a video bitstream is first split into several segments, each of which is packetized into \( k \) packets. For each segment, a channel code is applied to the \( k \) packets so that an \( n \)-packet block is generated, where \( n > k \). This block of packets is then transmitted to the receiver. To successfully recover a segment, we must guarantee that any \( k \) packets in the \( n \)-packet block are received correctly. However, FEC is ineffective in packet-based networks when a large number of packets may be lost in clusters and such loss exceeds the recovery capability of the FEC code [10]. In such a scenario, more than \( n - k \) packets are corrupted due to transmission errors. Examples of source coding techniques include layered coding (LC) [11] and multiple description (MD) coding [12]. LC and MD coding have similar characteristics: they both create multiple substreams. However, they are different in the sense that the substreams created in LC have different levels of importance while the substreams created in MD coding are equally important. Specifically, the layered coder encodes the video into several layers, which are one basic layer (BL) and one or more enhancement layers (EL). Each EL received correctly can be used with the BL to incrementally improve the
reconstructed video quality. But without the BL, video sequences cannot be reconstructed sufficiently. In contrast to LC, MD coders encode the video into multiple substreams with equal importance. This means that there is no hierarchy among the substreams and each substream can be independently decoded. This property makes MD coding highly suitable for error-prone networks. Consequently, MD coding has recently attracted a lot of attention as a promising approach for error resilient video transmission. For a comprehensive performance comparison between LC and MD coding the reader is recommended to refer to [13,14].

- Those invoked at the decoder to reduce the deleterious effect of transmission errors. These techniques are referred to as error concealment techniques, which can be further classified as: active and passive error concealment [15]. In active error concealment, auto retransmission request (ARQ) [16] is applied for the receiver to request retransmission of any corrupted information. If the amount of data lost is not significant, we can obtain successful reconstruction at the end of decoding process. However, retransmission may not be acceptable for real-time video applications due to long round-trip delays. Passive error concealment techniques are used to estimate the missing information from the correctly received neighboring data, by making use of the correlation among the neighboring data. These neighboring data can be spatially neighboring data, temporally neighboring data, or a combination of both [17]. It should be noted that, to improve the error resilience of a video coder, source coding and error
concealment techniques can be combined together.

- Those which requires the interactions between the encoder and decoder, so that the encoder adapts its operations according to the transmission error conditions detected at the decoder. An adaptive video coder is attractive for error-prone networks. It is due to the fact that the transmission errors in networks usually vary with time. In order to implement an adaptive video coder, the network should provide a feedback channel, so that the encoder knows the transmission error conditions at the decoder [18].

1.1.3 MD Video Coding

As discussed above, MD coding has recently attracted attention as a promising approach for error-resilient video transmission [12]. In such a scenario, an MD video coder divides the source information into several substreams which are then separately transmitted over the network (in our work, each substream is a single description, so we will use the terms “description” and “substream” interchangeably throughout this document). Each substream is independently decoded so that the errors in some of the substreams do not jeopardize the decoding of the correctly received substreams. The quality of the decoded information improves as the number of correctly received substreams increases. Another desirable feature is that the substreams created in MD coding are equally important. This is attractive in the context of Internet transmission, where none of the packets receive preferential treatment. For this reason, extensive research efforts have been undertaken to develop MD video codecs. Among
existing MD coding schemes, the domain-partitioning approach has shown to be efficient in achieving MD coding [19-21]. Consider a signal \( f \) defined on the domain \( D \).

In domain-partitioning, \( S \) descriptions of a signal \( f \) are \( P_1, P_2, \ldots, P_S \) such that

\[
\bigcup_{i=1}^{S} P_i = D. \tag{1.1}
\]

In addition to domain-partitioning, some approaches to realize MD coding include scalar quantizations [22–24], correlating transforms [25–32], and frame expansions [33–35]. An excellent review of some notable MD video coding schemes can be found in [36].

1.1.4 Domain-Partitioning Based MD Coding With 3-D SPIHT

In our work, we are interested in investigating the error resilience performance offered by the domain-partitioning based MD coding approach built upon SPIHT. Several domain-partitioning based MD coders using 2-D or 3-D SPIHT [37–44] have been proposed in the literature. One successful method is spatial and temporal tree preserving 3-D SPIHT (STTP-SPIHT) [39]. The main idea is to perform a wavelet transform on a group of frames (GOF) both in the spatial and temporal domain. The wavelet coefficients are divided into some number \( S \) of different spatio-temporal (s-t) tree groups by using the contiguous grouping method. Each group is independently encoded with the 3-D SPIHT algorithm and separately transmitted over the network. Consequently, \( S \) SPIHT-encoded substreams are generated where each substream corresponds to a specific region of the GOF. The advantage of STTP-SPIHT is that
errors in one SPIHT-encoded substream do not adversely affect other substreams. As a result, any decoding failure in STTP-SPIHT only affects the associated region in the GOF. This is in contrast to 3-D SPIHT, where the decoding failure affects the full extent of the GOF. Based on STTP-SPIHT, the error resilient and error concealment 3-D SPIHT (ERC-SPIHT) scheme was proposed in [40]. Compared to STTP-SPIHT, ERC-SPIHT uses a dispersive grouping mechanism to divide the wavelet coefficients into $S$ different spatio-temporal tree groups.

Error-resilience of a domain-partitioning based MD video coder with 3-D SPIHT can be further ameliorated when the coder is supplemented with an error concealment technique at the decoder. In this context, error concealment is performed to estimate the missing information from the correctly received data. It has been demonstrated that even simple error concealment, such as bilinear interpolation (missing coefficients are estimated by the average value of their four adjacent neighbors), can improve the R-D performance when combined with STTP-SPIHT and ERC-SPIHT [41]. In this case, bilinear interpolation exploits the fact that a video signal in the wavelet transform domain is spatially and temporally smooth in a statistical sense. In addition, it has been shown in [41] that ERC-SPIHT gives about 1 dB better performance than STTP-SPIHT in the presence of full substream loss when bilinear interpolation is applied. Such error concealment improvement results from the dispersive grouping used by ERC-SPIHT: the spatially adjacent coefficients are spread onto different substreams and better error concealment performance can be achieved by the increase in availability of correctly received spatial neighbors.
1.2 Thesis Motivations

To improve the error-resilience performance of 3-D SPIHT coding, two domain-partitioning based MD video coding algorithms, STTP-SPIHT and ERC-SPIHT, have been proposed in the literature. Another error control method, error concealment, can be used with STTP-SPIHT and ERC-SPIHT to achieve a graceful quality degradation when transmission errors are unavoidable. However, it may still cause reduced presentation quality since it relies completely on the inherent redundancy present in the signal (for example, spatial and temporal smoothness in the wavelet domain). In particular, the coefficients in the spatial root subband (i.e., the approximation subband on the highest spatial decomposition level) typically dominate the signal energy. Small relative errors in the estimation of the missing coefficients in the spatial root subband can have a significant impact on the overall distortion.

Motivated by this observation, we are interested in designing a video coder to reduce such impact in this work. An elegant solution is to inject additional redundancy into the substreams so that the missing coefficients in the spatial root subband can be recovered with a high level of accuracy. By additional redundancy, we mean redundancy that is added intentionally to the signal instead of the inherent redundancy present in the signal. However, the downside to incorporating additional redundancy is reduced source coding efficiency. The key challenge in developing such a video coder is how to balance the tradeoff between source coding efficiency at the encoder and error concealment performance at the decoder.

A number of domain-partitioning based MD coding schemes with additional re-
dundancy have been proposed and successfully applied to wavelet-based image and video coders. One example of a related work was introduced in [45], the authors use a uniformly distributed random mask for each subband of the wavelet coefficients to generate different groups. These groups are then quantized at different rates and combined together so as to generate the descriptions. In [46], additional redundancy is introduced in each substream by adding a partial version of the other substreams. As a result, each substream consists of a main substream encoded at a high rate, and several secondary substreams encoded with a low rate. If some substreams are corrupted due to transmission errors, the missing coefficients are recovered from the other substreams that are correctly received. Recently, this method is applied to the JPEG 2000 framework [47–51] in [52], in which each JPEG 2000 code-block is encoded at two rates, one in each description, and the rate allocation is determined by Lagrangian optimization. Similar MDC schemes that exploit the data R-D characteristics are presented in [53–56]. Another example has been introduced in [57], in which three copies of the root subband information are individually transmitted. However, the problems with these solutions are that they only exploit the additional redundancy to estimate the missing coefficients but such additional redundancy may considerably decrease source coding efficiency and eventually yield a significantly reduced reconstruction quality.

In this work, we mainly focus on designing an error-resilient MD coding framework for SPIHT-coded video with added redundancy. We consider both the coding efficiency and the error concealment performance. In contrast to existing algorithms, the missing coefficients in the spatial root subband can be reconstructed by exploiting
both inherent redundancy and additional redundancy.

1.3 Thesis Statement

This thesis aims to design a domain-partitioning based MD coding framework in which missing coefficients can be initially estimated from the inherent redundancy and further refined from the additional redundancy. We apply this framework to 3-D SPIHT-coded video: the additional redundancy is generated by a wavelet projection based method of the spatial root subband and the missing coefficients are reconstructed by exploiting the consistency between the coefficients and inherent/additional redundancy in conjunction with constraints from SPIHT decoding range.

1.4 Thesis Contributions

In this thesis, we study the domain-partitioning based MD video coding algorithm built upon 3-D SPIHT in conjunction with error concealment. We propose an error-resilient MD coding framework for SPIHT-coded video with added redundancy sent across error-prone networks. As a result, the novelty of the proposed approach is that the spatial root subband coefficients lost during transmission in any substream can be reconstructed by exploiting both inherent redundancy and inserted redundancy. The results of this thesis have been published at a number of conferences. Parts of the material have also been accepted for publication, or submitted for possible publication, in journals.
We start with a simple example in which additional redundancy is the average value of $2 \times 2$ wavelet coefficients in the spatial root subband and such redundancy is then intentionally inserted into the substreams by a careful video coding strategy. Correspondingly, the missing coefficients in the spatial root subband can be recovered directly from the added redundancy in a specific scenario (i.e., only four substreams are created and only one of the substreams is allowed lost due to noisy channels). In addition, a modified SPIHT coder is introduced. The performance of the proposed MD video coder is examined with respect to source coding efficiency and error concealment performance. Experimental results on different video sequences show that the proposed algorithm maintains error-resilience with high coding efficiency. In particular, our results demonstrate that the proposed algorithm achieves a significant improvement on the reconstructed video quality by up to 1.6899 dB compared to ERC-SPIHT when one substream is lost. These results have been published in *Proc. IEEE PacRim Conf. Commun., Computers and Signal Process.* [58].

A general video coding algorithm is proposed and it can be used in any type of missing coefficient patterns with an arbitrary number $S$ of substreams. In the generalized approach, additional redundancy is generated by a wavelet projection based method: each frame of the spatial root subband is spatially decomposed by one additional transform level and the resulting coefficients in the approximation subband are grouped together in a 3-D structure to form the additional redundancy. We first focus on the full substream loss case. The reconstruction procedure of the missing coefficients in the spatial root subband is implemented by two steps: first by using existing 2-D error concealment techniques (such as bilinear interpolation), and sec-
ond with the proposed root subband recovery approach. The former step is used to estimate the missing coefficients in the spatial root and high-frequency subbands by exploiting the inherent redundancy, while the latter attempts to utilize the inserted redundancy to further improve the precision in the estimation of the missing spatial root subband coefficients. The proposed root subband recovery method can be iteratively applied and accuracy of the reconstruction can be gradually increased with each iteration. We carry out our experiments for different coding rates under different missing coefficient patterns. The proposed video coder enhances the reconstructed video quality significantly by up to 2.58 dB in the presence of full substream loss compared to ERC-SPIHT. These results have been published in *Int. Conf. Image Signal Process. (ICISP’10)* [59] and accepted to publication in *IEEE Trans. Circuits Syst. Video Technol.* [60].

We then extend the proposed MD video coder to the partial substream loss case. The main modification of the video coder is the incorporation of the SPIHT decoding range concept to the system. As a result, the missing bitplane values of the spatial root subband coefficients in any substream can be reconstructed from three types of resources: inherent redundancy, additional redundancy and SPIHT decoding range. The reconstruction procedure of the missing bitplane values is achieved by introducing a new feature (i.e., range comparison and replacement) to bilinear interpolation and the proposed root subband recovery approach. We compare our proposed video coding algorithm to existing video coders in the partial substream loss case. The simulation results show that our proposed method provides better video reconstructions by up to 7.82 dB compared to ERC-SPIHT. The results will soon be submitted for possible
1.5 Thesis Organization

The reminder of the thesis is organized as follows:

Chapter 2 gives a background review on basic concepts and related topics involved in the rest of the thesis. We review the basics of wavelet zerotree coding and some particular coders, such as 2-D and 3-D SPIHT. We then survey some well-known domain-partitioning based MD video coding algorithms built upon 3-D SPIHT. We also briefly introduce some important error concealment techniques and video quality assessment metrics.

In Chapter 3, we focus on a simple scenario. We first proposed a domain-partitioning based MD coding framework for SPIHT-coded video. We then design a simplified dataset as additional redundancy and propose our error concealment technique.

Chapter 4 generalizes the technique proposed in Chapter 3 for the generation of an arbitrary number $S$ of substreams. In this chapter, we concentrate on the full substream loss case. We propose a wavelet projection based additional redundancy generation and root subband recovery algorithm. We show that the proposed root subband recovery can be used with any existing 2-D error concealment techniques to estimate the missing coefficients in the spatial root subband. We also show that the proposed root subband recovery scheme can be applied under any type of missing coefficient patterns.
In Chapter 5, we investigate the partial substream loss case and introduce some modifications to the proposed root subband recovery. We show that these modifications can be used to find the missing bitplane values of the spatial root subband coefficients and eventually improve the reconstructed video quality.

Finally, Chapter 6 concludes the thesis with directions for future research work.
Chapter 2

Background Review

In this chapter, we review some basic concepts of discrete wavelet transforms and wavelet-based coding techniques as well as several state-of-the-art video coders. First, we review the 1-D discrete wavelet transform and its extensions to 2-D and 3-D in Section 2.1. Following is a brief review of typical wavelet-based coding procedures in Section 2.2. We then introduce the 2-D SPIHT image coding approach in Section 2.3 and the 3-D SPIHT video coding mechanism is presented in Section 2.4. In Section 2.5, we also introduce two important domain-partitioning based MD video coders built upon 3-D SPIHT. In Section 2.6 we describe three error concealment techniques, which can be integrated into a domain-partitioning based MD video coder to further improve error-resilience. Finally, we discuss some video quality assessment techniques in Section 2.7.
2.1 Wavelet Transforms

Due to its desirable time-frequency characteristics, the wavelet transform was first introduced into the field of signal processing as an excellent computational tool for multi-resolution signal analysis in the 1980s [61]. The wavelet transform was developed in image and video coding as an alternative to the discrete cosine transform (DCT) [62-64]. Since then, it has gained widespread acceptance in the field of image and video coding applications.

2.1.1 1-D Wavelet Transforms

The discrete wavelet decomposition of a one-dimensional (1-D) continuous signal \( f(t) \) is defined as:

\[
f(t) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} c_{i,j} \psi_{i,j}(t),
\]

where \( c_{i,j} \) is the wavelet coefficient

\[
c_{i,j} = \langle \psi_{i,j}(t), f(t) \rangle = \int_{-\infty}^{\infty} \psi_{i,j}(t)f(t)dt
\]

and \( \psi_{i,j}(t) \) is the wavelet function generated from a mother wavelet \( \psi(t) \) via binary shifting and dyadic translation

\[
\psi_{i,j}(t) = 2^{-i/2}\psi(2^{-i}t - j).
\]

For a 1-D discrete signal \( x[n] \), its one-stage discrete dyadic wavelet decomposition
consists of a low-frequency signal $x_1[n]$ (i.e., approximation subband) and a high-frequency signal $d_1[n]$ (i.e., detail subband):

$$x_1[n] = \sum_k x[k]h[2n - k]$$  \hspace{1cm} (2.1.4)

$$d_1[n] = \sum_k x[k]g[2n - k]$$  \hspace{1cm} (2.1.5)

where $h[n]$ and $g[n]$ are the low-pass and high-pass analysis filters. The length of two subbands, $x_1[n]$ and $d_1[n]$ are half length of the original signal $x[n]$. The original signal $x[n]$ can be reconstructed from $x_1[n]$ and $d_1[n]$ by

$$x[n] = \sum_k x_1[k]\tilde{h}[2n - k] + \sum_k d_1[k]\tilde{g}[2n - k],$$  \hspace{1cm} (2.1.6)

where $\tilde{h}[n]$ and $\tilde{g}[n]$ are the low-pass and high-pass synthesis filters.

In multiple-resolution analysis, such decomposition can be conducted recursively on the approximation subband generated from the previous level. The procedure of 1-D discrete transform with two levels of decomposition is illustrated in Fig. 2.1.

### 2.1.2 2-D Wavelet Transforms

2-D wavelet decompositions are typically achieved by applying the 1-D wavelet transform along rows and columns of the image dataset independently. Figure 2.2 depicts a dyadic decomposition structure for a 2-D image with two levels of decomposition. In the initial stage of decomposition, the original image is decomposed into approximation ($a_1$), horizontal ($h_1$), vertical ($v_1$), and diagonal ($d_1$) subbands, each being
one fourth of the size of the original image. In the second stage of decomposition, only the approximation subband $a_1$ is further decomposed. As shown in Fig. 2.2, subband $a_1$ is split into $a_2$, $h_2$, $v_2$, and $d_2$.

2.1.3 3-D Wavelet Transforms

For 3-D signals, 3-D wavelet transforms are conducted on a GOF, usually 8, 16 or 32 frames. The design issue now is how to extend the above 2-D transform structure to three dimensions to accommodate the addition of the temporal direction. Note that such addition of a third dimension permits several options for the structure of decomposition.

The most straightforward 3-D generalization of the 2-D dyadic structure is 3-D dyadic wavelet transform, as illustrated in Fig. 2.3(a). In essence, this transform is constructed by employing one-scale of a 1-D decomposition separately along the spatial-row, spatial-column, and temporal directions. The approximation subband is then further decomposed to achieve the dyadic decomposition.
Another approach to 3-D wavelet transforms is to use wavelet-packet transforms. One example is given in Fig. 2.3(b), in which a 1-D decomposition is first performed across the frames within one GOF temporally and a 2-D transform is then applied spatially. An advantage of this approach is that we conduct an \( m \)-scale temporal decomposition followed by an \( n \)-scale spatial decomposition. For example, the wavelet-packet transform shown in Fig. 2.3(b) uses a two-level decomposition \( (m = n = 2) \) along all directions.

Compared to the 3-D dyadic wavelet transform, the wavelet-packet decomposition structure is more flexible and typically yields greater coding efficiency. Additionally, it has been shown in [65, 66] that the particular wavelet-packet transform of Fig. 2.3(b) is very close in coding performance to the optimal 3-D transform structure selected in an R-D best-basis sense.
2.2 Wavelet-Based Coding Procedures and Zerotrees

Many wavelet-based 2-D image and 3-D video coders are based on the following important observations that tend to hold true for images and videos in dyadic transforms: (1) since most images and videos in nature have a low-pass spectrum, most signal energy is compacted into lower-frequency subbands; (2) most coefficients are small or even tend to be zero for higher-frequency subbands; (3) small- or zero-valued coefficients tend to be clustered together within a given subband; (4) clusters of small- or zero-valued coefficients tend to be located in the same relative position within subbands of different frequencies.

In wavelet-based coders, the coefficients are usually coded in sign-magnitude form with the signs and magnitudes coded separately. After coefficient signs are coded,
coefficient magnitudes are represented via bitplane coding. The basic idea of such bitplane coding is that the most important bit of all coefficient magnitudes, which yields the largest distortion reduction, is coded first, followed by the next-most important bit, and so forth. For practical implementation, the bitplane coding is accomplished in two coding passes for each bitplane: a significance pass and a refinement pass. In essence, the significance pass performs the significance test through the set of coefficients. If the coefficient is greater or equal to a certain threshold, the coefficient is significant and its most significant nonzero bit is described correspondingly. The refinement pass is then used to generate a successive approximation of each significant coefficient. In addition to coding the significant coefficients, the insignificant coefficients (i.e., the coefficients which are less than the current threshold) also need to be represented in the significance pass. The trick is now how to use dependency between the wavelet coefficients to efficiently encode the insignificant coefficients. To solve this problem, a significance map that contains the insignificant coefficients is built based on such dependency. The significance pass works by successively coding this significance map. It should be noted that the primary difference between wavelet-based coders lies in what dependency is used to build the significance map and how this map coding is performed [20].

Zerotrees are one of the most widely employed schemes for building and coding significance maps in wavelet-based coders. Zerotrees are beneficial from the properties that, for dyadic decompositions of images and videos, insignificant coefficients tend to be clustered together within a given subband and clusters of insignificant coefficients tend to be located in the same relative position within subbands of different
frequencies. Specifically, the dependency used by zerotrees is based on a cross-scale parent-children relationship. For example, a parent coefficient in a subband can be related to children coefficients in the same relative location in a next higher-frequency subband. A zerotree is formed when a coefficient and all of its descendants are insignificant. Several zerotree-based coders include EZW [20], 2-D SPIHT [1] and 3-D SPIHT [2-4,41,67-69].

In addition to zerotrees, different directions have been investigated for wavelet-based coders. An alternative to zerotrees is within-subband coding. For example, [70-73] employ within-subband set partitioning algorithms. In [74-76], a binary set splitting with $k - d$ trees method was demonstrated. In these algorithms, the dependency between the wavelet coefficients is only based on within-subband information and the sets are thus confined to reside within a single subband at all times. This is in contrast to zerotree-based coders, where zerotrees span across multiple transform subbands with different frequencies. Another approach to within-subband coding is to employ extensively conditional, multiple-context adaptive arithmetic coding, such as JPEG 2000 [47-51].

In this work, we focus on 3-D SPIHT, a zerotree-based video coder. Below, we give an overview on 3-D SPIHT and its recent use in domain-partitioning based MD video coding. Before moving onto the review of 3-D SPIHT, we first briefly describe the 2-D SPIHT image coder to introduce necessary terminology.
2.3 2-D SPIHT

As a zerotree-based image coder, 2-D SPIHT [2] made great strides in wavelet-based image coding. It provides excellent R-D performance with low computational cost. In addition, it generates a progressive (or embedded) bitstream in which the information is organized in such a way that the most significant bits are coded first. This means that the transmission of a bitstream can be stopped at any point and the best possible image quality can be obtained by decoding the received incomplete bitstream.

For a given 2-D wavelet transformed image, spatial orientation trees with parent-children relationships were introduced to group the coefficients for the significance test. As illustrated in Fig. 2.4, each node consists of $2 \times 2$ adjacent wavelet coefficients and each coefficient in the node (except those belonging to the lowest and highest frequency subbands) has four children of the same spatial orientation in the next higher-frequency subband. However, this spatial orientation tree is different in the approximation subband: in each node, the top-left coefficient (indicated by a '*' in Fig. 2.4) has no children at all.

After the coefficients are grouped in the form of the spatial orientation tree structure, a natural question arises as to how to efficiently test the magnitudes of the coefficients for significance. In 2-D SPIHT, such significance test is implemented by performing a set-partitioning approach, which capitalizes on the fact that, with a spatial orientation tree, if the value of the parent is insignificant according to some threshold, it is highly likely that its children in the next higher-frequency subband are also insignificant.
In practice, 2-D SPIHT maintains three ordered lists for the coefficients: (1) a list of significant pixels (LSP), (2) a list of insignificant pixels (LIP), and (3) a list of insignificant sets (LIS, i.e., zerotrees). Initially, the LSP is empty, the LIP keeps all the coefficients in the approximation subband, and the LIS keeps all the coefficients in the approximation subband except those which do not have children. The initial threshold is defined as

\[ d = \left\lfloor \log_2(\max\{|W_{x,y}|\}) \right\rfloor, \]

(2.3.7)

where \( W_{x,y} \) is the wavelet coefficient at coordinate \((x, y)\).

2-D SPIHT consists of two passes: sorting and refinement. In the sorting phase, the set-partitioning approach is recursively applied in the LIS and LIP to find the significant coefficients, the insignificant coefficients, and the insignificant subsets and the coordinates are moved to the corresponding lists, the LSP, LIP, and LIS, respectively.
More specifically, the coefficients in the LIP are examined and those which become significant are moved to the LSP. Similarly, the sets in the LIS are also examined. When the set is found to become significant, it is further partitioned into the isolated coefficients and several new subsets. These isolated coefficients are tested again for significance. The significant coefficients are moved to the end of the LSP, while the insignificant coefficients are moved to the end of LIP. The new subsets are added to the end of LIS. In the refinement pass, the coefficients which have been moved to the LSP at higher thresholds are visited and the magnitudes of such coefficients are refined. After the first round of sorting and refinement, the threshold is divided by 2 and the same sorting and refinement phases are applied again until the target bit rate or image quality is achieved.

### 2.4 3-D SPIHT

Inspired by the outstanding performance of 2-D SPIHT for image coding, there has been extensive research on applying 2-D SPIHT for video coding based on the 3-D wavelet transform. In essence, the 3-D SPIHT algorithm is similar to 2-D SPIHT, except it adopts a 3-D spatio-temporal orientation tree structure. Several versions of spatio-temporal tree structures for 3-D SPIHT have been proposed in the literature [3, 4, 41, 67–69, 77, 78]. Two successful versions are a symmetric tree structure [3] and an asymmetric tree structure [41]. In our work, we refer to the 3-D SPIHT using an asymmetric tree structure as AT-SPIHT. In general, the 3-D SPIHT using a symmetric tree structure and AT-SPIHT keep the same simplicity of 2-D SPIHT.
and still provide high performance, full embeddedness, and precise rate control.

Fig. 2.5 illustrates how spatio-temporal tree structures are defined in the 3-D SPIHT using a symmetric tree structure and AT-SPIHT. A GOF is decomposed both in spatial and temporal domains by performing the wavelet-packet transform illustrated in Fig. 2.3(b). Here, we assume that there are sixteen frames in a GOF and the frames are both spatially and temporally decomposed into two levels ($m = n = 2$). As we can see in Fig. 2.5(a), in the symmetric spatio-temporal trees, each node consists of $2 \times 2 \times 2$ adjacent coefficients and each coefficient in the node (except those belonging to the lowest and highest subbands) have eight children in the next higher-frequency level.

The asymmetric spatio-temporal tree structure for AT-SPIHT is shown in Fig. 2.5(b). In the asymmetric spatio-temporal trees, each node consists of $2 \times 2$ adjacent coefficients and the asymmetric tree structure in each coefficient frame is exactly the same as the 2-D case. Recall that the top-left coefficient in each $2 \times 2$ group in the approximation subband has no children in the 2-D case. However, the top-left coefficients of each $2 \times 2$ group in the lowest spatial subband (except those belonging to the highest temporal subband) in the asymmetric spatio-temporal tree have children in the same spatial location of the next higher temporal subband. For example, in Fig. 2.5(b), the top-left coefficient in each $2 \times 2$ group in the lowest spatial subband of temporal $LL_t$ band has one $2 \times 2$ children group in the same location of temporal band $LH_t$. Again, the top-left coefficient in each $2 \times 2$ group in the lowest spatial subband of temporal band $LH_t$ has two $2 \times 2$ children in the same location of temporal band $H$. 

26
Simulation results [3, 5, 41] show that even without motion compensation, the 3-D SPIHT using a symmetric tree structure and AT-SPIHT both provide R-D performance superior to that of MPEG-2 and also comparable to that of H.263. This efficient source coding capability along with many desirable features, such as full embeddedness and low computational cost, makes the 3-D SPIHT using a symmetric tree structure and AT-SPIHT an attractive candidate for video applications. In addition, the simulation results in [41] have shown that AT-SPIHT outperforms the 3-D SPIHT using a symmetric tree structure. It is due to the fact that AT-SPIHT uses a longer tree than its symmetric counterpart and SPIHT as a tree-based coder obtains better R-D performance when the tree depth is longer. Another advantage is that compared to the symmetric spatio-temporal tree structure, the asymmetric
spatio-temporal tree structure can be more easily applied to a different number of decomposition levels in the spatial and temporal dimensions [41].

2.5 STTP-SPIHT and ERC-SPIHT

As discussed above, 3-D SPIHT typically exhibits great R-D performance in addition to simplicity and full embeddedness. However, the produced SPIHT-coded bitstream is extremely sensitive to transmission errors. A single bit transmission error can cause an uncontrolled degradation of the reconstructed video quality. The initial error-free portion of the bitstream up to the point where the transmission error occurs is decodable, while the rest of available bitstream is likely to be undecodable because of the loss of synchronization between encoder and decoder execution paths. Therefore, it is very important to incorporate error resilience into the 3-D SPIHT algorithm to achieve robust video transmission.

Several methods for error-resilient domain-partitioning based MD video coding built upon 3-D SPIHT have been proposed in the literature [37–44]. One such successful method is the STTP-SPIHT algorithm [39]. The main idea of STTP-SPIHT is to use the contiguous grouping method to divide the wavelet coefficients into some number $S$ of different spatio-temporal tree groups. Each group is independently coded and transmitted separately. Consequently, $S$ SPIHT-encoded substreams are generated where each substream corresponds to a specific region of the GOF. The advantage of the STTP-SPIHT algorithm is that errors in one substream do not adversely affect the other substreams. As a result, any decoding failure for one sub-
stream in STTP-SPIHT only affects the associated region in the GOF. This is in contrast to 3-D SPIHT, where the decoding failure affects the full extent of the GOF. The structure of the STTP-SPIHT algorithm is shown in Fig. 2.6. We assume there are sixteen frames in a GOF and the frames are spatially decomposed in one level and temporally decomposed in three levels. In the figure, the wavelet coefficients are divided into four independent groups, denoted by 'a', 'b', 'c', and 'd'. It should be noted that each group can retain either the symmetric spatio-temporal tree structure or the asymmetric spatio-temporal tree structure.

Based on STTP-SPIHT, ERC-SPIHT was proposed in [40, 41]. Compared to STTP-SPIHT, ERC-SPIHT uses dispersive grouping to divide the wavelet coefficients according to their spatial and temporal relationships. The structure of the ERC-SPIHT algorithm is shown in Fig 2.7. Different from STTP-SPIHT, ERC-SPIHT partitions the coefficients in such a way that the spatially adjacent coefficients are spread over different substreams.

The benefits of using a domain-partitioning based MD video coder with 3-D SPIHT can be further amplified when the MD video algorithm is combined with error concealment. In such a context, error concealment exploits the inherent redundancy in the wavelet domain of the signal to estimate missing coefficients from the correctly received ones. By “inherent” redundancy, we mean the perceptual redundancy which is inherent in the signal, not the one intentionally added to the signal. The framework for combining STTP-SPIHT and ERC-SPIHT with error concealment was proposed in [41]. As illustrated in Fig. 2.8, an error concealment technique is applied at the decoder in each individual frame (i.e., spatial domain) right after the inverse temporal
Figure 2.6: Structure of the STTP-SPIHT decomposition using contiguous grouping.
Figure 2.7: Structure of the ERC-SPIHT decomposition using dispersive grouping.
Figure 2.8: Block diagram of domain-partitioning based MD video coding using 3-D SPIHT in conjunction with error concealment.

transform, and the reverse spatial transform is then applied. Note that, due to the properties of the spatio-temporal orientation tree structure, the coefficients encoded in any substream are located in the same spatial location in each frame of coefficients. As a result, the temporal transform does not affect any correctly received coefficients. Since the missing coefficient values (or blocks) are estimated from the available spatially neighboring coefficients, successful concealment performance can be achieved by increase in availability of correctly received spatial neighbors. Using dispersive grouping, ERC-SPIHT typically raises the chance of having a higher number of correctly received spatially adjacent coefficients, thus providing better error concealment performance over STTP-SPIHT. It has been shown in [41] that ERC-SPIHT gives about 1 dB better performance than STTP-SPIHT in the presence of substream loss when error concealment is applied.
2.6 Error Concealment Techniques

As discussed above, error concealment, also referred to as post-processing concealment, can be adopted at the decoder to further improve error resilience. The general idea of error concealment is to estimate the missing information from the correctly received neighboring data, by exploiting the autocorrelation of the signal. These neighboring data can be spatially neighboring data, temporally neighboring data, or a combination of both [17, 79]. Among existing error concealment techniques, some approaches can be integrated into a domain-partitioning based MD video coder. We briefly review three techniques in this section.

2.6.1 Bilinear Interpolation

In bilinear interpolation, the lost coefficient is replaced by the average of its four adjacent neighbors. For example, the missing coefficient \( W_{x,y} \) at the location \((x, y)\) is estimated by

\[
\hat{W}_{x,y} = \frac{1}{4} \{ W_{x-1,y} + W_{x+1,y} + W_{x,y+1} + W_{x,y-1} \}. \quad (2.6.8)
\]

It was demonstrated in [21, 40] that bilinear interpolation can be effectively used to improve the R-D performance. It works fairly well for smooth areas, but artifacts arise near edges.

2.6.2 Locally Adaptive Passive Error Concealment

In [80], locally adaptive passive error concealment (LAPEC) was proposed using a weighted average of the neighboring coefficients. A missing coefficient at location
$(x, y)$ is denoted by $W_{x,y}$. Let $\hat{W}_{x,y}^H$ and $\hat{W}_{x,y}^V$ be the estimates of the coefficient $W_{x,y}$ using horizontal and vertical linear interpolation. That is,

$$\hat{W}_{x,y}^H = \frac{1}{2}\{W_{x,y-1} + W_{x,y+1}\}, \quad (2.6.9)$$

and

$$\hat{W}_{x,y}^V = \frac{1}{2}\{W_{x-1,y} + W_{x+1,y}\}. \quad (2.6.10)$$

The missing coefficient $W_{x,y}$ is then estimated as a weighted average of $\hat{W}_{x,y}^H$ and $\hat{W}_{x,y}^V$.

$$\tilde{W}_{x,y} = \alpha_{x,y}^H \hat{W}_{x,y}^H + \alpha_{x,y}^V \hat{W}_{x,y}^V. \quad (2.6.11)$$

The weights are calculated as

$$\alpha_{x,y}^H = \frac{(\sigma_{x,y}^V)^2}{\{(\sigma_{x,y}^H)^2 + (\sigma_{x,y}^V)^2\}} \quad (2.6.12)$$

and

$$\alpha_{x,y}^V = \frac{(\sigma_{x,y}^H)^2}{\{(\sigma_{x,y}^H)^2 + (\sigma_{x,y}^V)^2\}}, \quad (2.6.13)$$

where

$$(\sigma_{x,y}^H)^2 = \frac{1}{2}\{(W_{x-1,y} - \hat{W}_{x-1,y}^H)^2 + (W_{x+1,y} - \hat{W}_{x+1,y}^H)^2\}, \quad (2.6.14)$$

and

$$(\sigma_{x,y}^V)^2 = \frac{1}{2}\{(W_{x,y-1} - \hat{W}_{x,y-1}^V)^2 + (W_{x,y+1} - \hat{W}_{x,y+1}^V)^2\}. \quad (2.6.15)$$
The results in [80] have demonstrated that LAPEC offers a significant improvement over bilinear interpolation.

2.6.3 Interband Interpolation

The main idea of interband interpolation is that the missing coefficient can be predicted from its parent node in the next-lower frequency subband [57]. The philosophy of the algorithm is following.

Suppose that the estimate $\hat{Y}_n$ of a missing coefficient $Y_n$ is obtained from its parent node $X_{n-1}$ such that

$$\hat{Y}_n = a X_{n-1}. \quad (2.6.16)$$

The optimal weighting vector $a$ can be calculated from the equation

$$\hat{R}_{xx} a = \hat{r}_{yx} \quad (2.6.17)$$

where $\hat{R}_{xx} = \frac{1}{M} H^T H$ and $\hat{r}_{yx} = \frac{1}{M} H^T u$. In practical implementations, four surrounding coefficients around $Y_n$ are taken into consideration ($M = 4$) and we have

$$H = \begin{bmatrix} X_{n-2} \\ X_{n-3} \\ X_{n-4} \\ X_{n-5} \end{bmatrix} \quad (2.6.18)$$
Four scanning orders (i.e., rightward-down, leftward-up, downward-right, and upward-left) are used to form $u$ and their corresponding parent nodes to form $H$. The results are averaged with equal weight to be the final value.

### 2.7 Video Quality Assessment

Video quality can be assessed in two different ways:

- The first is called subjective quality assessment. It consists of human observers who should evaluate the quality of a particular video sequence during tests. Since video quality is a subjective notion, subjective quality assessment can be considered as the most fundamental method of evaluating video quality. However, subjective testing is an inconvenient, time-consuming and expensive process. It usually requires special assessment facilities to produce reliable and reproducible experiment results.

- The second one is called objective quality assessment. Objective video evaluation techniques are mathematical models that can approximate results of subjective quality assessment, but are based on computational metrics that can be automatically evaluated by a computer program. According to the available-
ity of the original video signal, objective methods can be roughly divided into three main categories: (1) full reference methods (FR), (2) reduced reference methods (RR) and (3) no-reference methods (NR). In FR methods, a complete reference video signal is assumed to be known. FR metrics compute the quality difference by comparing every pixel in each frame of the distorted video to its corresponding pixel in the original signal. When the reference video signal is only partially available, RR metrics extract some features of both videos as side information and compare them to evaluate the quality of the distorted video. In contrast to FR and RR methods, NR metrics use only the data from the distorted video to produce an objective quality evaluation without any reference to the original video.

In this work, we adopt subjective and FR objective video quality assessment. Below, we overview some FR based video quality metrics.

### 2.7.1 Peak Signal-to-Noise Ratio

One of the simplest and most widely used FR objective video quality metrics is the peak signal-to-noise ratio (PSNR). It is defined as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \text{dB}$$

where $MSE$ is the mean-squared error between the original and the reconstructed video sequence.
2.7.2 Structural Similarity Index

In addition, we also adopt the structural similarity (SSIM) index to evaluate the quality of the reconstructed video sequence [81]. The SSIM metric is measured between two windows \( x \) and \( y \) in the original and reconstructed video sequence defined as

\[
SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}
\]  

(2.7.21)

where \( \mu_x \) and \( \mu_y \) are the averages of \( x \) and \( y \), \( \sigma_x^2 \) and \( \sigma_y^2 \) are the variances of \( x \) and \( y \), and \( \sigma_{xy} \) is the covariance of \( x \) and \( y \). In this work, both \( x \) and \( y \) are \( 8 \times 8 \) data. The mean SSIM (MSSIM) index is the overall quality measurement derived from the windowed SSIM expressed as

\[
MSSIM = \frac{1}{M} \sum_{i=1}^{M} SSIM_i
\]  

(2.7.22)

where \( M \) is the total number of windows in the video sequence. Note that the PSNR and MSSIM index are both measured on a frame basis in our work.
Chapter 3

Proposed Domain-Partitioning Based Multiple Description Video Coding Algorithm using 3-D SPIHT: A Simple Case Study

This thesis aims to design a domain-partitioning based MD video coder built upon 3-D SPIHT with added redundancy so that the missing coefficients in the spatial root subband can be initially estimated from the inherent redundancy and such estimation can be further refined from the added redundancy. To explain the underlying structure used in our proposed system, in this chapter we focus on a simple scenario, in which four substreams are created ($S = 4$) and only one of the substreams is affected by transmission errors. A similar scenario was also used in [41], in which sixteen
substreams are created \((S = 16)\) and only one of the substreams is affected by noisy channels. In this chapter, we first clarify the concept of spatial root subband. We then introduce the proposed domain-partitioning based MD video coding framework with 3-D SPIHT and several related design issues in detail, such as how to add a certain amount of redundancy among substreams and how to use such additional redundancy to recover the missing coefficients in the spatial root subband. Finally, we conduct some experiments to investigate the performance of the proposed coding algorithm with respect to (1) source coding efficiency and (2) error-resilience.

### 3.1 Spatial Root Subband

Before presenting our proposed MD video coding algorithm, we explain briefly the structure of the spatial root subband, as illustrated by the shaded area in Fig. 3.1, to specify the extent to which our algorithm is applied. In Fig. 3.1(a), we assume there are \(N\) frames in a GOF and the frames are spatially decomposed into one level. Symbol \(\hat{2D}W\) denotes the coefficients resulting from such spatial wavelet transform and symbol \(\hat{2D}WSR\) represents the spatial root subband which is composed of the lowest spatial frequency subband (i.e., the approximation subband on the highest spatial decomposition level in each frame) in the 2-D wavelet domain. As shown in Fig. 3.1(b), such \(N\) frames of coefficients (i.e., \(\hat{2D}W\)) are then temporally transformed in three levels and we define the resulting coefficients as \(\hat{3D}W\). The spatial root subband in the 3-D wavelet domain is denoted as \(\hat{3D}WSR\).

Several domain-partitioning based MD video coders with 3-D SPIHT have been
Figure 3.1: The structure of the spatial root subband defined in our proposed algorithm. (a) 2-D wavelet domain. (b) 3-D wavelet domain.
proposed in [39-41], such as STTP-SPIHT and ERC-SPIHT. As explained in the previous chapters, the benefits of a domain-partitioning based MD video coder with 3-D SPIHT can be further amplified when the video coder is combined with error concealment. In such a context, error concealment techniques exploit the inherent spatial correlation between the substreams to estimate the missing coefficients. Recall that by “inherent” redundancy, we mean the perceptual redundancy inherent in the signal, not the information intentionally added to the signal. It was demonstrated in [21,40] that even simple error concealment techniques, such as bilinear interpolation, can improve the R-D performance when combined with ERC-SPIHT or STTP-SPIHT. However, these techniques still bring some distortion. In particular, among all the wavelet coefficients in each frame, those in the spatial root subband (i.e., $2D W^{SR}$ in the 2-D wavelet domain or $3D W^{SR}$ in the 3-D wavelet domain) typically dominate the signal energy. Small relative errors in the estimation of missing coefficients in the spatial root subband can have a large impact in the overall distortion. Thus, it is necessary to protect such coefficients highly from transmission errors. As to the way to protect such important coefficients, a common approach is to insert additional redundancy into substreams. As a consequence, the missing coefficients in $2D W^{SR}$ due to transmission errors can be successfully recovered from such added redundancy. However, the downside to incorporating such additional redundancy is reduced source coding efficiency. In other words, incorporating such additional redundancy affects two contradictory aspects of the video coding algorithm: its source coding efficiency at the encoder side and the error concealment performance at the decoder side. Therefore, a key to the success of the proposed MD video coder is to balance the tradeoff
between the source coding efficiency and the error concealment performance.

3.2 Proposed Algorithm with Some Design Issues

3.2.1 System Overview

Motivated by these observations, we now propose an error-resilient MD video coding algorithm with 3-D SPIHT. The overall framework of the proposed coding algorithm is shown in Fig. 3.2, where two new functions are added to Cho et al.'s ERC-SPIHT system [41] (as shown in Fig. 2.8), namely additional redundancy generation at the encoder and root subband recovery at the decoder. In addition, we also modify the 3-D SPIHT encoder and decoder in order that they work with the proposed MD video coder. In the framework, we use the same order of spatial and temporal transformation as Cho et al.'s work [41] such that the spatial decomposition precedes the temporal decomposition at the encoder and the spatial reconstruction follows the temporal reconstruction at the decoder. In addition, we adopt AT-SPIHT in our work since it yields better R-D performance than its counterpart, the 3-D SPIHT with a symmetric tree structure. In this chapter, we have not used a model for the noise channel (such as two-state Markov model, etc.) with different bit error rates (BER) and packet loss rates. Similar to [41], we assume that the decoding failure always occurs at a very early stage of a substream and the entire substream is fully affected. Some noisy channel models with different packet loss rates will be incorporated when this work is extended to partial substream loss with packetization.
in Chapter 5. Arithmetic coding can be applied to the SPIHT-coded video substreams for further compression. However, it has been shown in [2] that arithmetic coding provides limited improvement on source coding efficiency while introducing intensive computational cost and possible error propagation. Similar to [57], we have thus dispensed with arithmetic coding in this work.

In general, there are three important design issues associated with our proposed system: (1) how to generate additional redundancy efficiently, (2) how to encode such added redundancy efficiently and resiliently, and (3) how to make the best use of all correctly received information at the decoder side to recover the missing coefficients in the spatial root subband. We will highlight these issues in the following discussion.
3.2.2 Additional Redundancy Generation with an Averaging Approach

Following Fig. 3.2, after a GOF is spatially transformed into the 2-D wavelet transform domain, the next step is to generate the additional redundancy. As explained above, in this chapter we concentrate on a simple scenario, in which four substreams are created \( S = 4 \) and only one of the substreams is affected by noisy channels. Considering the tradeoff between the source coding efficiency and the error concealment performance, we propose to create a new data set as the additional redundancy, denoted by \( 2D^R \). In essence, each element in \( 2D^R \) is obtained by averaging four spatially neighboring coefficients in the spatial root subband \( 2D W^{SR} \). Fig. 3.3 depicts an example of applying such an averaging approach in \( 2D W^{SR} \) on frame \( n \), where \( n = 1, 2, \ldots, N \). We have

\[
2D R_{x,y,n} = \frac{1}{4} \{2D W_{2x+1,2y+1,n} + 2D W_{2x-1,2y-1,n} + 2D W_{2x+1,2y-1,n} + 2D W_{2x-1,2y+1,n}\}, \quad (3.2.1)
\]

where \( 2D W_{x,y,n} \) and \( 2D R_{x,y,n} \) represent the coefficient at the spatial location \((x, y)\) of frame \( n \) in \( 2D W^{SR} \) and \( 2D^R \), respectively. As illustrated in this figure, the resulting averages on frame \( n \) (such as \( 2D R_{x,y,n} \)) are then grouped together in a 3-D structure to form the additional redundancy, denoted by \( 2D^R \) (since we are still in the 2-D wavelet domain at this step).

Continuing with the framework illustrated in Fig. 3.2, next \( 2D W \) and \( 2D^R \) are both temporally transformed into the 3-D wavelet domain and the resulting coefficients are
Figure 3.3: Relationship between original video sequence and additional redundancy generated by an averaging approach.
denoted by $3D W$ and $3D R$, respectively. Note that the relationship in (3.2.1) is still valid for $3D W^{SR}$ and $3D R$ such that

$$3D R_{x,y,n} = \frac{1}{4} \left\{ 3D W_{2x-1,2y-1,n} + 3D W_{2x-1,2y,n} + 3D W_{2x,2y-1,n} + 3D W_{2x,2y,n} \right\} \quad (3.2.2)$$

where $3D W_{x,y,n}$ and $3D R_{x,y,n}$ represent the coefficient at the spatial location $(x, y)$ of frame $n$ in $3D W^{SR}$ and $3D R$, respectively. Therefore, the order in which the additional redundancy generation and temporal transform are applied would not matter in the proposed framework.

### 3.2.3 Domain Partitioning and Encoding

After the wavelet decomposition and redundancy generation, the remaining issue is how to encode the original video sequence coefficients $3D W$ and the additional redundancy $3D R$ efficiently and resiliently. An effective solution is to distribute $3D W$ and $3D R$ onto different substreams. In such a context, the errors in one substream are isolated and do not adversely affect other correctly received substreams to be decoded. Since dispersive grouping partitions the coefficients in such a way that spatially adjacent coefficients are spread over different substreams, we adopt it to divide the coefficients in $3D W$ into $S$ groups ($S = 4$ in this chapter). In practice, this process is usually implemented by two steps: we first apply dispersive grouping on the lowest spatio-temporal subband of $3D W$; we then track the spatio-temporal related trees through all of the spatial and temporal decomposition levels and merge the corresponding coefficients together. Similarly, the coefficients in $3D R$ are also
organized into $S$ groups by using dispersive grouping. Note that the spatio-temporal tree structure in $3D R$ is exactly the same as the one in $3D W$ along the temporal direction. However, the coefficients in $3D R$ have no descendants along the spatial direction since such coefficients are generated by averaging the coefficients in $3D W_{SR}$.

Fig. 3.4 shows an example of applying the above procedure on $3D W$ and $3D R$ to generate the groups $3D W_i$ and $3D R_i$, where $i = 1, \ldots, S$. Each group $3D W_i$ and $3D R_i$ are then encoded with 3-D SPIHT to produce substream-$i$. As a result, we transmit not only the coefficients in $3D W_i$ but also the additional redundancy in $3D R_i$ across the communication channels. However, the conventional 3-D SPIHT coding algorithm takes only one set of coefficients as input. Since our proposed algorithm has coefficients in $3D W_i$ and $3D R_i$ to transmit, we need to either create an encompassing tree structure for the two sets of coefficients or we need to modify the 3-D SPIHT encoder accordingly. One possible solution is to borrow Cho et al.'s work for color video coding [39]. The basic idea is to treat $3D W_i$ and $3D R_i$ of a spatio-temporal block as one unit during the coding stage and generate a mixed $WR$ substream. However, this technique has a disadvantage when it is extended to packet-based networks. Consider assembly of this $WR$ substream into fixed-length packets. Some packets may contain the coefficients not only from $3D W_i$ but also from $3D R_i$. When one of these packets is corrupted by transmission errors, both the $3D W_i$ and $3D R_i$ decoding processes are affected.

For this reason, we propose a modified 3-D SPIHT coder. The modified SPIHT encoder maintains three ordered lists for the coefficients in $3D W_i$: a list of insignificant sets LIS, a list of insignificant pixels LIP and a list of significant pixels LSP. Similarly,
Figure 3.4: Procedure of domain partitioning and SPIHT encoding in the proposed algorithm.
it maintains another three ordered lists for the coefficients in $3^D R_i$: an LIS, an LIP and an LSP. In the initialization stage, the modified 3-D SPIHT encoder first examines the starting threshold derived from the maximum magnitude of the coefficients in group $3^D W_i$ and $3^D R_i$. For notational convenience, let $m_i^W$ be the starting threshold for $3^D W_i$ and $m_i^R$ be the starting threshold for $3^D R_i$. Correspondingly, the highest bitplane in $3^D W_i$ and $3^D R_i$ is $d_i^{\text{max}} = \lfloor \log_2 m_i^W \rfloor$ and $d_i^{\text{max}} = \lfloor \log_2 m_i^R \rfloor$, respectively. The modified encoder then chooses an initial bitplane $d_i^{\text{max}} = \max(d_i^W, d_i^R)$ to start the encoding process. Fig. 3.5 illustrates the segment of the final substream—$i$, where $W$ and $R$ represent the bit information associated with the coefficients in $3^D W_i$ and $3^D R_i$, respectively. For each pass of the encoder at a certain threshold, we first encode the coefficients in $3^D W_i$ and then the coefficients in $3^D R_i$. At the end of the encoding process for that threshold, we divide the current threshold by 2 and move to the next lower bitplane. This procedure is applied until the target bit rate is achieved.

After encoding the coefficients for $3^D W_i$ and $3^D R_i$, the final substream—$i$ is formed by serially concatenating the encoded bits for $3^D W_i$ and $3^D R_i$. To improve the odds of decoding over a noisy channel, we opt to place the encoded bits $R$ for the redundancy, at the end of the final substream in reverse order. By reversing the order, we remove the need for a pointer to the start location of $R$ in substream—$i$, so less SPIHT header information would be required. Compared to [39], the advantage of the proposed SPIHT coder is only having one packet related to some coefficients in $3^D W_i$ and $3^D R_i$ and these coefficients come from the lowest bitplane in $3^D W_i$ and $3^D R_i$ because of the forward order of $3^D W_i$ and the reverse order of $3^D R_i$. The other coefficients in $3^D W_i$ or $3^D R_i$ are arranged in separate packets. As a result, any decoding failure in $3^D W_i$
\[ d_i^w : \text{highest bitplane in } 3D W_i \]

\[ d_i^R : \text{highest bitplane in } 3D R_i \]

Figure 3.5: An example of substream-\( i \) generated by the proposed algorithm with reverse order concatenation of the additional redundancy.

or \( 3D R_i \) does not affect the other as decoding is from the beginning of the substream for \( 3D W_i \) and from the end of the substream for \( 3D R_i \).

For each substream, an amount of the overhead is required as the SPIHT header information. The header information for each substream includes:

- the total number of substreams \( S \) and the substream index \( i \).
- the dimensions of \( 3D W_i \), i.e., \( X \times Y \times GOF \).
- the spatial and temporal decomposition levels.
- the highest bitplane \( d_i^w \) and \( d_i^R \) for substream-\( i \).

Note that such SPIHT header information is necessary for the 3-D SPIHT decoder to decode the substreams correctly. However, it causes a deleterious side effect on the video reconstruction when a large number of substreams \( S \) is generated at the encoder side. This deleterious side effect is even worse especially in the case of lower
bit rates. A possible solution to minimize the deleterious side effect is to use a global header, as suggested in [39]. The objective of using a global header is to decrease the total size of the overhead information and enable the 3-D SPIHT encoder to use more bits to encode video sequence.

Using this idea, we can include the common parameters in the global header, such as the total number of substreams $S$, the dimensions of $3D_W$ (i.e., $X \times Y \times GOF$) and the temporal and spatial decomposition levels. The global header is the most important for the 3-D SPIHT decoder to decode the substreams correctly and thus should be protected from transmission errors. We assume in our work that the global header is always received correctly at the decoder. In addition to the global header, we also use a sub-header which contains the individual parameters for each substream, such as the substream index $i$, and the highest bitplane $d_i^w$ and $d_i^R$.

### 3.2.4 Decoding over Noiseless Channels

When all of the substreams are successfully received at the decoder over noiseless channels, we only need to decode the coefficients related to $3D_W$. We define $3D_W'_{i}$ as the coefficients decoded from the received substream-$i$, where $i = 1, \ldots, S$. We restore the original positions of the wavelet coefficients in $3D_W'_{i}$ to form $3D_W'$. In this case, the coefficients in $3D_W'$ are reconstructed at the full SPIHT encoder accuracy. The reverse temporal transform is first applied to $3D_W'$ and the resulting coefficients are denoted by $2D_W'$. The reverse spatial transform is then applied to $2D_W'$. Note that in the case of a noiseless transmission, the additional redundancy in $2DR$ is not
required and the error concealment step in Fig. 3.2 is skipped. The following section outlines the reconstruction procedure when the channel is noisy and the redundancy in $^{2D}R$ can be used.

### 3.3 Reconstruction of Lost Wavelet Coefficients

We now develop an error concealment technique and describe how we can use it to estimate the missing coefficients in the spatial root subband. As illustrated in Fig. 3.2, such error concealment is applied at the decoder right after the inverse temporal transform. As a consequence, error concealment is deployed in the 2-D wavelet domain.

In this chapter, we focus on a specific scenario, in which four substreams are created ($S = 4$) and only one of the substreams is missing due to transmission error. Recall that, as defined in 3.2.2, each element in $^{3D}R$ is only correlated to a block of $2 \times 2$ adjacent coefficients in $^{3D}W^{SR}$. When dispersive grouping is used, each coefficient in such a block (i.e., $^{3D}W_{2x-1,2y-1,n}$, $^{3D}W_{2x-1,2y,n}$, $^{3D}W_{2x,2y-1,n}$ and $^{3D}W_{2x,2y,n}$) can be dispatched into four different substreams. When one of these four substreams is corrupted by transmission errors, only one coefficient in the block is affected. For example, we assume the decoding error occurred in the beginning of substream-1, so that this substream is completely missing and the coefficient $^{3D}W_{2x-1,2y-1,n}$ is affected.

To further elaborate, let $^{3D}W'_{2x-1,2y-1,n}$, $^{3D}W'_{2x-1,2y,n}$, $^{3D}W'_{2x,2y-1,n}$ and $^{3D}W'_{2x,2y,n}$ be the coefficients decoded from the received substreams. In our case, the coefficients $^{3D}W'_{2x-1,2y,n}$, $^{3D}W'_{2x,2y-1,n}$ and $^{3D}W'_{2x,2y,n}$ are reconstructed from substream-2, 3 and 4.
at the full SPIHT encoder accuracy. Since substream-1 is unavailable at the decoder due to transmission errors, we have to set the coefficient $3^D W'_{2x-1,2y-1,n}$ as zero, that is $3^D W'_{2x-1,2y-1,n} = 0$. The reverse temporal transform is then applied to $3^D W'$ and the resulting coefficients are denoted by $2^D W'_{2x-1,2y-1,n}$, $2^D W_{2x-1,2y,n}$, $2^D W_{2x,2y-1,n}$ and $2^D W_{2x,2y,n}$. The problem is now to make the best use of all correctly received information at the decoder side to reconstruct $2^D W'_{2x-1,2y-1,n}$.

As illustrated in Fig. 3.2, the proposed error concealment technique is comprised of two functions: (i) 2-D error concealment on root and detail subbands using inherent redundancy, and (ii) root subband recovery using added redundancy. The former function is applied to estimate the missing coefficients in the root and detail subbands by exploiting the inherent redundancy present in the signal, while the latter attempts to use the additional redundancy to further improve the accuracy in estimating the missing coefficient $2^D W'_{2x-1,2y-1,n}$. We now describe how the root subband recovery algorithm can be used with existing 2-D error concealment schemes to solve the above problem.

### 3.3.1 2-D Error Concealment

This step can be considered as the initial process in the proposed error concealment technique. Specifically, the missing coefficient $2^D W'_{2x-1,2y-1,n}$ is first estimated by 2-D error concealment, such as bilinear interpolation and LAPEC we use in this chapter.
3.3.2 Root Subband Recovery

Recall that, in addition to $3D W$, the coefficients in $3D R$ are also spread over different substreams. If they are successfully received at the decoder, they can be used to further reconstruct the missing coefficients in $2D W_{SR}$ in this process. For example, if the coefficient $3D R_{x,y,n}$ is transmitted in substream-2, 3 or 4, we define $3D R'_{x,y,n}$ as the coefficient decoded from the successfully received substream and $2D R'_{x,y,n}$ as the resulting coefficient from the reverse temporal transform. In such a context, it is straightforward to reconstruct the lost coefficient $2D W'_{2x-1,2y-1,n}$ as:

$$W^*_{2x-1,2y-1,n} = 4 \cdot 2D R'_{x,y,n} - 2D W'_{2x-1,2y,n} - 2D W'_{2x,2y-1,n} - 2D W'_{2x,2y,n}.$$  \quad (3.3.3)

The success of root subband recovery depends on the rate at which the substreams are encoded. In our case, the coefficients $2D W'_{2x-1,2y,n}$, $2D W'_{2x,2y-1,n}$, $2D W'_{2x,2y,n}$ and $2D R'_{x,y,n}$ are reconstructed from substream-2, 3 and 4 according to the encoding bit rate specified on the encoder side. We can see from (3.2.1) and (3.3.3) that the error between $2D W_{2x-1,2y,n}$ and $2D W'_{2x-1,2y,n}$, between $2D W_{2x,2y-1,n}$ and $2D W'_{2x,2y-1,n}$, between $2D W_{2x,2y,n}$ and $2D W'_{2x,2y,n}$, as well as between $2D R_{x,y,n}$ and $2D R'_{x,y,n}$ would naturally affect the accuracy in estimating the missing coefficient $2D W'_{2x-1,2y-1,n}$. For example, the SPIHT decoding process stops right after the refinement pass at the $d^{th}$ bit plane. The error between the real value of the coefficient and the reconstructed value is $|e| < 2^d$. Thus, such error gradually decreases as we use higher bit rate (i.e. smaller $d$), which will gradually decrease the distortion in the estimation of $2D W'_{2x-1,2y-1,n}$ in (3.3.3).
However, not all of the missing coefficients in the spatial root subband $2D W^{SR}$ can be recovered by this process. For example, $3D R'_{x,y,n}$ is also lost due to transmission errors when it is transmitted in substream-1. In this case, we can only utilize the existing error concealment techniques (e.g., bilinear interpolation and LAPEC) to estimate the missing coefficient $2D W^{'}_{2x-1,2y-1,n}$. For example, we consider a specific scenario, in which four substreams are created ($S = 4$) and only substream-1 is affected by noisy channels. In such a context, most of the missing coefficients in $2D W^{SR}$ can be further refined by the proposed root subband recovery method in (3.3.3) and only a small number of the missing coefficients in $2D W^{SR}$ are estimated using 2-D error concealment. This is in contrast to ERC-SPIHT, where all of the missing coefficients are recovered by 2-D error concealment. Fig. 3.6 graphically illustrates the comparison of the missing coefficients in the spatial root subband which are only estimated by existing 2-D error concealment techniques. To simplify the illustration, we assume that the size of the spatial root subband on an individual frame is $8 \times 8$. In this figure, the black pixels represent the missing coefficients which are only approximated by 2-D error concealment. Fig. 3.6(a) shows that there are up to sixteen missing coefficients in ERC-SPIHT when one substream is lost. Compared to ERC-SPIHT, only four missing coefficients are only estimated by 2-D error concealment in our proposed video coding algorithm, as illustrated in Fig. 3.6(b); other missing coefficients in the spatial root subband can be further refined by the root subband recovery method using (3.3.3), which will improve the quality of the reconstructed video.
Figure 3.6: Comparison of positions of the missing coefficients that are estimated using 2-D error concealment in the spatial root subband of size $8 \times 8$. (a) ERC-SPIHT. (b) Our proposed MD video coding algorithm.

3.4 Simulation Results

In this section, we conduct experiments to investigate the performance of the proposed MD video coding algorithm in noiseless and noisy channels. We start with an introduction of the simulation parameters. We then present the simulation results in noiseless and noisy channels and compare the proposed MD video coder to ERC-SPIHT [40,41] with respect to (1) source coding efficiency and (2) error concealment performance.

3.4.1 Experimental Data

In this work, the encoders and the corresponding decoders (such as ERC-SPIHT and the proposed method) with different error concealment techniques are implemented in Matlab. The original video sequence is first encoded and stored in Matlab files. These files are then decoded and refined to produce the reconstructed video sequence.

The experiments are conducted using the $352 \times 240 \times 48$ monochrome “Football”
and "Susie" (frame number 16–63) video sequences with frame rate of 30 frames/s. These video sequences are characterized by different motion properties. The "Football" sequence is considered to have a relatively high level of motion, while the "Susie" sequence has a relatively low level of motion. We use 16 frames for the GOF since GOF of 16 provides better R-D performance than smaller GOFs [5]. A three-level decomposition using the biorthogonal Cohen-Daubechies-Feauveau (CDF) 9/7 wavelet filter [62] is applied in both the spatial and temporal domains with reflection extensions both at each image boundary and each GOF boundary. In this chapter, the wavelet coefficients and additional redundancy are divided into four groups ($S = 4$). Each group is independently encoded at a coding rate $c$, which is 1.0 bit/pixel (bpp) or 0.5 bpp. Correspondingly, the total transmission rate is $352 \times 240 \times 30 \times 1.0 = 2.53$ Mbps or $352 \times 240 \times 30 \times 0.5 = 1.27$ Mbps, respectively. We use one global header for all the substreams and one sub-header for each substream. The PSNR defined in (2.6.8) is used as the distortion metric to evaluate the reconstructed video sequence.

### 3.4.2 Source Coding Efficiency

To compare the source coding efficiency of the proposed MD video coding algorithm to that of ERC-SPIHT, we assume that the channels are error-free. Thus, four substreams are received correctly at the decoder. Figs. 3.7 and 3.8 show the comparison of frame by frame PSNR values of the "Football" and "Susie" sequences when $c = 0.5$ bpp and $c = 1.0$ bpp. In these figures, the solid lines represent the PSNR values for
Figure 3.7: Comparison of frame by frame PSNR (dB) of “Football” video sequence in noiseless channels at different coding rates $c$ ($S = 4$ substreams).

The proposed algorithm, while the dotted lines illustrate the PSNR values for ERC-SPIHT. As we can see, there are small losses (0.11–0.25 dB at $c = 0.5$ bpp; 0.11–0.22 dB at $c = 1.0$ bpp) on the “Football” sequence and (0.05–0.28 dB at $c = 0.5$ bpp; 0.06–0.18 dB at $c = 1.0$ bpp) on the “Susie” sequence with the proposed method compared to ERC-SPIHT. These small PSNR losses result from the additional redundancy overhead that is inserted in the substreams.

The average PSNR values across the frames in Figs. 3.7 and 3.8 for the proposed method and ERC-SPIHT are summarized in Table 3.1. Also, the average PSNR values obtained by the proposed method are slightly lower than those of ERC-SPIHT. Specifically, the PSNR differences range from 0.15 to 0.17 dB on the “Football” sequence and from 0.12 to 0.17 dB on the “Susie” sequence with the proposed method compared to ERC-SPIHT.
Figure 3.8: Comparison of frame by frame PSNR (dB) of “Susie” video sequence in noiseless channels at different coding rates $c$ ($S = 4$ substreams).

Table 3.1: Comparison of average PSNR (dB) of “Football” and “Susie” video sequences in noiseless channels at different coding rates $c$ ($S = 4$ substreams).

(a) Football

<table>
<thead>
<tr>
<th>$c$ (bpp)</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>29.35</td>
<td>29.18</td>
</tr>
<tr>
<td>1.0</td>
<td>33.35</td>
<td>33.20</td>
</tr>
</tbody>
</table>

(b) Susie

<table>
<thead>
<tr>
<th>$c$ (bpp)</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>40.07</td>
<td>39.90</td>
</tr>
<tr>
<td>1.0</td>
<td>43.87</td>
<td>43.75</td>
</tr>
</tbody>
</table>
3.4.3 Error Concealment Performance

We compare the R-D performance of the proposed MD video coding algorithm to that of ERC-SPIHT in noisy channels to investigate the error resilience feature. We assume that there is only one substream corrupted because of transmission errors and other substreams are received correctly at the decoder. We also assume that the global header is not corrupted from transmission errors. In this chapter, we use two error concealment techniques (bilinear interpolation and LAPEC). We focus on recovering the missing coefficients in the spatial root subband $2D W^{SR}$ and we set the missing coefficients in other subbands to zero.

Figs. 3.9 and 3.10 illustrate the comparison of frame by frame PSNR values for the “Football” and “Susie” sequences coded to $c = 0.5$ bpp and 1.0 bpp in noisy channels by the proposed method and ERC-SPIHT. We assume that substream-2 or substream-4 is corrupted due to transmission errors. In these experiments, we use bilinear interpolation to estimate the missing coefficients in $2D W^{SR}$. In contrast to ERC-SPIHT, a large number of the missing coefficients can be further refined using (3.3.3) in the proposed algorithm. As a result, the proposed method improves the reconstructed video quality at the decoder significantly for both low coding rate ($c = 0.5$ bpp) and high coding rate ($c = 1.0$ bpp). For example, we can observe that the PSNRs of the proposed algorithm are much higher than those of ERC-SPIHT, ranging from 0.03 to 0.95 dB ($c = 0.5$ bpp) and from 0.66 to 1.37 dB ($c = 1.0$ bpp) on the “Football” sequence. In the case of “Susie” sequence, the proposed algorithm outperforms ERC-SPIHT by 0.60–1.60 dB ($c = 0.5$ bpp) and 0.63–1.69 dB ($c = 1.0$
Table 3.2: Comparison of average PSNR (dB) of “Football” and “Susie” video sequences in noisy channels with bilinear interpolation at different coding rates $c$ ($S = 4$ substreams and one substream is lost).

(a) *Football*

<table>
<thead>
<tr>
<th></th>
<th>$c$ (bpp)</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substream-2 is lost</td>
<td>0.5</td>
<td>24.26</td>
<td>24.85</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>24.93</td>
<td>25.93</td>
</tr>
<tr>
<td>Substream-4 is lost</td>
<td>0.5</td>
<td>24.16</td>
<td>24.75</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>24.81</td>
<td>25.84</td>
</tr>
</tbody>
</table>

(b) *Susie*

<table>
<thead>
<tr>
<th></th>
<th>$c$ (bpp)</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substream-2 is lost</td>
<td>0.5</td>
<td>29.53</td>
<td>30.45</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>29.71</td>
<td>30.73</td>
</tr>
<tr>
<td>Substream-4 is lost</td>
<td>0.5</td>
<td>29.14</td>
<td>30.30</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>29.31</td>
<td>30.57</td>
</tr>
</tbody>
</table>

Table 3.2 shows the average PSNR values for the proposed algorithm and ERC-SPIHT of Figs. 3.9 and 3.10. The average PSNR improvement obtained by the proposed method is about 0.58–1.03 dB on the “Football” sequence and 0.92–1.26 dB on the “Susie” sequence. In addition, the amount of performance improvement is more noticeable at a higher source coding rate ($c = 1.0$ bpp). Specifically, such performance improvement in the estimation of the missing coefficients in $2DWSR$ is increased by up to 0.44 dB on the “Football” sequence and 0.11 dB on the “Susie” sequence when we increase the coding rate $c$ from 0.5 to 1.0 bpp.

The comparison of frame by frame PSNR values obtained by the proposed algorithm and ERC-SPIHT with LAPEC is given in Figs. 3.11 and 3.12. The average PSNR values are summarized in Table 3.3. Similar to Figs. 3.9 and 3.10, the proposed algorithm enhances the reconstructed video quality at the decoder significantly.
Figure 3.9: Comparison of frame by frame PSNR (dB) of “Football” video sequence in noisy channels with bilinear interpolation at different coding rates $c$ ($S = 4$ substreams and one substream is lost). (a) Substream-2 is lost. (b) Substream-4 is lost.
Figure 3.10: Comparison of frame by frame PSNR (dB) of “Susie” video sequence in noisy channels with bilinear interpolation at different coding rates $c$ ($S = 4$ substreams and one substream is lost). (a) Substream-2 is lost. (b) Substream-4 is lost.
Specifically, the performance differences range from 0.02 to 1.02 dB \((c = 0.5 \text{ bpp})\) and from 0.57 to 1.23 dB \((c = 1.0 \text{ bpp})\) on the “Football” sequence, and in the case of “Susie” sequence the differences are 0.42–1.32 dB \((c = 0.5 \text{ bpp})\) and 0.45–1.40 dB \((c = 1.0 \text{ bpp})\). Similar to Table 3.2, such performance gain achieved by the proposed method is more significant when \(c = 1.0 \text{ bpp}\). For example, the PSNR improvement is increased by up to 0.43 dB on the “Football” sequence and 0.10 dB on the “Susie” sequence when the coding rate \(c\) is increased from 0.5 to 1.0 bpp. In addition, LAPEC yields a slightly better reconstruction performance than bilinear interpolation when combined with the proposed method. We can observe from Tables 3.2 and 3.3 that the PSNRs of the proposed method with LAPEC are 0.01–0.03 dB higher than those of the proposed method with bilinear interpolation on the “Football” sequence and 0.09–0.12 dB higher on the “Susie” sequence. Such difference results from the fact that a small number of the missing coefficients in \(2DWSR\) are only estimated by 2-D error concealment. In such a context, LAPEC typically offers a better PSNR performance than bilinear interpolation.

Figs. 3.13 and 3.14 show sample frames to compare the visual results obtained in Figs. 3.9–3.12. As we can see, in the case of no error concealment, there are many black spots in the frames. When the missing coefficients are estimated in ERC-SPIHT by using bilinear interpolation or LAPEC (Fig. 3.13(c)-(d) and Fig. 3.14(c)-(d)), some missing areas are not recovered successfully and these spots are marked by rectangles. Fig. 3.13(e)-(f) and Fig. 3.14(e)-(f) show that the reconstruction of these spots are improved in the proposed algorithm. Compared to Fig.3.13(e) and Fig. 3.14(e) with bilinear interpolation applied, the proposed algorithm based on LAPEC
Figure 3.11: Comparison of frame by frame PSNR (dB) of “Football” sequence in noisy channels with LAPEC at different coding rates $c$ ($S = 4$ substreams and one substream is lost). (a) Substream-2 is lost. (b) Substream-4 is lost.
Figure 3.12: Comparison of frame by frame PSNR (dB) of “Susie” sequence in noisy channels with LAPEC at different coding rates $c$ ($S = 4$ substreams and one substream is lost). (a) Substream-2 is lost. (b) Substream-4 is lost.
Table 3.3: Comparison of average PSNR (dB) of “Football” and “Susie” video sequences in noisy channels with LAPEC at different coding rates $c$ ($S = 4$ substreams and one substream is lost).

(a) Football

<table>
<thead>
<tr>
<th>$c$ (bpp)</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substream-2 is lost</td>
<td>0.5</td>
<td>24.32</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>25.00</td>
</tr>
<tr>
<td>Substream-4 is lost</td>
<td>0.5</td>
<td>24.20</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>24.86</td>
</tr>
</tbody>
</table>

(b) Susie

<table>
<thead>
<tr>
<th>$c$ (bpp)</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substream-2 is lost</td>
<td>0.5</td>
<td>29.76</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>29.95</td>
</tr>
<tr>
<td>Substream-4 is lost</td>
<td>0.5</td>
<td>29.38</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>29.55</td>
</tr>
</tbody>
</table>

yields a better visual performance in Fig. 3.13(f) and Fig. 3.14(f). Such difference agrees with the objective PSNR evaluation.

3.5 Conclusions

In this chapter, we present an error-resilient MD coding algorithm based on ERC-SPIHT for robust video transmission. In our proposed algorithm, the added redundancy is obtained by averaging four spatially neighboring coefficients in the spatial root subband $2D W^{SR}$ and such additional redundancy is then intentionally inserted into the substreams. As a result, the missing coefficients in $2D W^{SR}$ can be reconstructed from the added redundancy by using the proposed root subband recovery method. We can see from the simulation results that the proposed video coder provides lower source coding efficiency than ERC-SPIHT in lossless transmission but it is more resilient in an error-prone transmission environment. However, the techniques
Figure 3.13: 352 × 240 “Football” sequence \((S = 4\ \text{substreams} \text{ and } c = 1.0 \text{ bpp})\). (a) Original sequence (frame 47). (b) Corrupted sequence when substream-4 is missing: \(PSNR = 11.93\ \text{dB}\). (c) ERC-SPIHT with bilinear interpolation: \(PSNR = 23.96\ \text{dB}\). (d) ERC-SPIHT with LAPEC: \(PSNR = 24.07\ \text{dB}\). (e) Proposed algorithm with bilinear interpolation: \(PSNR = 24.61\ \text{dB}\), (f) Proposed algorithm with LAPEC: \(PSNR = 24.63\ \text{dB}\).
Figure 3.14: 352 × 240 “Susie” sequence ($S = 4$ substreams and $c = 1.0$ bpp). (a) Original sequence (frame 27). (b) Corrupted sequence when substream-2 is missing: $PSNR = 13.47$ dB. (c) ERC-SPIHT with bilinear interpolation: $PSNR = 29.14$ dB. (d) ERC-SPIHT with LAPEC: $PSNR = 29.48$ dB. (e) Proposed algorithm with bilinear interpolation: $PSNR = 30.10$ dB. (f) Proposed algorithm with LAPEC: $PSNR = 30.22$ dB.
used for additional redundancy generation and root subband recovery in this chapter is restricted to a specific scenario (i.e., four substreams are created and only one of the substreams is lost due to noisy channels). The challenge now is how to generalize these techniques for the generation of an arbitrary number of substreams $S$ and extend the recovery of missing coefficients. We will work on this topic in the next chapter.
Chapter 4

Wavelet Projection Based Additional Redundancy Generation and Error Concealment Techniques: Full Substream Loss

Motivated by the success of the preliminary work in Chapter 3, we are now interested in generalizing the proposed domain-partitioning based MD video coder for the generation of an arbitrary number $S$ of substreams. In this chapter we consider the full substream loss case, in which we assume that the decoding failure always occurs at a very early stage of a substream and all the coefficients in that substream are affected. To achieve such generalization in the full substream loss case, we use the proposed framework presented in Fig. 3.2, but with different strategies for implementing the
additional redundancy generation method used at the encoder and the error concealment technique applied at the decoder. In this chapter we begin by explaining a new additional redundancy generation method. We then describe how such additional redundancy can be used to reconstruct the missing coefficients. Some experimental results are also reported to explore the performance of the proposed coding algorithm with respect to (1) source coding efficiency and (2) error-resilience.

4.1 Additional Redundancy Generation using 2-D Wavelet Decomposition

Following Fig. 3.2, after a GOF is spatially transformed into the 2-D wavelet transform domain, the next step is to generate the additional redundancy. In essence, the additional redundancy is generated as described in Fig. 4.1. To simplify the explanation, in this figure we assume there are \( N \) frames in a GOF and the frames are spatially transformed by only one wavelet decomposition level. The notation \( \mathcal{W} \) denotes the wavelet coefficients resulting from the spatial wavelet transform and the notation \( \mathcal{W}_{SR} \), shown by the shaded area in Fig. 4.1, represents the spatial root subband. The additional redundancy is obtained by a wavelet projection based method: each frame of the spatial root subband \( \mathcal{W}_{SR} \) is spatially decomposed by one additional transform level and the resulting coefficients in the approximation subband are grouped together. Fig. 4.1 depicts an example of decomposing the spatial root subband \( \mathcal{W}_{SR} \) by one level on frame \( n \), where \( n = 1, 2, \ldots, N \). This 1-level
Figure 4.1: Relationship between original video sequence and additional redundancy generated by a 2-D wavelet decomposition.

The wavelet transform of $2D^W^{SR}$ on frame $n$ can be represented as

$$2D^W_n^{SR} \rightarrow \begin{bmatrix} a_n & h_n & v_n & d_n \end{bmatrix}.$$ (4.1.1)

The coefficients in subband $a_n$, $h_n$, $v_n$, and $d_n$ can be computed as

$$a_n : a_{x,y,n} = \sum_i \sum_j h_i h_j^2D^W_{n}^{SR}(2x-i,y)^2D^W_{n}^{SR}(x,2y-j)$$

$$h_n : h_{x,y,n} = \sum_i \sum_j g_i g_j^2D^W_{n}^{SR}(2x-i,y)^2D^W_{n}^{SR}(x,2y-j)$$ (4.1.2)

$$v_n : v_{x,y,n} = \sum_i \sum_j h_i g_j^2D^W_{n}^{SR}(2x-i,y)^2D^W_{n}^{SR}(x,2y-j)$$

$$d_n : d_{x,y,n} = \sum_i \sum_j g_i g_j^2D^W_{n}^{SR}(2x-i,y)^2D^W_{n}^{SR}(x,2y-j).$$
where \( h \) and \( g \) are low-pass and high-pass filter coefficients, respectively. Then, the resulting coefficients in subband \( a_n \) (i.e., approximation coefficients in the lowest frequency subband) are grouped together in a 3-D structure as \( \{a_1; a_2; \ldots; a_N\} \) to form the additional redundancy, denoted by \( 2^D R \) (since we are still in the 2-D wavelet domain at this step).

Note that a wide range of wavelet transform filters could be chosen at this step; the wavelet filter at the additional redundancy generation step can be different from the one used when decomposing the GOF into the 2-D (and then 3-D) wavelet domain. For example, we use the Haar wavelet transform and the CDF 9/7 wavelet filter in this work. One reason to choose the Haar wavelet is that computation is simpler and the results are easier to analyze mathematically. Another nice feature of the Haar wavelet is that it is a 2-tap filter. Recall that the coefficients of the Haar low-pass filter are \( h = \left[ \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \) and the coefficients of the Haar high-pass filter are \( g = \left[ \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}} \right] \).

Let \( 2^D R_{x,y,n} \) and \( 2^D W_{x,y,n} \) denote the coefficient at the spatial location \((x, y)\) of frame \( n \) in \( 2^D R \) and \( 2^D W_{SR} \), respectively. Using the Haar wavelet transform, from (4.1.2) we find that

\[
2^D R_{x,y,n} = a_{x,y,n} = h \begin{bmatrix} 2^D W_{2x-1,2y-1,n} & 2^D W_{2x,2y-1,n} \\ 2^D W_{2x-1,2y,n} & 2^D W_{2x,2y,n} \end{bmatrix} h^T \\
= \frac{1}{2} \left\{ 2^D W_{2x-1,2y-1,n} + 2^D W_{2x-1,2y,n} + 2^D W_{2x,2y-1,n} + 2^D W_{2x,2y,n} \right\}. \quad (4.1.3)
\]

As a result, each element in \( 2^D R \) is only correlated to a block of \( 2 \times 2 \) adjacent wavelet coefficients in \( 2^D W_{SR} \), which can simplify the mathematical analysis as discussed
further in Sec. 4.2.3. Recall that in our previous work in Chapter 3, the additional redundancy $2^D R$ is obtained by averaging four spatially neighboring coefficients in the spatial root subband $2^D W^{SR}$ such that

$$2^D R_{x,y,n} = \frac{1}{4} \left\{ 2^D W_{2x-1,2y-1,n} + 2^D W_{2x-1,2y,n} + W_{2x,2y-1,n} + 2^D W_{2x,2y,n} \right\}. \quad (4.1.4)$$

Compared to (4.1.3), we can find that the method used in Chapter 3 is just a special case of the Haar based 2-D wavelet decomposition. The only difference is that each element in $2^D R$ is half the magnitude of the one in (4.1.3).

In addition to the Haar transform, we also experiment with the CDF 9/7 biorthogonal wavelet filter to generate the additional redundancy in our work. Compared to the 2-tap Haar filter, CDF 9/7 is a longer filter with up to 9 taps and typically offers better performance on image and video coding. Similar to (4.1.3), the relationship between $2^D W$ and $2^D R$ for the CDF 9/7 filter is given as

$$2^D R_{x,y,n} = a_{x,y,n} = h \begin{bmatrix} 2^D W_{2x-5,2y-5,n} & \cdots & 2^D W_{2x+3,2y-5,n} \\ \vdots & \ddots & \vdots \\ 2^D W_{2x-1,2y-1,n} & \cdots & 2^D W_{2x+3,2y+3,n} \\ 2^D W_{2x-5,2y+3,n} & \cdots & 2^D W_{2x+3,2y+3,n} \end{bmatrix} h^T, \quad (4.1.5)$$

where $h = [h_1 \cdots h_9]$ is the vector of CDF 9/7 low-pass filter coefficients.

Continuing with the framework illustrated in Fig. 3.2, next $2^D W$ and $2^D R$ are
both temporally transformed to the 3-D wavelet domain and the resulting coefficients are denoted by $3D W^{SR}$ and $3D R$, respectively. The relationships in (4.1.3) and (4.1.5) are still valid for $3D W$ and $3D R$ since Haar and CDF 9/7 are separable transforms.

We have

$$3D R_{x,y,n} = \frac{1}{2} \{ 3D W_{2x-1,2y-1,n} + 3D W_{2x-1,2y,n} + 3D W_{2x-1,2y-1,n} + 3D W_{2x,2y-1,n} \}, \quad (4.1.6)$$

and

$$3D R_{x,y,n} = h \begin{bmatrix}
3D W_{2x-5,2y-5,n} & \cdots & 3D W_{2x+3,2y-5,n} \\
\vdots & \ddots & \vdots \\
3D W_{2x-5,2y+3,n} & \cdots & 3D W_{2x+3,2y+3,n}
\end{bmatrix} h^T. \quad (4.1.7)$$

After the wavelet decomposition and redundancy generation, we follow the same procedure to encode $3D W$ and $3D R$ as introduced in Chapter 3. For example, we use dispersive grouping to divide the coefficients in $3D W$ and $3D R$ into $S$ groups. Each group is then encoded by the proposed SPIHT coder and the additional redundancy is added at the end of each substream in reverse order.

When all of the substreams are successfully received at the decoder over noiseless channels, the additional redundancy $2D R$ is not required and we only need to decode the coefficients related to $3D W$. We define $3D W'$ as the coefficients reconstructed at
the decoder. In the noiseless case, the coefficients in $3^D W'$ are reconstructed at the full SPIHT encoder accuracy. The reverse temporal transform is first applied to $3^D W'$ and the resulting coefficients are denoted by $2^D W'$. The reverse spatial transform is then applied to $2^D W'$ to produce the reconstructed video sequence. Note that in the case of noiseless transmission, the error concealment step in Fig. 3.2 can be skipped. In the following section we describe the error concealment procedure when the channel is noisy and the reconstructed redundancy $2^D R'$ can be used.

4.2 Proposed Error Concealment Technique for Full Substream Loss

The aim of our error concealment technique is to estimate the missing coefficients from the correctly received neighboring coefficients and the additional redundancy. As illustrated in Fig. 3.2, error concealment is applied at the decoder right after the inverse temporal transform. As a consequence, error concealment is deployed in the 2-D wavelet domain.

In this chapter, error concealment is comprised of two functions: (i) 2-D error concealment on root and detail subbands, and (ii) root subband recovery using redundant information. The former function is applied to estimate the missing coefficients in the root and detail subbands by exploiting the inherent redundancy present in the signal, while the latter attempts to further improve the accuracy in estimating the missing coefficients in $2^D W^{SR}$. We now describe how the root subband recovery algorithm can
be used with existing 2-D error concealment schemes in each frame. In this chapter we concentrate on the full substream loss case, in which we assume that when an error occurs in any substream, all coefficients in that substream are affected and we set them to zero.

4.2.1 2-D Error Concealment

This step can be considered as the initial process in the proposed error concealment technique. Specifically, the missing coefficients in the spatial root subband are first estimated by 2-D error concealment. Although the dispersive grouping method minimizes the probability that the neighbors of the missing coefficient are also lost, this situation can still occur, especially at a high loss rate. For LAPEC, the loss of neighbors decreases the reconstruction quality as the estimates of $\alpha_{x,y}^H$ and $\alpha_{x,y}^V$ are not optimal [80]. Due to this reason, we adopt bilinear interpolation in this chapter. When the neighbors of the missing coefficient are lost, they are disregarded and are not taken into the average. If all neighbors of the missing coefficient are lost, the missing coefficient is recovered as: the neighbors are first estimated with bilinear interpolation and the estimated neighbors are then taken into account to reconstruct estimates of the missing coefficient. In addition to bilinear interpolation, the missing coefficients are then refined by the proposed root subband recovery.
4.2.2 Iterative Root Subband Recovery

The root subband recovery method is motivated by the fact that the same relationship in (4.1.2) is supposed to still exist between the spatial root subband in $2^D W'$ and $2^D R'$ on each frame when all of the substreams are correctly received at the decoder, and we can write

$$2^D R'_{x,y,n} \approx a'_{x,y,n} = \sum_i \sum_j h_i h_j^{2^D W_n^{SR}} (2x - i, y)^{2^D W_n^{SR}} (x, 2y - j).$$

(4.2.8)

Note that the approximation in (4.2.8) is due to any bitstream truncation that may occur during encoding/transmission of the substream for a target bit rate, and the approximation would tend to improve as the bit rate is increased. For example, suppose the decoding process stops right after the bit plane $d$. The error between the actual value and the reconstructed value is $|\epsilon| < 2^d$. Notice that such error is gradually decreased as we use a higher bit rate (i.e., smaller $d$). Therefore, when we increase the bit rate, we assume that the difference between $2^D W$ and $2^D W'$ and between $2^D R$ and $2^D R'$ is small enough to be ignored.

The proposed root subband recovery approach is illustrated in Fig. 4.2. Recall that root subband recovery is implemented on each individual frame after temporal reconstruction. Starting with the top-left 4 x 4 block of Fig. 4.2, an example root subband for a single frame $n$ is shown with a missing coefficient shown in black at location $(x, y)$, i.e., $2^D W'_{x,y,n}$. Also in this example illustration, a redundancy coefficient at $2^D R'_{x,y,n}$ is also missing, shown in black in the top-right 2 x 2 block of Fig. 4.2. As discussed above, the missing coefficient $2^D W'_{x,y,n}$ is first estimated as
\(2D \hat{W}_{x,y,n}\) by using 2-D error concealment (e.g., bilinear interpolation). To differentiate from the noiseless case, the notation \(2D W'_n\) and \(2D R'_n\) are now denoted as \(2D \hat{W}_n\) and \(2D \hat{R}_n\), respectively. To further elaborate, the spatial root subband of \(2D \hat{W}_n\) is denoted as \(2D \hat{W}^{SR}_n\). To simplify the illustration, we assume that the size of \(2D \hat{W}^{SR}_n\) is \(4 \times 4\) in Fig. 4.2. Correspondingly, the additional redundancy \(2D \hat{R}_n\) is of size \(2 \times 2\). Similar to (4.1.2), we have the \(\hat{a}_n\), \(\hat{h}_n\), \(\hat{v}_n\) and \(\hat{d}_n\) subbands with wavelet decompositions as follows:

\[
\begin{align*}
\hat{a}_n : \hat{a}_{x,y,n} &= \sum_i \sum_j h_i h_j 2D \hat{W}^{SR}_n(2x - i, y)2D \hat{W}^{SR}_n(x, 2y - j) \\
\hat{h}_n : \hat{h}_{x,y,n} &= \sum_i \sum_j g_i h_j 2D \hat{W}^{SR}_n(2x - i, y)2D \hat{W}^{SR}_n(x, 2y - j) \\
\hat{v}_n : \hat{v}_{x,y,n} &= \sum_i \sum_j h_i g_j 2D \hat{W}^{SR}_n(2x - i, y)2D \hat{W}^{SR}_n(x, 2y - j) \\
\hat{d}_n : \hat{d}_{x,y,n} &= \sum_i \sum_j g_i g_j 2D \hat{W}^{SR}_n(2x - i, y)2D \hat{W}^{SR}_n(x, 2y - j).
\end{align*}
\] (4.2.9)

From (4.2.8), we know that the additional redundancy \(2D \hat{R}_n\) is approximately derived from the real values of \(2D W'_n\) at a high enough bit rate. Thus, the corrupted subband \(\hat{a}_n\) can be replaced with \(2D \hat{R}_n = \hat{a}'_n\). This replacement of \(\hat{a}_n\) is illustrated on the right side of Fig. 4.2 where first a 2-D wavelet decomposition is done on the spatial root subband of \(2D \hat{W}_n\) (i.e., \(2D \hat{W}^{SR}_n\)) to form the greyed result on the right (greyed because the missing coefficient affects all coefficients in the wavelet decomposition).

Then, the correctly received additional redundancy coefficients in \(2D \hat{R}_n\) shown in white are inserted into the \(\hat{a}_n\) block matrix as shown in the bottom-right \(4 \times 4\) block of Fig. 4.2. It should be emphasized that some coefficients of \(2D \hat{R}_n\) may also be lost due to transmission errors, so the coefficient at location \((x,y)\) in the subband \(\hat{a}_n\) is
kept unchanged (still shown in gray in the bottom-right 4 × 4 block of Fig. 4.2).

Next, an inverse wavelet transform is applied (refined \( \hat{a}_n, \hat{h}_n, \hat{v}_n \) and \( \hat{d}_n \)) with a dual low-pass filter \( \tilde{h} \) and a dual high-pass filter \( \tilde{g} \) as below:

\[
2D\hat{W}_{x,y,n} = \sum_{i} \sum_{j} \hat{h}_i \tilde{h}_j Z(\hat{a}_n) + \sum_{i} \sum_{j} \tilde{g}_i \tilde{h}_j Z(\hat{h}_n) + \sum_{i} \sum_{j} \hat{h}_i \tilde{g}_j Z(\hat{v}_n) + \sum_{i} \sum_{j} \tilde{g}_i \tilde{g}_j Z(\hat{d}_n),
\]

(4.2.10)

where the \( Z \) operator represents the insertion of zeros between two consecutive samples on the refined \( \hat{a}_n, \hat{h}_n, \hat{v}_n \) and \( \hat{d}_n \). The missing coefficient \( 2D\hat{W}_{x,y,n}^* \) in grey in the bottom-left 4 × 4 block of Fig. 4.2 is then used as an improved estimate for \( 2D\hat{W}_{x,y,n}' \).

Note that the process illustrated in Fig. 4.2 can be iterated a number of times. With each iteration, the estimate is gradually improved, up to the limit of what the additional redundancy can reconstruct. Convergence usually occurs after only a few iterations, with diminishing gains achieved with further iterations. Although such quality improvement comes at the expense of computational cost and processing time, this coding scheme can still be used in real-time video applications since the decoder is able to choose the number of iterations to perform according to real-time constraints and computational power. Even after one iteration, the performance is already better than the initial estimates, so if additional time remains for further iterations then further gains are seen. The following subsection delves into the process for the specific case of the Haar wavelet transform, where some modifications can be made depending on the pattern of missing coefficients.
Figure 4.2: Illustration of the proposed iterative root subband recovery method for the full substream loss case.
4.2.3 Root Subband Recovery with Haar Wavelet Transform

When the additional redundancy is generated by applying the 2-tap Haar wavelet decomposition on $^{2D}W^{SR}$, each element in $^{2D}R$ is only correlated to a block of $2 \times 2$ adjacent wavelet coefficients in $^{2D}W^{SR}$. As a result, root subband recovery is implemented in each block of size $2 \times 2$ for adjacent wavelet coefficients. In order to simplify the notation used in this subsection, let us denote the coefficients $^{2D}W'_{2x-1,2y-1,n}$, $^{2D}W'_{2x-1,2y,n}$, $^{2D}W'_{2x,2y-1,n}$ and $^{2D}W'_{2x,2y,n}$ in the same block as $W'_1$, $W'_2$, $W'_3$ and $W'_4$, respectively. From (4.1.3) and (4.2.8), we find that

$$^{2D}R'_{x,y,n} = \frac{1}{2} \{W'_1 + W'_2 + W'_3 + W'_4\}, \quad (4.2.11)$$

when $W'_1$, $W'_2$, $W'_3$ and $W'_4$ are decoded at a high enough bit rate. Also, for the Haar wavelet transform we see that (4.1.2) can be rewritten at the decoder as

$$a' = \frac{1}{2} \{W'_1 + W'_2 + W'_3 + W'_4\}$$

$$h' = \frac{1}{2} \{W'_1 - W'_2 + W'_3 - W'_4\}$$

$$v' = \frac{1}{2} \{W'_1 + W'_2 - W'_3 - W'_4\}$$

$$d' = \frac{1}{2} \{W'_1 - W'_2 - W'_3 + W'_4\}. \quad (4.2.12)$$

For the $2 \times 2$ subblocks operated on by the Haar wavelet transform, we can break down any missing/lost coefficients to the following four cases.

Case 1: only one coefficient is missing from the $2 \times 2$ block. For example, if $W'_1$ is
missing, it can be reconstructed from (4.2.11) as

\[
\hat{W}^*_1 = 2^{2D} R_{x,y,n} - W_2' - W_3' - W_4'.
\] (4.2.13)

We note that the 2-D error concealment process can be skipped in this case since the redundancy contains sufficient information to reconstruct the missing coefficient.

Case 2: two coefficients are missing from the 2 × 2 block. Note that (4.2.13) is not suitable when there is more than one lost coefficient. A solution for this problem is to recalculate the missing coefficients by exploiting the Haar wavelet decomposition and reconstruction roughly using the procedure outlined in Sec. 4.2.2. Let us assume that two coefficients, \( W_1' \) and \( W_2' \), in the same block are affected by the noisy channel. Initially, the lost coefficients \( W_1' \) and \( W_2' \) are estimated with bilinear interpolation to give \( \hat{W}_1 \) and \( \hat{W}_2 \). This block is then spatially decomposed by one additional level by applying the Haar wavelet transform. Modifying (4.2.12) to estimate the subblocks, we have

\[
\hat{h} = \frac{1}{2} \left\{ \hat{W}_1 - \hat{W}_2 + W_3' - W_4' \right\}
\] (4.2.14)

and

\[
\hat{d} = \frac{1}{2} \left\{ \hat{W}_1 - \hat{W}_2 - W_3' + W_4' \right\}.
\] (4.2.15)

Meanwhile, \( a' \) and \( v' \) can be reconstructed from the additional redundancy
$2D R'_{x,y,n}$. That is

\[ a' = 2D R'_{x,y,n} = \frac{1}{2} \{ W'_1 + W'_2 + W'_3 + W'_4 \} \tag{4.2.16} \]

and

\[ v' = 2D R'_{x,y,n} - W'_3 - W'_4 = \frac{1}{2} \{ W'_1 + W'_2 - W'_3 - W'_4 \}. \tag{4.2.17} \]

If a different pair of missing coefficients existed, then the above four equations would change in terms of which coefficients are received (\(\prime\)), which are estimated (\(~\)), and which take advantage of having $2D R'_{x,y,n}$. From these four equations, the estimates of the missing coefficients $\hat{W}_1$ and $\hat{W}_2$ can be refined by applying an inverse Haar wavelet transform such that

\[ \hat{W}_1 = \frac{1}{2} \{ a' + v' + h + d \} = \frac{1}{2} \{ W'_1 + W'_2 + W'_3 + W'_4 \} \tag{4.2.18} \]

and

\[ \hat{W}_2 = \frac{1}{2} \{ a' + v' - h - d \} = \frac{1}{2} \{ W'_1 + W'_2 - W'_3 - W'_4 \}. \tag{4.2.19} \]

We use a distortion metric which approximates the sum of squared differences (SD) between the original and reconstructed coefficients as

\[ SD = \left( \hat{W}_1 - W'_1 \right)^2 + \left( \hat{W}_2 - W'_2 \right)^2 \]

\[ = \frac{1}{2} \left\{ (\hat{W}_1 - W'_1)^2 + (\hat{W}_2 - W'_2)^2 - 2 \cdot (\hat{W}_1 - W'_1)(\hat{W}_2 - W'_2) \right\}. \tag{4.2.20} \]
Compared to the sum of SD obtained using only bilinear interpolation, the sum of SD can be decreased by

\[
\Delta = \left( \tilde{W}_1 - W'_1 \right)^2 + \left( \tilde{W}_2 - W'_2 \right)^2 - SD
\]

\[
= \frac{1}{2} \left( (\tilde{W}_1 + \tilde{W}_2) - (W'_1 + W'_2) \right)^2. \tag{4.2.21}
\]

Case 3: three coefficients are missing from the $2 \times 2$ block. Let us assume that only $W'_4$ is correctly received and the three missing coefficients $W'_1$, $W'_2$, and $W'_3$ are first estimated with a 2-D error concealment method as $\tilde{W}_1$, $\tilde{W}_2$, and $\tilde{W}_3$, respectively. We then have

\[
a' = 2D_{r,y,n} = \frac{1}{2} \{ W'_1 + W'_2 + W'_3 + W'_4 \}, \tag{4.2.22}
\]

\[
\hat{v} = \frac{1}{2} \left\{ \tilde{W}_1 + \tilde{W}_2 - \tilde{W}_3 - W'_4 \right\}, \tag{4.2.23}
\]

\[
\hat{h} = \frac{1}{2} \left\{ \tilde{W}_1 - \tilde{W}_2 + \tilde{W}_3 - W'_4 \right\} \tag{4.2.24}
\]

and

\[
\hat{d} = \frac{1}{2} \left\{ \tilde{W}_1 - \tilde{W}_2 - \tilde{W}_3 + W'_4 \right\}. \tag{4.2.25}
\]

Then, the estimates of the missing coefficients can be refined as

\[
\tilde{W}_1^* = \frac{1}{2} \{ a' + \hat{v} + \hat{h} + \hat{d} \}
\]

\[
= \frac{1}{4} \left\{ W'_1 + W'_2 + W'_3 + 3 \cdot \tilde{W}_1 - \tilde{W}_2 - \tilde{W}_3 \right\}, \tag{4.2.26}
\]
\[ \tilde{W}_2^* = \frac{1}{2} \left\{ \alpha' + \hat{v} - \hat{h} - \hat{d} \right\} \]
\[ = \frac{1}{4} \left\{ W_1' + W_2' + W_3' - \tilde{W}_1 - 3 \cdot \tilde{W}_2 - \tilde{W}_3 \right\} \quad (4.2.27) \]

and

\[ \tilde{W}_3^* = \frac{1}{2} \left\{ \alpha' - \hat{v} + \hat{h} - \hat{d} \right\} \]
\[ = \frac{1}{4} \left\{ W_1' + W_2' + W_3' - \tilde{W}_1 - \tilde{W}_2 + 3 \cdot \tilde{W}_3 \right\}. \quad (4.2.28) \]

Similar to (4.2.21), such modification can reduce the sum of SD by

\[ \Delta = \frac{5}{16} \left\{ (\tilde{W}_1 + \tilde{W}_2 + \tilde{W}_3) - (W_1' + W_2' + W_3') \right\}^2. \quad (4.2.29) \]

Case 4: four coefficients are missing from the 2 × 2 block. In this case, all of the coefficients in the same block are missing. Continuing with the same approach, the estimated coefficients can be modified from \( \tilde{W}_1, \tilde{W}_2, \tilde{W}_3, \) and \( \tilde{W}_4 \) to

\[ \tilde{W}_1^* = \frac{1}{2} \left\{ \alpha' + \hat{v} + \hat{h} + \hat{d} \right\} \]
\[ = \frac{1}{4} \left\{ W_1' + W_2' + W_3' + W_4' + 3 \cdot \tilde{W}_1 - \tilde{W}_2 - \tilde{W}_3 - \tilde{W}_4 \right\}, \quad (4.2.30) \]

\[ \tilde{W}_2^* = \frac{1}{2} \left\{ \alpha' + \hat{v} - \hat{h} - \hat{d} \right\} \]
\[ = \frac{1}{4} \left\{ W_1' + W_2' + W_3' + W_4' - \tilde{W}_1 + 3 \cdot \tilde{W}_2 - \tilde{W}_3 - \tilde{W}_4 \right\}, \quad (4.2.31) \]
\[
\hat{W}_3^* = \frac{1}{2} \{a' - \hat{v} + \hat{h} - \hat{d}\} \\
= \frac{1}{4} \{W'_1 + W'_2 + W'_3 + W'_4 - \hat{W}_1 - \hat{W}_2 + 3 \cdot \hat{W}_3 - \hat{W}_4\} 
\]

and

\[
\hat{W}_4^* = \frac{1}{2} \{a' - \hat{v} - \hat{h} + \hat{d}\} \\
= \frac{1}{4} \{W'_1 + W'_2 + W'_3 + W'_4 - \hat{W}_1 - \hat{W}_2 - \hat{W}_3 + 3 \cdot \hat{W}_4\}. 
\]

The sum of SD can be reduced by

\[
\triangle = \frac{1}{4} \left\{ \left( \hat{W}_1 + \hat{W}_2 + \hat{W}_3 + \hat{W}_4 \right) - \left( W'_1 + W'_2 + W'_3 + W'_4 \right) \right\}^2. 
\]

As can be seen from the above discussion, the performance of the proposed method based on the Haar transform is dependent on different missing patterns of coefficients in the 2×2 block. It provides the best reconstruction when there is only one coefficient lost in the same block. The quality improvement decreases as the number of missing coefficients increases. The processes in Cases 2–4 can be iterated to further refine the estimates of the missing coefficients, but we have found that one iteration is sufficient when bilinear interpolation is used to obtain the initial estimates. Note that issues in recovering some of the missing coefficients in the spatial root subband will occur if some coefficients in \(2D R_n'\) are missing. In those instances, the missing coefficients need to be estimated only by using other error concealment techniques, such as bilinear
interpolation.

4.3 Simulation Results

In this section, we conduct experiments to investigate the performance of the proposed MD video coding algorithm in noiseless and noisy channels. We compare the performance of the proposed algorithm to ERC-SPIHT [40,41] and interband interpolation [57] for estimating missing coefficients. The simulation results are evaluated with respect to (1) source coding efficiency and (2) error concealment performance.

4.3.1 Experimental Data

The experiments are conducted using the 352 x 240 x 48 monochrome “Football” and “Susie” video sequences with frame rate of 30 frames/s. We use 16 frames for the GOF. A three-level decomposition using the CDF 9/7 filter is applied in both the spatial and temporal domains with reflection extensions both at each image boundary and each GOF boundary. In this work, we employ the Haar or CDF 9/7 wavelet transform to generate the additional redundancy. The wavelet coefficients and additional redundancy are divided into multiple groups ($S = 16$). Each group is independently encoded at a coding rate of $c$ bpp. Therefore, the total transmission rate is $352 \times 240 \times 30 \times c$ bps. In these experiments, we use PSNR and MSSIM to assess the video reconstruction quality. The reference software [81] is used to implement the MSSIM algorithm. Note that the PSNR and MSSIM index are both measured on a frame basis in our work.
4.3.2 Source Coding Efficiency

To compare the source coding efficiency of the proposed MD video coding algorithm to that of ERC-SPIHT, we assume that the channel is error-free and all substreams are thus received correctly at the decoder. Figs. 4.3 and 4.4 show the average PSNR and MSSIM values for the proposed algorithm and ERC-SPIHT with different coding rates \( c \) when the total number of substreams is \( S = 16 \). There are just small losses in performance with the proposed algorithm compared to ERC-SPIHT. These small losses result from the additional redundancy overhead that is inserted into the substreams. We also observe that such small losses do not change much with different coding rates \( c \). Specifically, the small losses fluctuate from 0.17 dB to 0.20 dB in terms of PSNR and from 0.0021 to 0.0064 in terms of MSSIM in the "Football" sequence; these losses range from 0.13 dB to 0.20 dB in terms of PSNR and from 0.0004 to 0.0016 in terms of MSSIM in the "Susie" sequence. In addition, the results also show that the performance with the proposed method for the Haar filter is very similar to that of the CDF 9/7 filter and the corresponding results almost overlap in Figs. 4.3 and 4.4.

4.3.3 Error Concealment Performance

We compare the R-D performance of the proposed algorithm to ERC-SPIHT and the interband interpolation technique in [57] for estimating missing coefficients in noisy channels to investigate the error concealment feature. In this work, we only focus on recovering the missing coefficients in the spatial root subband and any missing
Figure 4.3: Comparison of PSNR (dB) and MSSIM of aFootball video sequence in noiseless channels at different coding rates $c$ ($S = 16$ substreams). (a) PSNR (dB). (b) MSSIM.
Figure 4.4: Comparison of PSNR (dB) and MSSIM of "Susie" video sequence in noiseless channels at different coding rates $c$ ($S = 16$ substreams). (a) PSNR (dB). (b) MSSIM.
coefficients in other subbands are zeroed. With the proposed method, the missing
coefficients in the spatial root subband are first estimated by bilinear interpolation and
then recalculated by the proposed root subband recovery. In contrast to the proposed
method, the missing coefficients in the spatial root subband are only reconstructed
by bilinear interpolation in ERC-SPIHT [40,41] and interband interpolation in [57].

4.3.3.1 Performance Evaluation with Different Missing Patterns of Coef­
ficients

We first explore the individual benefit offered by the proposed method with CDF
9/7 and Haar on different missing patterns of coefficients. In this section, we assume
that there are four substreams corrupted due to transmission errors of the $S = 16$
substreams, which results in $\binom{16}{4} = 1820$ possible missing patterns of coefficients
in each $4 \times 4$ block for dispersive grouping. If we consider the 2-D Haar wavelet
transform, which effectively operates on non-overlapping $2 \times 2$ subblocks, then we
can classify the possible missing coefficients patterns in a $4 \times 4$ ($S = 16$) block into
the following five categories, with examples shown in Fig. 4.5:

- Category 1: Four missing coefficients in the same $2 \times 2$ block (4 possible com­
  binations);

- Category 2: Three missing coefficients in the same $2 \times 2$ block and one missing
  coefficient is in another (192 possible combinations);

- Category 3: Two missing coefficients in the same $2 \times 2$ block and two more in
  another $2 \times 2$ block (216 possible combinations);
• Category 4: Two missing coefficients in the same $2 \times 2$ block and the remaining two are in two different $2 \times 2$ blocks (1152 possible combinations);

• Category 5: Each missing coefficient is in a different $2 \times 2$ block (256 possible combinations);

Correspondingly, the probability of each category is 0.0022, 0.1055, 0.11867, 0.6336, and 0.1407, respectively, when four substreams are lost. A similar analysis could be done if more or less than four substreams were corrupted.

We choose one pattern as an example from each category as shown in Fig. 4.5(a)-(e) and four scenarios are investigated for each pattern of lost substreams as given below:

• Scenario 1: ERC-SPIHT (bilinear interpolation) [40,41];

• Scenario 2: our proposed additional redundancy generated by CDF 9/7 + interband interpolation [57];

• Scenario 3: our proposed additional redundancy generated by CDF 9/7 + bilinear interpolation + iterative root subband recovery (Sec.4.2.2);

• Scenario 4: our proposed additional redundancy generated by Haar + bilinear interpolation + root subband recovery (Sec. 4.2.3);

The simulation results are summarized in Tables 4.1 and 4.2. Scenario 1 does not incorporate the additional redundancy and thus has a significant performance loss compared to our proposed method (Scenario 3 and 4) by up to 2.58 dB in terms of
Figure 4.5: Patterns randomly chosen from the five categories in Sec. 4.3.3.1 when considering Haar $2 \times 2$ subblock based missing coefficient cases in $4 \times 4$ blocks ($S = 16$ substreams and four substreams are lost with dispersive grouping). (a) Pattern 1. (b) Pattern 2. (c) Pattern 3. (d) Pattern 4. (e) Pattern 5.
PSNR and 0.0173 in terms of MSSIM. Scenario 2 combines our partial work (additional redundancy at the encoder) and the interband interpolation approach from [57] at the decoder. In this scenario, the additional redundancy $2^D R'$ is only used to estimate the missing coefficients in the spatial root subband of $2^D W'$ by interband interpolation. However, some coefficients in the additional redundancy $2^D R'$ may be lost and the sign of some coefficients in $2^D W'$ may also be lost. Such loss will eventually affect the reconstruction performance in Scenario 2. In Scenario 3, the accuracy in the estimation of the missing coefficient in the spatial root subband of $2^D W'$ can be further improved by using the proposed iterative root subband recovery.

In these tables, we can see that one iteration is sufficient to have improved PSNR and MSSIM values over competing techniques (Scenario 1 and 2). Additional iterations further improve the results with an exponential decay towards saturation, so the first few iterations provide the most improvement. Specifically, the bulk of the quality improvements are made after only 5-10 iterations, with marginal gains afterwards. The performance improvement stops after about 50 iterations ($I = 50$). As discussed in Sec. 4.2.3, an important difference with respect to the proposed method based on CDF 9/7 is that the proposed method with Haar is sensitive to the pattern of missing coefficients such as those shown in Fig. 4.5. The experiments show that Scenario 3 yields a better performance than Scenario 4 in patterns 1-4, and is only worse in pattern 5 where Haar has an advantage when only one coefficient is missing per $2 \times 2$ block.
Table 4.1: Comparison of PSNR (dB) and MSSIM for “Football” video sequence under different missing patterns of Fig. 4.5 ($S = 16$ substreams and $c = 1.0$ bpp)

(a) $PSNR \ (dB)$

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Pattern 1</th>
<th>Pattern 2</th>
<th>Pattern 3</th>
<th>Pattern 4</th>
<th>Pattern 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>24.61</td>
<td>25.14</td>
<td>25.46</td>
<td>24.39</td>
<td>25.15</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>23.74</td>
<td>24.73</td>
<td>24.83</td>
<td>23.88</td>
<td>24.50</td>
</tr>
<tr>
<td>$I = 1$</td>
<td>25.29</td>
<td>25.73</td>
<td>25.95</td>
<td>24.99</td>
<td>25.52</td>
</tr>
<tr>
<td>$I = 5$</td>
<td>25.70</td>
<td>26.12</td>
<td>26.40</td>
<td>25.58</td>
<td>25.90</td>
</tr>
<tr>
<td>$I = 10$</td>
<td>25.86</td>
<td>26.20</td>
<td>26.50</td>
<td>25.79</td>
<td>25.99</td>
</tr>
<tr>
<td>$I = 50$</td>
<td>25.99</td>
<td>26.24</td>
<td><strong>26.56</strong></td>
<td><strong>25.96</strong></td>
<td>26.03</td>
</tr>
<tr>
<td>$I = 100$</td>
<td>25.98</td>
<td><strong>26.26</strong></td>
<td>26.56</td>
<td>25.95</td>
<td>26.00</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>25.24</td>
<td>25.94</td>
<td>26.20</td>
<td>25.72</td>
<td><strong>26.27</strong></td>
</tr>
</tbody>
</table>

(b) $MSSIM$

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Pattern 1</th>
<th>Pattern 2</th>
<th>Pattern 3</th>
<th>Pattern 4</th>
<th>Pattern 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>0.7695</td>
<td>0.7747</td>
<td>0.7792</td>
<td>0.7771</td>
<td>0.7798</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>0.7597</td>
<td>0.7647</td>
<td>0.7664</td>
<td>0.7657</td>
<td>0.7669</td>
</tr>
<tr>
<td>$I = 1$</td>
<td>0.7737</td>
<td>0.7788</td>
<td>0.7827</td>
<td>0.7807</td>
<td>0.7835</td>
</tr>
<tr>
<td>$I = 5$</td>
<td>0.7785</td>
<td>0.7826</td>
<td>0.7867</td>
<td>0.7850</td>
<td>0.7888</td>
</tr>
<tr>
<td>$I = 10$</td>
<td>0.7805</td>
<td>0.7833</td>
<td>0.7875</td>
<td>0.7866</td>
<td>0.7899</td>
</tr>
<tr>
<td>$I = 50$</td>
<td>0.7819</td>
<td>0.7835</td>
<td><strong>0.7879</strong></td>
<td><strong>0.7885</strong></td>
<td>0.7895</td>
</tr>
<tr>
<td>$I = 100$</td>
<td>0.7816</td>
<td><strong>0.7837</strong></td>
<td>0.7878</td>
<td>0.7880</td>
<td>0.7889</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>0.7720</td>
<td>0.7810</td>
<td>0.7843</td>
<td>0.7869</td>
<td><strong>0.7922</strong></td>
</tr>
</tbody>
</table>
Table 4.2: Comparison of PSNR (dB) and MSSIM for “Susie” video sequence under different missing patterns of Fig. 4.5 ($S = 16$ substreams and $c = 1.0$ bpp)

(a) **PSNR (dB)**

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Pattern 1</th>
<th>Pattern 2</th>
<th>Pattern 3</th>
<th>Pattern 4</th>
<th>Pattern 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30.19</td>
<td>31.54</td>
<td>32.03</td>
<td>29.09</td>
<td>30.64</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>27.72</td>
<td>29.63</td>
<td>29.99</td>
<td>27.98</td>
<td>29.00</td>
</tr>
<tr>
<td>$I = 1$</td>
<td>31.41</td>
<td>32.45</td>
<td>32.86</td>
<td>29.69</td>
<td>31.14</td>
</tr>
<tr>
<td>$I = 5$</td>
<td>32.16</td>
<td>33.20</td>
<td>33.77</td>
<td>30.23</td>
<td>31.68</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>32.48</td>
<td>33.36</td>
<td>33.98</td>
<td>30.41</td>
<td>31.82</td>
</tr>
<tr>
<td>$I = 10$</td>
<td>32.75</td>
<td>33.42</td>
<td><strong>34.09</strong></td>
<td>30.69</td>
<td>31.94</td>
</tr>
<tr>
<td>$I = 50$</td>
<td>32.77</td>
<td>33.44</td>
<td>34.09</td>
<td><strong>30.71</strong></td>
<td>31.93</td>
</tr>
<tr>
<td>$I = 100$</td>
<td><strong>32.77</strong></td>
<td><strong>33.44</strong></td>
<td>34.09</td>
<td><strong>30.71</strong></td>
<td><strong>32.13</strong></td>
</tr>
<tr>
<td>Scenario 4</td>
<td>31.33</td>
<td>33.03</td>
<td>33.42</td>
<td>30.45</td>
<td><strong>32.13</strong></td>
</tr>
</tbody>
</table>

(b) **MSSIM**

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Pattern 1</th>
<th>Pattern 2</th>
<th>Pattern 3</th>
<th>Pattern 4</th>
<th>Pattern 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.9057</td>
<td>0.9129</td>
<td>0.9159</td>
<td>0.9073</td>
<td>0.9149</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>0.8919</td>
<td>0.8956</td>
<td>0.8966</td>
<td>0.8902</td>
<td>0.8945</td>
</tr>
<tr>
<td>$I = 1$</td>
<td>0.9121</td>
<td>0.9179</td>
<td>0.9208</td>
<td>0.9134</td>
<td>0.9197</td>
</tr>
<tr>
<td>$I = 5$</td>
<td>0.9173</td>
<td>0.9218</td>
<td>0.9251</td>
<td>0.9184</td>
<td>0.9246</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>0.9195</td>
<td>0.9226</td>
<td>0.9260</td>
<td>0.9205</td>
<td>0.9256</td>
</tr>
<tr>
<td>$I = 10$</td>
<td>0.9215</td>
<td>0.9228</td>
<td><strong>0.9263</strong></td>
<td>0.9244</td>
<td>0.9260</td>
</tr>
<tr>
<td>$I = 50$</td>
<td><strong>0.9216</strong></td>
<td><strong>0.9230</strong></td>
<td>0.9262</td>
<td><strong>0.9246</strong></td>
<td>0.9258</td>
</tr>
<tr>
<td>$I = 100$</td>
<td><strong>0.9216</strong></td>
<td><strong>0.9230</strong></td>
<td>0.9262</td>
<td><strong>0.9246</strong></td>
<td><strong>0.9258</strong></td>
</tr>
<tr>
<td>Scenario 4</td>
<td>0.9092</td>
<td>0.9201</td>
<td>0.9227</td>
<td>0.9222</td>
<td><strong>0.9284</strong></td>
</tr>
</tbody>
</table>
4.3.3.2 Performance Evaluation at Different Coding Rates

We now investigate the performance of the proposed method at different coding rates $c$ (0.5–1.5 bpp). In these experiments, we assume that there are $S = 16$ substreams and four substreams are lost. As introduced above, we have five categories in terms of the missing coefficient patterns. In each category, the PSNR and MSSIM are the average over 10 independent runs, each of which arbitrarily chooses a different missing pattern (except for Category 1, which only has 4 unique missing coefficient patterns and thus 4 runs correspondingly). The final PSNR and MSSIM are derived from $4 + 4 \times 10 = 44$ runs and they are calculated as the weighted sum of the average PSNR and MSSIM in each category with the corresponding probability. We choose bilinear interpolation as the 2-D error concealment technique and $I = 50$ for the proposed method with CDF 9/7. Figs. 4.6 and 4.7 illustrate the difference of final PSNR and MSSIM among ERC-SPIHT (Scenario 1), the proposed method with CDF 9/7 (Scenario 3) and Haar (Scenario 4) for the “Football” and “Susie” sequences, respectively. From the simulation results in Fig. 4.6, we can see that the proposed method enhances the reconstructed video quality at the decoder significantly. As discussed in Sec. 4.2.2, the success of root subband recovery depends on the rate at which the substreams are encoded. Correspondingly, the performance gain achieved by the proposed method noticeably improves as $c$ increases. As an example in Fig. 4.6, the amount of the improvement obtained by the proposed method with CDF 9/7 is about 0.86 dB in terms of PSNR and 0.0049 in terms of MSSIM at $c = 0.5$ bpp for the “Football” sequence. When $c = 1.5$ bpp, the amount of the improvement obtained by the
proposed method with CDF 9/7 is up to 1.36 dB in terms of PSNR and 0.0117 in terms of MSSIM. We also observe that the PSNR of the proposed method with CDF 9/7 is 0.08 dB higher at \( c = 0.5 \) bpp and 0.27 dB higher at \( c = 1.5 \) bpp higher than that of the proposed method with Haar. In addition, the MSSIM of the proposed method with CDF 9/7 is 0.0006 higher at \( c = 0.5 \) bpp and 0.0023 higher at \( c = 1.5 \) bpp higher than that of the proposed method with Haar. Fig. 4.7(a) shows that the PSNRs of the proposed method with CDF 9/7 are 1.62–1.84 dB higher than those of ERC-SPIHT and 0.28–0.36 dB higher than those of the proposed method with Haar. Fig. 4.7(b) shows that the MSSIMs of the proposed method with CDF 9/7 are 0.0110–0.0128 higher than those of ERC-SPIHT, and still 0.0015–0.0023 higher than those of the proposed with Haar.

4.3.3.3 Performance Evaluation with Different Number of Substreams Lost

We next extend the performance studies on the proposed method with CDF 9/7 and Haar under different number of substreams lost. We fix the coding rate at \( c = 1.0 \) bpp. According to the number of substreams lost, we first decide the number of categories in terms of the missing patterns. Similar to the above experiments, the PSNR and MSSIM are averaged over 10 independent runs in each category. The final PSNR and MSSIM are then calculated as the weighted sum of the average PSNR and MSSIM with the corresponding probability. The simulation results are given in Figs. 4.8 and 4.9 for “Football” and “Susie”, respectively. In these experiments, we also choose \( I = 50 \) for the proposed method with CDF 9/7. The results in Figs.
Figure 4.6: Comparison of final PSNR (dB) and MSSIM of “Football” sequence at different coding rates $c$ ($S = 16$ substreams and four substreams are lost). (a) PSNR (dB) and (b) MSSIM.
Figure 4.7: Comparison of final PSNR (dB) and MSSIM of “Susie” sequence at different coding rates $c$ ($S = 16$ substreams and four substreams are lost). (a) PSNR (dB) and (b) MSSIM.
4.3 and 4.4 are included for reference (corresponding to 0 substreams lost in Figs. 4.8 and 4.9). We can observe that the proposed method with CDF 9/7 (Scenario 3) and Haar (Scenario 4) improves the reconstructed video quality at the decoder. The results obtained by the proposed method are both higher than those of ERC-SPIHT (Scenario 1) by at least 0.58 dB in terms of PSNR and 0.0018 in terms of MSSIM, depending on the number of substreams lost. The amount of performance improvement is much more noticeable when the number of substreams lost is smaller than 5. Specifically, the performance differences range from 0.58 dB to 1.26 dB in terms of PSNR and 0.0018 to 0.0106 in terms of MSSIM on the “Football” sequence, and in the case of “Susie” sequence the differences are 0.76–2.32 dB in terms of PSNR and 0.0038–0.0124 in terms of MSSIM. The results in Figs. 4.8 and 4.9 also show that the proposed algorithm based on CDF 9/7 yields a higher PSNR than Haar. For the “Football” sequence, the PSNR improvement obtained by the proposed method based on CDF 9/7 is about 0.02–0.22 dB. For the “Susie” sequence, the proposed method with CDF 9/7 outperforms by 0.05–0.34 dB. For both video sequences, such PSNR improvement is easier to perceive when the total number of substreams lost is between 2 and 4. In addition, the MSSIM values of the proposed algorithm using CDF 9/7 are very close to those of Haar and the corresponding lines almost overlap in Figs. 4.8 and 4.9.

4.3.3.4 Visual Comparison

Figs. 4.10 and 4.11 show sample frames to compare the visual results obtained in Sec. 4.3.3.1. As we can see, in the case of no error concealment, there are many black spots
Figure 4.8: Comparison of final PSNR (dB) and MSSIM of “Football” sequence with different number of substreams lost ($S = 16$ substreams and $c = 1.0$ bpp). (a) PSNR (dB). (b) MSSIM.
Figure 4.9: Comparison of final PSNR (dB) and MSSIM of “Susie” sequence with different number of substreams lost ($S = 16$ substreams and $c = 1.0$ bpp). (a) PSNR (dB). (b) MSSIM.
in the frames. When the missing coefficients are estimated in Scenario 1 and 2 (Fig. 4.10(c)–(d) and Fig. 4.11(c)–(d)), some missing areas are not recovered successfully and these spots are marked by rectangles. Fig. 4.10(e)–(h) and Fig. 4.11(e)–(h) show that when the missing coefficients are estimated in Scenario 3, the reconstruction for these spots are improved as more iterations are applied. We can see that a reasonable reconstruction can be achieved at $I = 10$ and the perceptual gain is less significant after $I = 10$. Fig. 4.10(i) and Fig. 4.11(i) represent the reconstruction in Scenario 4. The visual performance is slightly worse than that of Scenario 3 when $I = 10$ in the “Football” sequence and $I = 1$ in the “Susie” sequence. Such difference agrees with the objective PSNR and MSSIM evaluation.

4.4 Conclusions

Motivated by the previous work, in this chapter we generalize the proposed domain-partitioning based MD coding algorithm based on ERC-SPIHT for the generation of an arbitrary number $S$ of substreams. In general, we adopt the proposed framework presented in Fig. 3.2 and propose a different strategy for implementing the additional redundancy generation method used at the encoder and the error concealment technique applied at the decoder. In this chapter, the added redundancy is generated by a wavelet projection based method of the spatial root subband $2DWSR$ with two types of wavelet transforms (Haar and CDF 9/7). As a result, the missing coefficients in $2DWSR$ can be reconstructed in two steps: first by using existing 2-D error concealment technique, and second with the proposed root subband recovery. The
Figure 4.10: 352 × 240 “Football” sequence (S = 16 substreams and c = 1.0 bpp). (a) Original sequence (frame 47). (b) Corrupted sequence using the missing coefficient pattern in Fig. 4.5(d): $PSNR = 11.33$ dB and $MSSIM = 0.4179$. (c) Scenario 1: $PSNR = 23.25$ dB and $MSSIM = 0.7377$. (d) Scenario 2: $PSNR = 22.69$ dB and $MSSIM = 0.7199$. (e) Scenario 3 ($I = 1$): $PSNR = 23.91$ dB and $MSSIM = 0.7410$. (f) Scenario 3 ($I = 5$): $PSNR = 24.49$ dB and $MSSIM = 0.7456$. (g) Scenario 3 ($I = 10$): $PSNR = 24.65$ dB and $MSSIM = 0.7470$. (h) Scenario 3 ($I = 50$): $PSNR = 24.77$ dB and $MSSIM = 0.7479$. (i) Scenario 4: $PSNR = 24.41$ dB and $MSSIM = 0.7464$. 
Figure 4.11: 352 × 240 “Susie” sequence (S = 16 substreams and c = 1.0 bpp). (a) Original sequence (frame 27). (b) Corrupted sequence using the missing coefficient pattern in Fig. 4.5(a): \( \text{PSNR} = 12.11 \text{ dB} \) and \( \text{MSSIM} = 0.4189 \). (c) Scenario 1: \( \text{PSNR} = 29.80 \text{ dB} \) and \( \text{MSSIM} = 0.8923 \). (d) Scenario 2: \( \text{PSNR} = 26.38 \text{ dB} \) and \( \text{MSSIM} = 0.8802 \). (e) Scenario 3 (\( I = 1 \)): \( \text{PSNR} = 30.89 \text{ dB} \) and \( \text{MSSIM} = 0.8982 \). (f) Scenario 3 (\( I = 5 \)): \( \text{PSNR} = 31.56 \text{ dB} \) and \( \text{MSSIM} = 0.9029 \). (g) Scenario 3 (\( I = 10 \)): \( \text{PSNR} = 31.86 \text{ dB} \) and \( \text{MSSIM} = 0.9050 \). (h) Scenario 3 (\( I = 50 \)): \( \text{PSNR} = 32.17 \text{ dB} \) and \( \text{MSSIM} = 0.9077 \). (i) Scenario 4: \( \text{PSNR} = 30.68 \text{ dB} \) and \( \text{MSSIM} = 0.8946 \).
former step is used to estimate the missing coefficients in the spatial root and high-frequency subbands by exploiting the inherent redundancy, while the latter attempts to utilize the inserted redundancy to further improve the precision in the estimation of the missing spatial root subband coefficients. The proposed root subband recovery method can be iteratively applied and accuracy of the reconstruction can be gradually increased with each iteration. Experimental results on different video sequences show that the general domain-partitioning based MD video coder can be applied for the generation of an arbitrary number \( S \) of substreams with any type of missing coefficient patterns. It provides lower source coding efficiency than ERC-SPIHT in lossless transmission but it is more resilient in an error-prone transmission environment. In particular, the proposed approach achieves a significant improvement on video quality by up to 2.58 dB in the presence of full substream loss when compared to its counterpart ERC-SPIHT.

However, in this chapter we only consider the full substream loss case, in which we assume that the decoding failure always occurs at a very early stage of a substream and all the coefficients in that substream are affected. The problem now is how to extend the proposed video coder to the partial substream loss case. For example, when the coefficient \( 3^D W_{x,y,n} \) is partially lost due to transmission errors, we need to find its missing bitplane values from the additional redundancy at the decoder. We will attempt to solve this problem in the following chapter.
Chapter 5

Revised Domain-Partitioning Based Multiple Description Video Coding Algorithm in Conjunction with 3-D SPIHT Decoding Range: Partial Substream Loss

Although the proposed method in previous studies has been shown to be more resilient in an error-prone transmission environment, it is restricted to the full substream loss case, in which we assume that the transmission errors occur at the beginning of some substreams and other substreams are correctly received at the decoder. However, this is not always the case because transmission errors can occur at any stage of a
substream. Consider that the substreams are sent across error-prone packet-based networks. In this context, the substreams need to be packetized for transmission but there are various types of transmission errors, such as random bit errors and packet losses. Packets generated from the substreams are subject to these adverse conditions. As a consequence, any packet may be corrupted due to transmission errors. For example, assume that all of the packets generated from substream-i are correctly received at the decoder except the 11th one is missing. Due to the properties of 3-D SPIHT, the initial error-free portion of the substream up to the point where the packet corruption occurs is decodable, while the rest of the available substream is likely to be useless. In our example, only the first 10 packets can be used to decode the coefficients and the substream-i is considered as partially missing. If substream-i is decoded without any error concealment, we only have a small number of bitplane values and other bitplane values are missing. As a result, the coefficients are reconstructed at a low level of accuracy and a poor reconstructed video quality will be produced at the decoder. For this reason, it is important to consider partial substream loss for robust video transmission with effective error concealment.

However, there has been little comparable development of MD video coding algorithms with error concealment for partial substream loss. The study [42] was, to the best of our knowledge, the only work to investigate the domain-partitioning based MD video coder based on 3-D SPIHT for partial substream loss. Although it provided some useful insights, the coder considered there can only be applied to a specific case, in which only two substreams are generated and one of them is partially missing. The missing bitplane values can be estimated by using the smoothness between the two
substreams in the wavelet transform domain.

In this chapter, we aim to extend the proposed domain-partitioning based MD video coder to the partial substream loss case. In contrast to the full substream loss case, three types of resources are available at the decoder: (1) inherent redundancy, (2) added redundancy, and (3) partially received information (such as the first 10 packets in substream-i in the above example). A key to the success of the extension is how to exploit the above resources to reconstruct the missing information in the spatial root subband.

5.1 Proposed System Overview

We now consider a new error-resilient framework for ERC-SPIHT-coded video sent across lossy links. The overall system is depicted in Fig. 5.1. Compared to the original coding schemes in previous work, the proposed framework contains a different spatial root subband error concealment module at the decoder side, where two new functions are included, namely range calculation and iterative root subband recovery. Furthermore, the spatial root subband error concealment module is applied at the decoder right after the 3-D SPIHT decoder. As a consequence, the error concealment module in this framework is deployed in the 3-D wavelet domain. This is in contrast to the coding scheme in Fig. 3.2, where the root subband error concealment method is implemented in the 2-D wavelet domain.

The rest of the proposed framework is similar to the coding work in Fig. 3.2. For example, the additional redundancy is generated by using the same technique at the
Figure 2.1: Framework of the proposed video coding algorithm with extensions to partial substream loss.

- GOF
  - 2D Spatial Reconstruction
  - Error Concealment on Detail Subband
  - 1D Temporal Reconstruction
  - Error Concealment on Root Subband
  - Iterative Root Subband Recovery
  - Bilinear Interpolation on Root Subband
  - Range Calculation
  - 3D SPIHT Decoder

- Encoder
  - 2D Spatial Decomposition
  - Additional Redundancy Generation
  - 1D Temporal Decomposition
  - Domain Partitioning
  - 3D SPIHT Encoder
  - Channels

- Decoder
encoder. In essence, the additional redundancy is obtained by a wavelet projection based method: each frame of the spatial root subband $^{2D}W^{SR}$ is spatially decomposed by one additional transform level and the resulting coefficients in the approximation subband are grouped together in a 3-D structure to form the additional redundancy.

After the wavelet decomposition and redundancy generation, we follow the same procedure to encode $^{3D}W$ and $^{3D}R$ as proposed in Chapters 3 and 4. For example, we use dispersive grouping to divide the coefficients in $^{3D}W$ and $^{3D}R$ into $S$ groups (i.e., $^{3D}W_i$ and $^{3D}R_i$, where $i = 1, 2, \ldots, S$). Each group is then encoded by the modified AT-SPIHT coder and the additional redundancy is added at the end of each substream in reverse order. However, the substream generation procedure depicted in Fig. 3.5 is inapplicable in the partial substream loss case. For example, all of the packets generated for substream-$i$ are available at the decoder except the $11^{th}$ one is missing. Assume that this packet is only related to the coefficients in $^{3D}W_i$. In this case, decoding of $^{3D}R_i$ can still be performed from the end of substream-$i$. A natural challenge that arises is where to stop the decoding process of $^{3D}R_i$ (i.e., how to determine the end location of the bits related to $^{3D}R_i$ in substream-$i$). To solve this problem, we adopt a new parameter $c^R$ in the global header as the coding rate for the additional redundancy $^{3D}R$. The objective of adopting $c^R$ is for the decoder to use $c^R$ and the dimensions of $^{3D}R_i$ to calculate the total number of bits assigned for the additional redundancy $^{3D}R_i$. Furthermore, the number of bits related to the coefficients in $^{3D}W_i$ is also decided. As a result, any decoding failure in $^{3D}W_i$ and $^{3D}R_i$ does not affect the other in the partial substream loss case.

When all of the substreams are successfully received at the decoder over noiseless
channels, the additional redundancy $3^D R$ is not required and we only need to decode the coefficients related to $3^D W$. We define $3^D W'$ as the coefficients reconstructed at the decoder. In the noiseless case, the coefficients in $3^D W'$ are reconstructed at the full SPIHT encoder accuracy. The reverse temporal transform is first applied to $3^D W'$ and the resulting coefficients are denoted by $2^D W'$. The reverse spatial transform is then applied to $2^D W'$ to produce the reconstructed video sequence. Note that in the case of noiseless transmission, the error concealment step in Fig. 5.1 can be skipped. In the following section we describe the error concealment procedure when the channel is noisy and three types of resources can be used (1) inherent redundancy, (2) added redundancy, and (3) partially received information.

5.2 Revised Error Concealment Technique for Partial Substream Loss

Due to packet losses, the decoding process of the substreams may stop at different bitplanes and thus each coefficient may have a different number of decodable bitplanes. For example, we assume that the spatial root subband coefficient $3^D W_{x,y,n}$ is encoded in substream-$i$ and only a few number of bitplane values are received at the decoder due to transmission errors. As illustrated in Fig. 5.2, we assume that the SPIHT decoding process of the spatial root subband coefficient $3^D W_{x,y,n}$ starts from the highest bitplane $dW_{x,y,n}^{max}$ and stops at the end of the bitplane $dW_{x,y,n}^{min}$; the rest of bitplane values are missing due to transmission errors. At the highest bitplane $dW_{x,y,n}^{max}$
(holding the first nonzero bit), it is known that $2^{d_{W_{x,y,n}}^{max}} \leq |3D_{W_{x,y,n}}| < 2^{d_{W_{x,y,n}}^{max}+1}$. The SPIHT decoder uses this information, plus the sign bit that is output just after the first nonzero value, to set $3D_{W_{x,y,n}} = \pm 1.5 \times 2^{d_{W_{x,y,n}}^{max}}$. This reconstruction corresponds to the significance pass in 3-D SPIHT. In addition, such reconstruction can be further approximated by 3-D SPIHT during the refinement pass. For example, for the bitplane $d$, the decoder adds or subtracts $2^{d-1}$ to $3D_{W_{x,y,n}}^{'}$, depending on the corresponding bitplane value (either 1 or 0) coded in the substream-$i$. In the lossless case, this refinement pass can be applied until the full SPIHT encoder accuracy is achieved. However, in the lossy case, such refinement pass has to be interrupted since some bitplane information is unavailable (such as $d < d_{W_{x,y,n}}^{min}$ in Fig. 5.2). In this example, the coefficient $3D_{W_{x,y,n}}^{'}$ is now denoted as $3D_{\hat{W}_{x,y,n}}$ and it can be initially approximated by the 3-D SPIHT decoder as

$$3D_{\hat{W}_{x,y,n}} = \pm \left\{ \sum_{d=d_{W_{x,y,n}}^{min}}^{d_{W_{x,y,n}}^{max}} 2^d \times I(d; 3D_{W_{x,y,n}}) \right\} + 2^{d_{W_{x,y,n}}^{min}-1},$$  \hspace{1cm} (5.2.1)

where function $I(d; 3D_{W_{x,y,n}})$ is the value of the coefficient $3D_{W_{x,y,n}}$ at the bitplane $d$. In this case, if the coefficient $3D_{W_{x,y,n}}$ is only decoded without effective error concealment, a lower reconstructed video quality will be produced at the decoder.

The aim of our revised error concealment technique in the proposed MD video coding framework is to find the missing bitplane values of the coefficient $3D_{W_{x,y,n}}$ in the spatial root subband. As a consequence, the reconstructed video quality can be enhanced at the decoder. As illustrated in Fig. 5.1, the error concealment algorithm is comprised of three functions: (1) range calculation, (2) bilinear interpolation on
Figure 5.2: Binary representation of the coefficient $3D\hat{W}_{x,y,n}$ partially decoded with some missing bitplane values.

root subband, and (3) root subband recovery. The first two functions (1) and (2) are applied to define the decoding range of $3D\hat{W}_{x,y,n}$ and initially estimate its missing bitplane values, while the function (3) attempts to improve the accuracy in recovering the missing bitplane values. We now introduce these functions in detail in the following subsections. In our work, we consider that the coefficient $3D\hat{W}_{x,y,n}$ is partially lost at the decoder when $dW_{x,y,n}^{\text{min}} > T_1$ and the error concealment technique is required. To differentiate from the noiseless case, the notation $2D\hat{W}^{SR}$ and $3D\hat{W}^{SR}$ are denoted as $2D\hat{W}^{SR}$ and $3D\hat{W}^{SR}$, respectively.

5.2.1 Range Calculation

In general, the range calculation method is used to specify the limits to which the reconstruction obtained in the following bilinear interpolation and root subband recovery techniques is required to agree. As illustrated in Fig. 5.1, error concealment
is applied right after the 3-D SPIHT decoder. Correspondingly, the range calculation function in this work is deployed in the 3-D wavelet domain. As discussed above, we consider that the spatial root subband coefficient \(3D W_{x,y,n}\) is partially lost and the lower/upper bounds need to be computed by the range calculation function when \(dW_{x,y,n}^{\text{min}} > T_1\). Compared to Fig. 5.2, the magnitude of the approximation in (5.2.1) can be interpreted as the sum of two parts: (1) binary representation of the available bitplanes from \(dW_{x,y,n}^{\text{min}}\) to \(dW_{x,y,n}^{\text{max}}\) and (2) the mid-value of the possible space (i.e. \([0, 2^{dW_{x,y,n}^{\text{max}}}]\)) given by the missing bitplanes. For this reason, the range of \(3D \hat{W}_{x,y,n}\) can be calculated as:

\[
3D W_{x,y,n}^{\text{min}} = \pm (3D \hat{W}_{x,y,n} - 2^{dW_{x,y,n}^{\text{min}} - 1}) ,
\]

and

\[
3D W_{x,y,n}^{\text{max}} = \pm (3D \hat{W}_{x,y,n} + 2^{dW_{x,y,n}^{\text{max}} - 1}) .
\]

Similarly, the SPIHT decoding and range calculation process are also implemented on each coefficient in the additional redundancy \(3D R\). For example, we assume the decoding process of the additional redundancy \(3D R_{x,y,n}\) starts from the highest bitplane \(dR_{x,y,n}^{\text{max}}\) and stops at the end of the bitplane \(dR_{x,y,n}^{\text{min}}\). The coefficient \(3D R_{x,y,n}\) can be initially approximated by the 3-D SPIHT decoder as

\[
3D R_{x,y,n} = \pm \left\{ \sum_{d=dR_{x,y,n}^{\text{min}}}^{dR_{x,y,n}^{\text{max}}} 2^d \times I(d;3D R_{x,y,n}) + 2^{dR_{x,y,n}^{\text{min}} - 1} \right\} ,
\]

where function \(I(d;3D R_{x,y,n})\) is the value of the coefficient \(3D R_{x,y,n}\) at the bitplane \(d\).
The range of $3D \hat{R}_{x,y,n}$ can be found as:

$$3D R_{x,y,n}^{\text{min}} = \pm (3D \hat{R}_{x,y,n} - 2^{dR_{x,y,n}^{\text{min}} - 1}),$$  \hspace{1cm} (5.2.5)

and

$$3D R_{x,y,n}^{\text{max}} = \pm (3D \hat{R}_{x,y,n} + 2^{dR_{x,y,n}^{\text{max}} - 1}).$$  \hspace{1cm} (5.2.6)

### 5.2.2 Bilinear Interpolation on Root Subband

After the range of $3D \hat{W}_{x,y,n}$ is calculated, the next step is to apply bilinear interpolation on the spatial root subband. This process is only needed when the approximation $3D \hat{W}_{x,y,n}$ in (5.2.1) is not a good initial estimate of $3D W_{x,y,n}$. For example, if $dW_{x,y,n}^{\text{min}} > T_2$, we consider that the initial approximation $3D \hat{W}_{x,y,n}$ in (5.2.1) needs to be adjusted by any existing error concealment technique, such as bilinear interpolation that we use in this chapter. If all neighbors of $3D \hat{W}_{x,y,n}$ are taken into consideration (i.e., their corresponding $dW_{x,y,n}^{\text{min}}$ is less than or equal to $T_2$), the estimate $3D \hat{W}_{x,y,n}$ in (5.2.1) is adjusted as:

$$3D \hat{W}_{x,y,n} = \frac{1}{4} \{3D W'_{x-1,y,n} + 3D W'_{x+1,y,n} + 3D W'_{x,y-1,n} + 3D W'_{x,y+1,n}\}. \hspace{1cm} (5.2.7)$$

Although the dispersive grouping method used at the encoder side minimizes the probability that some neighbors of $3D \hat{W}_{x,y,n}$ cannot be used in (5.2.7), this situation can still occur, especially at a high loss rate. In such a context, such neighbors are disregarded and are not taken into the average.
Figure 5.3: Procedure of the range comparison and replacement used in bilinear interpolation and iterative root subband recovery. (a) $3D\tilde{W}_{x,y,n}^\text{min} \leq 3D\tilde{W}_{x,y,n} \leq 3D\tilde{W}_{x,y,n}^\text{max}$. (b) $3D\tilde{W}_{x,y,n} > 3D\tilde{W}_{x,y,n}^\text{max}$. (c) $3D\tilde{W}_{x,y,n} < 3D\tilde{W}_{x,y,n}^\text{min}$.

Note that such adjustment should satisfy the range of $3D\tilde{W}_{x,y,n}$ computed in (5.2.2) and (5.2.3). When $3D\tilde{W}_{x,y,n}$ is not in the range defined by $3D\tilde{W}_{x,y,n}^\text{min}$ and $3D\tilde{W}_{x,y,n}^\text{max}$, we need to follow the procedure depicted in Fig. 5.3 to modify the reconstruction of $3D\tilde{W}_{x,y,n}$. For example, when $3D\tilde{W}_{x,y,n} > 3D\tilde{W}_{x,y,n}^\text{max}$, we set $3D\tilde{W}_{x,y,n} = 3D\tilde{W}_{x,y,n}^\text{max}$ (Fig. 5.3(b)); when $3D\tilde{W}_{x,y,n} < 3D\tilde{W}_{x,y,n}^\text{min}$, we set $3D\tilde{W}_{x,y,n} = 3D\tilde{W}_{x,y,n}^\text{min}$ (Fig. 5.3(c)). It should be emphasized that when $dW_{x,y,n}^\text{min} \leq T_2$, this step can be skipped and the result in (5.2.1) is still taken as the initial estimate, which is then further refined by the proposed root subband recovery as follows.

5.2.3 Iterative Root Subband Recovery

Recall that the subband recovery method in Chapter 4 is based on the fact that the same relationship in (4.1.2) is supposed to still exist between the spatial root subband in $2D\tilde{W}'$ and $2DR'$ on each frame $n$ when all of the substreams are correctly received.
at the decoder and we can write

\[ 2^{D}R'_{x,y,n} \approx \sum_{i} \sum_{j} h_i h_{j} 2^{D}W_{n}^{SR'}(2x - i, y) 2^{D}W_{n}^{SR'}(x, 2y - j). \] (5.2.8)

When the root subband recovery scheme proposed in Chapter 4 is applied, we have

\[ \hat{a}_{x,y,n} = \sum_{i} \sum_{j} h_i h_{j} 2^{D}\hat{W}_{n}^{SR'}(2x - i, y) 2^{D}\hat{W}_{n}^{SR'}(x, 2y - j). \] (5.2.9)

Assume that when the additional redundancy coefficient \( 2^{D}R'_{x,y,n} \) is decoded at a high enough bit rate, the coefficient \( \hat{a}_{x,y,n} \) can be replaced with \( 2^{D}R'_{x,y,n} \) at the decoder and we set \( \hat{a}_{x,y,n} = 2^{D}R'_{x,y,n} \). It should be noted that the success of such replacement depends on the rate at which coefficient \( 2^{D}R'_{x,y,n} \) is decoded. It would tend to improve as the decoding rate is increased. However, a decoding failure may occur at any location of the bitstream. For example, only a few number of bitplane values of \( 2^{D}R'_{x,y,n} \) are available at the decoder. In this case, this replacement may cause an uncontrolled performance degradation.

For this reason, we proposed a revised root subband recovery technique. In contrast to the previous work, the revised error concealment technique is applied on the spatial root subband right after the 3-D SPIHT decoder. Correspondingly, the root subband recovery function in this chapter is deployed in the 3-D wavelet domain. The philosophy of the algorithm is the following. After all of the coefficients are reconstructed by the 3-D SPIHT decoder (or bilinear interpolation), we have the subband
\[ \hat{a}_{x,y,n} = \sum_i \sum_j h_i h_j 3D\hat{W}^{SR}_n(2x - i, y) 3D\hat{W}^{SR}_n(x, 2y - j). \] (5.2.10)

Instead of applying the replacement on \( \hat{a}_{x,y,n} \), we adopt the range information of \( 3D\hat{R}_{x,y,n} \) given by (5.2.5) and (5.2.6). We expect that the coefficient \( \hat{a}_{x,y,n} \) should always be in the range of \( 3D\hat{R}_{x,y,n} \), that is \( 3D\hat{R}_{x,y,n}^{min} \leq \hat{a}_{x,y,n} \leq 3D\hat{R}_{x,y,n}^{max} \).

The revised root subband recovery approach is illustrated in Fig. 5.4. Starting with the top-left 4 x 4 block of Fig. 5.4, an example root subband for a single frame \( n \) in the 3-D wavelet domain is shown with a partially missing coefficient shown in black at location \((x, y)\), i.e., \( 3D\hat{W}_{x,y,n} \). As introduced in Sec. 5.2.1 and 5.2.2, the missing coefficient \( 3D\hat{W}_{x,y,n} \) is first estimated as \( 3D\hat{W}_{x,y,n} \) by the 3-D SPIHT decoder or bilinear interpolation. To simplify the illustration, we assume that the size of \( 3D\hat{W}_n^{SR} \) is 4 x 4 in Fig. 5.4. Correspondingly, the additional redundancy \( 3D\hat{R}_n \) is of size 2 x 2. Similar to (4.1.2), we have the \( \hat{a}_n, \hat{h}_n, \hat{v}_n \) and \( \hat{d}_n \) subbands with wavelet decompositions in the 3-D wavelet domain as follows:

\[
\hat{a}_n : \hat{a}_{x,y,n} = \sum_i \sum_j h_i h_j 3D\hat{W}_n^{SR}(2x - i, y) 3D\hat{W}_n^{SR}(x, 2y - j)
\]

\[
\hat{h}_n : \hat{h}_{x,y,n} = \sum_i \sum_j g_i h_j 3D\hat{W}_n^{SR}(2x - i, y) 3D\hat{W}_n^{SR}(x, 2y - j)
\]

\[
\hat{v}_n : \hat{v}_{x,y,n} = \sum_i \sum_j h_i g_j 3D\hat{W}_n^{SR}(2x - i, y) 3D\hat{W}_n^{SR}(x, 2y - j)
\]

\[
\hat{d}_n : \hat{d}_{x,y,n} = \sum_i \sum_j g_i g_j 3D\hat{W}_n^{SR}(2x - i, y) 3D\hat{W}_n^{SR}(x, 2y - j).
\] (5.2.11)

As discussed above, each coefficient in subband \( \hat{a}_n \) should be always in the range of \( 3D\hat{R}_n \) defined in (5.2.5) and (5.2.6). To guarantee this, we need to compare each coef-
cient in \( \hat{a}_n \) to its corresponding range information. Fig. 5.5 depicts such comparison and possible replacement for different cases. For example, when \( \hat{a}_{x,y,n} > 3^D R_{x,y,n}^{\text{max}} \), we set \( \hat{a}_{x,y,n} = 3^D R_{x,y,n}^{\text{max}} \) (Fig. 5.5(b)); when \( \hat{a}_{x,y,n} < 3^D R_{x,y,n}^{\text{min}} \), we set \( \hat{a}_{x,y,n} = 3^D R_{x,y,n}^{\text{min}} \) (Fig. 5.5(c)). Note that as shown in Fig. 5.5(a), the coefficient \( \hat{a}_{x,y,n} \) keeps unchanged because it is already in the range of \( 3^D \hat{R}_{x,y,n} \) (i.e., \( 3^D R_{x,y,n}^{\text{min}} \leq \hat{a}_{x,y,n} \leq 3^D R_{x,y,n}^{\text{max}} \)). This comparison and possible replacement of \( \hat{a}_n \) is illustrated as step (2) on the right side of Fig. 5.4 where first a 2-D wavelet decomposition is done as step (1) on the spatial root subband of \( 3^D \hat{W}_n^{SR} \) to form the greyed result on the right (greyed because we assume that the partially missing coefficient affects all coefficients in the wavelet decomposition). Then, the revised coefficients shown in white are inserted into the \( \hat{a}_n \) block matrix as shown in the bottom-right 4 x 4 block of Fig. 5.4.

As shown as step (3) in Fig. 5.4, an inverse wavelet transform is then applied (refined \( \hat{a}_n, \hat{h}_n, \hat{v}_n \) and \( \hat{d}_n \)) with a dual low-pass filter \( \tilde{h} \) and a dual high-pass filter \( \tilde{g} \) as below

\[
3^D \hat{W}_{x,y,n}' = \sum_i \sum_j \tilde{h}_i \tilde{h}_j Z(\hat{a}_n) + \sum_i \sum_j \tilde{g}_i \tilde{h}_j Z(\hat{h}_n) + \sum_i \sum_j \tilde{h}_i \tilde{g}_j Z(\hat{v}_n) + \sum_i \sum_j \tilde{g}_i \tilde{g}_j Z(\hat{d}_n), \tag{5.2.12}
\]

where the \( Z \) operator represents the insertion of zeros between two consecutive samples on the refined \( \hat{a}_n, \hat{h}_n, \hat{v}_n \) and \( \hat{d}_n \). The missing coefficient \( 3^D \hat{W}_{x,y,n}' \) in grey in the bottom-left 4 x 4 block of Fig. 5.4 is then used as an improved estimate for \( 3^D W_{x,y,n}' \). It should be emphasized that such estimation can be further adjusted in step (4) by using the range information of \( 3^D W_{x,y,n} \). This process is similar to Fig.
Figure 5.4: Illustration of the proposed iterative root subband recovery method for the partial substream loss case.
Figure 5.5: Procedure of the range comparison and possible replacement used in iterative root subband recovery. (a) $3D_R^{\min} < 3D\hat{a}_{x,y,n} < 3D_R^{\max}$, (b) $3D\hat{a}_{x,y,n} > 3D_R^{\max}$, (c) $3D\hat{a}_{x,y,n} < 3D_R^{\min}$.

5.3 depicted in Sec. 5.2.2: when $3D\hat{W}^*_{x,y,n} < 3D\hat{W}^{\min}_{x,y,n}$, we set $3D\hat{W}^*_{x,y,n} = 3D\hat{W}^{\min}_{x,y,n}$, when $3D\hat{W}^*_{x,y,n} > 3D\hat{W}^{\max}_{x,y,n}$, we set $3D\hat{W}^*_{x,y,n} = 3D\hat{W}^{\max}_{x,y,n}$.

Note that the process illustrated in Fig. 5.4 can be iterated a number of times. With each iteration, the estimate is gradually improved, up to the limit of what the additional redundancy can reconstruct. Convergence usually occurs after only a few iterations, with diminishing gains achieved with further iterations. In this case, the coefficient $\hat{a}_{x,y,n}$ is already in or very close to the range of $3D\hat{R}_{x,y,n}$. It should be noted that the procedures used in this revised error concealment technique can also be applied to the full substream loss case. In such a context, the decoding range of $3D\hat{W}_{x,y,n}$ can be set as: $3D\hat{W}^{\min}_{x,y,n} = -\infty$ and $3D\hat{W}^{\max}_{x,y,n} = +\infty$. The range comparison and replacement used in bilinear interpolation and iterative root subband recovery (as the step (4) in Fig. 5.4) can be skipped.
5.3 Simulation Results

In this section, we conduct experiments to investigate the performance of the proposed MD video coding algorithm in the partial substream loss case. We compare the performance of the proposed algorithm to ERC-SPIHT [40,41] for estimating the missing bitplane values.

5.3.1 Experimental Data

The experiments are conducted using the 352 x 240 x 48 monochrome “Football” and “Susie” video sequences with frame rate of 30 frames/s. We use 16 frames for the GOF. A three-level decomposition using the CDF 9/7 filter is applied in both the spatial and temporal domains with reflection extensions both at each image boundary and each GOF boundary. In this chapter, the CDF 9/7 wavelet transform is used to generate the additional redundancy. Therefore, the size of the additional redundancy is 22 x 15 x 48. The wavelet coefficients and additional redundancy are divided into multiple groups (S = 16). Each group is independently encoded at a coding rate of 1.0 bpp. We set $c^R = 8.0$ bpp as the additional redundancy coding rate for the “Football” sequence and $c^R = 5.0$ bpp for the “Susie” sequence. Correspondingly, the total transmission rate is 352 x 240 x 30 x 1.0 = 2.53 Mbps. Among the substreams, the transmission rate for the additional redundancy is $22 \times 15 \times 30 \times 8.0 = 0.08$ Mbps for the “Football” sequence and $22 \times 15 \times 30 \times 5.0 = 0.05$ Mbps for the “Susie” sequence, respectively. It has been shown in Chapter 4 that the performance gain achieved by the proposed iterative root subband recovery stops after about 50 iterations. Therefore, we choose
$I = 50$ for the proposed method. In addition, the empirical values obtained for $T_1$ and $T_2$ are found to be 5 and 10, respectively. Among the different values of $T_1$ and $T_2$, $T_1 = 5$ and $T_2 = 10$ provide the best reconstruction results in terms of PSNR and MSSIM.

5.3.2 Error Concealment Performance at Different Bitplanes

Figs. 5.6 and 5.7 demonstrate the comparison of PSNR and MSSIM values with the proposed MD video coder and ERC-SPIHT in noisy channels for the “Football” and “Susie” sequences, respectively. In these figures, 16 substreams are generated and substream-1 is corrupted at the end of different bitplanes $d$. As we can see from the simulation results, the proposed algorithm outperforms ERC-SPIHT in the partial substream loss case (when $d$ is between 6 and 12). We can observe that the performance improvement obtained by the proposed method is more significant when $d$ is between 9 and 12. For example, the improvement is up to 1.72 dB in terms of PSNR and 0.0102 in terms of MSSIM on the “Football” sequence. For the “Susie” sequence, the improvement is up to 7.82 dB in terms of PSNR and 0.0441 in terms of MSSIM. Such performance gain is obtained by the fact that the missing bitplane values of the coefficients in the spatial root subband can be reconstructed by the revised error concealment technique from the inherent/additional redundancy in conjunction with the decoding range information. This is in contrast to ERC-SPIHT: bilinear interpolation is only applied for the coefficients completely lost; when the coefficients are partially missing, they are only reconstructed by the 3-D SPIHT decoder using
(5.2.1) without error concealment. The performance for the full substream loss case is also included in Figs. 5.6 and 5.7 (corresponding to the results when $d = 13$). Compared to the full substream loss case, the reconstructed video quality obtained by ERC-SPIHT gets worse when a few number of bitplane values are received (such as $d = 10, 11, \text{and } 12$). The lower PSNR and MSSIM values for ERC-SPIHT are mainly due to the approximation of the partially lost coefficients in (5.2.1). In this condition, the accuracy in estimating the partially lost coefficients in (5.2.1) is worse than that of bilinear interpolation.

### 5.3.3 Error Concealment Performance at Different Packet Loss Rates

We now investigate the error concealment performance of the proposed MD video coder in error-prone packet-based networks. To simplify the task, we focus on a memoryless bit-error-free but packet loss environment in our study. We use a simple two-state Markov model (i.e., Gilbert-Elliott model) [82, 83] to simulate the burst packet loss behavior. As illustrated in Fig. 5.8, all packets are correctly received in the good state “G”, while packets are assumed to be lost in the bad state “B”. This model can be specified by transition probabilities: $p$ of going from state G to state B and $q$ of going from state B to G. Then the average packet loss probability is given by

$$P_L = \frac{p}{p + q}. \quad (5.3.13)$$
Figure 5.6: Comparison of PSNR (dB) and MSSIM of “Football” video sequence in noisy channels ($S = 16$ substreams with 1 substream corrupted at the end of different bitplanes $d$, $c = 1.0$ bpp and $c^R = 8.0$ bpp). (a) PSNR (dB). (b) MSSIM.
Figure 5.7: Comparison of PSNR (dB) and MSSIM of “Susie” video sequence in noisy channels ($S = 16$ substreams with 1 substream corrupted at the end of different bitplanes $d$, $c = 1.0$ bpp and $c^R = 5.0$ bpp). (a) PSNR (dB). (b) MSSIM.
And the average length of burst errors is given by

\[ L_B = \frac{1}{q}. \] (5.3.14)

In our simulations, the average burst error length \( L_B \) is set to 5 (i.e., \( q = \frac{1}{5} \)). Different packet loss rates \( P_L \) can be obtained by adjusting the value of \( p \).

In addition to the two-state Markov model, different approaches have been proposed to model packet losses. For example, the burst length distribution is divided into multiple segments. In each segment, the distribution is set to a modified exponential or Pareto function [84]. Another method to model packet losses is to use higher order Markov process [85]. These techniques are usually more accurate than the two-state Markov model at the expense of higher computational cost. Since our focus is on the error concealment performance of the proposed video coder and not on the model for packet losses, we adopt the two-state Markov model in this work. We should note that the proposed video coder could be used with other packet loss
Tables 5.1 and 5.2 show the comparison of PSNR and MSSIM values with the proposed algorithm and ERC-SPIHT at different packet loss rates ($P_L = 0\%, 0.5\%, 1\%, 5\%, 10\%$ and $20\%$) for the “Football” and “Susie” sequences, respectively. In these tables, each substream is packetized with fixed packet length $L = 200$ bits [39–41]. All final PSNR and MSSIM values reported are averaged over 50 independent runs. As seen from the simulation results, the performance of the proposed video coder is better than that of ERC-SPIHT when the channel is noisy (such as the case of $P_L = 0.5\%, 1\%, 5\%, 10\%$ and $20\%$). In addition, such performance improvement is much more noticeable at high packet loss rates. For example, the amount of improvement obtained by the proposed video coder is about $3.46$ dB in terms of PSNR and $0.0536$ in terms of MSSIM at $P_L = 20\%$ for the “Football” sequence, and for the “Susie” sequence, the amount of the improvement is $2.74$ dB in terms of PSNR and $0.0822$ in terms of MSSIM. The PSNR and MSSIM values in error-free condition are also included in Tables 5.1 and 5.2 (corresponding to the results when $P_L = 0\%$). There are just small losses in performance with the proposed algorithm compared to ERC-SPIHT. These small losses result from the additional redundancy overhead that is inserted into the substreams by a careful coding strategy in our proposed work such that the penalty in performance degradation is limited.

The PSNR and MSSIM values at different packet loss rates ($P_L = 0\%, 0.5\%, 1\%, 5\%, 10\%$ and $20\%$) with $L = 2000$ bits are summarized in Tables 5.3 and 5.4. Similar to Tables 5.1 and 5.2, final PSNR and MSSIM values reported are averaged over 50 independent runs. We can observe that the proposed MD video coder enhances the
Table 5.1: Comparison of PSNR (dB) and MSSIM of “Football” video sequence under different packet loss rates ($S = 16$ substreams, $c = 1.0$ bpp, $c^R = 8.0$ bpp and $L = 200$ bits)

(a) PSNR (dB)

<table>
<thead>
<tr>
<th>$P_L$</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>33.16</td>
<td>32.96</td>
</tr>
<tr>
<td>0.005</td>
<td>29.48</td>
<td>29.86</td>
</tr>
<tr>
<td>0.01</td>
<td>28.04</td>
<td>28.50</td>
</tr>
<tr>
<td>0.05</td>
<td>23.06</td>
<td>23.99</td>
</tr>
<tr>
<td>0.1</td>
<td>20.57</td>
<td>22.23</td>
</tr>
<tr>
<td>0.2</td>
<td>17.19</td>
<td>20.66</td>
</tr>
</tbody>
</table>

(b) MSSIM

<table>
<thead>
<tr>
<th>$P_L$</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.9098</td>
<td>0.9068</td>
</tr>
<tr>
<td>0.005</td>
<td>0.8497</td>
<td>0.8510</td>
</tr>
<tr>
<td>0.01</td>
<td>0.8080</td>
<td>0.8116</td>
</tr>
<tr>
<td>0.05</td>
<td>0.5842</td>
<td>0.6008</td>
</tr>
<tr>
<td>0.1</td>
<td>0.4595</td>
<td>0.4919</td>
</tr>
<tr>
<td>0.2</td>
<td>0.3508</td>
<td>0.4044</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of PSNR (dB) and MSSIM of “Susie” video sequence under different packet loss rates ($S = 16$ substreams, $c = 1.0$ bpp, $c^R = 5.0$ bpp and $L = 200$ bits)

(a) PSNR (dB)

<table>
<thead>
<tr>
<th>$P_L$</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>43.66</td>
<td>43.55</td>
</tr>
<tr>
<td>0.005</td>
<td>38.60</td>
<td>39.55</td>
</tr>
<tr>
<td>0.01</td>
<td>36.91</td>
<td>38.07</td>
</tr>
<tr>
<td>0.05</td>
<td>30.30</td>
<td>32.27</td>
</tr>
<tr>
<td>0.1</td>
<td>27.00</td>
<td>29.35</td>
</tr>
<tr>
<td>0.2</td>
<td>24.04</td>
<td>26.78</td>
</tr>
</tbody>
</table>

(b) MSSIM

<table>
<thead>
<tr>
<th>$P_L$</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.9767</td>
<td>0.9761</td>
</tr>
<tr>
<td>0.005</td>
<td>0.9543</td>
<td>0.9593</td>
</tr>
<tr>
<td>0.01</td>
<td>0.9393</td>
<td>0.9473</td>
</tr>
<tr>
<td>0.05</td>
<td>0.8466</td>
<td>0.8742</td>
</tr>
<tr>
<td>0.1</td>
<td>0.7709</td>
<td>0.8225</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6833</td>
<td>0.7655</td>
</tr>
</tbody>
</table>
Table 5.3: Comparison of PSNR (dB) and MSSIM of “Football” video sequence under different packet loss rates ($S = 16$ substreams, $c = 1.0$ bpp, $c^R = 8.0$ bpp and $L = 2000$ bits)

<table>
<thead>
<tr>
<th>$P_L$</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>33.16</td>
<td>32.96</td>
</tr>
<tr>
<td>0.005</td>
<td>32.23</td>
<td>32.19</td>
</tr>
<tr>
<td>0.01</td>
<td>31.97</td>
<td>31.96</td>
</tr>
<tr>
<td>0.05</td>
<td>28.75</td>
<td>29.20</td>
</tr>
<tr>
<td>0.1</td>
<td>26.12</td>
<td>26.76</td>
</tr>
<tr>
<td>0.2</td>
<td>23.64</td>
<td>24.31</td>
</tr>
</tbody>
</table>

(b) MSSIM

<table>
<thead>
<tr>
<th>$P_L$</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.9098</td>
<td>0.9068</td>
</tr>
<tr>
<td>0.005</td>
<td>0.8993</td>
<td>0.8968</td>
</tr>
<tr>
<td>0.01</td>
<td>0.8944</td>
<td>0.8946</td>
</tr>
<tr>
<td>0.05</td>
<td>0.8350</td>
<td>0.8370</td>
</tr>
<tr>
<td>0.1</td>
<td>0.7578</td>
<td>0.7639</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6423</td>
<td>0.6522</td>
</tr>
</tbody>
</table>

reconstructed video quality at the decoder in most of the noisy channel cases. For example, the performance improvement obtained by the propose method is up to 0.67 dB in terms of PSNR and 0.0099 in terms of MSSIM on the “Football” sequence, and in the case of the “Susie” sequence the difference is up to 1.01 dB in terms of PSNR and 0.0086 in terms of MSSIM. In addition, we can see from the results in Tables 5.1–5.4 that the PSNR and MSSIM improvement obtained by the proposed method is easier to perceive when $L = 200$ bits.

5.3.4 Visual Comparison

Figs. 5.9 and 5.10 show sample frames to compare the visual results obtained in Figs. 5.6 and 5.7 when $d = 11$. As we can see, with ERC-SPIHT (Fig. 5.9(b) and Fig. 5.10(a)
Table 5.4: Comparison of PSNR (dB) and MSSIM of “Susie” video sequence under different packet loss rates ($S = 16$ substreams, $c = 1.0$ bpp, $c^R = 5.0$ bpp and $L = 2000$ bits)

<table>
<thead>
<tr>
<th>$P_L$</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>43.66</td>
<td>43.55</td>
</tr>
<tr>
<td>0.005</td>
<td>42.26</td>
<td>42.43</td>
</tr>
<tr>
<td>0.01</td>
<td>41.99</td>
<td>42.20</td>
</tr>
<tr>
<td>0.05</td>
<td>37.41</td>
<td>38.22</td>
</tr>
<tr>
<td>0.1</td>
<td>33.84</td>
<td>34.96</td>
</tr>
<tr>
<td>0.2</td>
<td>30.59</td>
<td>31.61</td>
</tr>
</tbody>
</table>

(b) MSSIM

<table>
<thead>
<tr>
<th>$P_L$</th>
<th>ERC-SPIHT</th>
<th>Proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.9767</td>
<td>0.9761</td>
</tr>
<tr>
<td>0.005</td>
<td>0.9728</td>
<td>0.9729</td>
</tr>
<tr>
<td>0.01</td>
<td>0.9705</td>
<td>0.9714</td>
</tr>
<tr>
<td>0.05</td>
<td>0.9505</td>
<td>0.9534</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9215</td>
<td>0.9277</td>
</tr>
<tr>
<td>0.2</td>
<td>0.8761</td>
<td>0.8847</td>
</tr>
</tbody>
</table>

5.10 (b)), there are many white spots in the frames. Fig. 5.9(c) and Fig. 5.10 (c) show that the reconstruction of these spot are significantly improved by the proposed method. Such difference agrees with the objective PSNR and MSSIM evaluation.

5.4 Conclusions

We present an error-resilient MD video coding algorithm built upon ERC-SPIHT for partial substream loss. Our proposed algorithm injects additional redundancy into the substreams so that the missing bitplane values of the coefficients in the spatial root subband can be reconstructed with a high level of accuracy. Such reconstruction procedure is established on three types of resources (1) inherent redundancy, (2) added redundancy, and (3) partially received substreams. The proposed method has
Figure 5.9: 352 × 240 “Football” video sequence (S = 16 substreams with 1 substream lost after the bitplane d = 11, c = 1.0 bpp and cR = 8.0 bpp). (a) Original sequence (frame 47). (b) ERC-SPIHT: $PSNR = 27.01$ dB and $MSSIM = 0.8259$. (c) Proposed algorithm: $PSNR = 28.18$ dB and $MSSIM = 0.8344$. 
Figure 5.10: 352 × 240 “Susie” video sequence (\(S = 16\) substreams with 1 substream lost after the bitplane \(d = 11\), \(c = 1.0\) bpp and \(c^R = 5.0\) bpp). (a) Original sequence (frame 27). (b) ERC-SPIHT: \(PSNR = 31.70\) dB and \(MSSIM = 0.9240\). (c) Proposed algorithm: \(PSNR = 38.44\) dB and \(MSSIM = 0.9586\).
been shown to be more resilient in an error-prone transmission environment when the substreams are partially or completely lost. The proposed approach achieves a significant improvement on video quality in the presence of substream loss when compared to its counterpart ERC-SPIHT.
Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, we propose an error-resilient domain-partitioning based MD video algorithm for ERC-SPIHT-coded video with added redundancy sent across error-prone networks. In contrast to existing schemes, the novelty of the proposed approach is that the spatial root subband coefficients lost during transmission in any substream can be reconstructed by exploiting both inherent redundancy and inserted redundancy.

In our work, the proposed video coding algorithm is designed in three progressive stages. In stage (1), we concentrate on a simple scenario in which four substreams are created and only one of the substreams is allowed lost due to transmission errors. In this case, the additional redundancy is obtained by averaging $2 \times 2$ spatially neighboring coefficients in the spatial root subband and such redundancy is then intentionally inserted into the end of the substreams in reverse order. As a result, the missing coefficients in the spatial root subband can be recovered directly from the
added redundancy. The performance of the proposed MD video coding approach is examined with respect to source coding efficiency and error concealment performance. Experimental results on different video sequences show that although the proposed algorithm provides lower source coding efficiency than ERC-SPIHT in lossless transmission, the proposed method is more resilient to transmission errors in an error-prone transmission environment. In particular, our results demonstrate that the proposed MD video coding approach achieves a significant improvement on video quality by up to 1.69 dB compared to ERC-SPIHT when one substream is lost.

Inspired by the success of the work in stage (1), we generalize the MD based video coding algorithm for the generation of an arbitrary number of substreams $S$. In such a context, the additional redundancy is generated by a wavelet projection method: each frame of the spatial root subband is spatially decomposed by one additional transform level and the resulting coefficients in the approximation subband are grouped together in a 3-D structure to form the additional redundancy. A natural question that arises is how to use such additional redundancy to reconstruct the missing spatial root subband coefficients in conjunction with the inherent redundancy. In stage (2), we focus on the full substream loss case and the reconstruction procedure is implemented by two steps: first by using existing 2-D error concealment techniques (such as bilinear interpolation), and second with the proposed root subband recovery approach. The former step is used to estimate the missing coefficients in the spatial root subband and high-frequency subbands by exploiting the inherent redundancy, while the latter attempts to utilize the inserted redundancy to further improve the precision in the estimation of the missing spatial roots subband coefficients. The pro-
posed root subband recovery method can be iteratively applied and accuracy of the reconstruction can be gradually increased with each iteration. Similar to stage (1), the performance of the generalized MD video coding approach is also evaluated with respect to source coding efficiency and error concealment performance. We carry our experiments for different coding rates $c$ under different number of substreams lost. From the simulation results, we can see that the proposed video coding algorithm maintains error-resilience with high coding efficiency. Specifically, the proposed algorithm enhances the reconstructed video quality significantly in the full substream loss case by up to 2.58 dB compared to ERC-SPIHT. Such performance gain achieved by the proposed method noticeably improves as $c$ increases. We also investigate the performance of the proposed method with the additional redundancy generated by two different wavelet transform filters (Haar and CDF 9/7). The experiments show that, depending on the missing pattern of coefficients, one wavelet transform yields a better reconstructed video quality than the other.

In stage (3), we extend the generalized MD based coding technique for video transmission over packet networks. In order to achieve this, we need to revise the video coding algorithm and make it suited for the partial substreams loss. In this context, the additional redundancy is generated by the wavelet projection method and the main modification of the video coder is the incorporation of the SPIHT decoding range concept to the system. As a result, missing bitplane values of the spatial root subband coefficients in any substream can be reconstructed from three types of resources: (1) inherent redundancy, (2) added redundancy, and (3) SPIHT decoding range. This reconstruction procedure is implemented in three steps: (1) range cal-
culation, (2) existing error concealment techniques (such as bilinear interpolation), and (3) root subband recovery. The first two functions (1) and (2) use the inherent redundancy and the partially received information to define the decoding ranges and initially estimate the missing bitplane values, while the function (3) attempts to improve the accuracy in recovering the missing bitplane values by exploiting the added redundancy and decoding ranges. We compare our revised video coding algorithm to ERC-SPIHT. The simulation results show that our revised video coder gives a higher level of error resilience and achieves better numerical and visual quality compared to existing coding schemes for video transmission over error-prone packet networks. We can observe that the PSNR and MSSIM improvement is easier to perceive when a small packet length $L$ is used (such as $L = 200$ bits in the simulations).

### 6.2 Future Work

#### 6.2.1 Error Concealment on High Frequency Coefficients

At present, we only focus on recovering the missing coefficients in the spatial root subband. We do not attempt to conceal the missing coefficients in the high-frequency subbands, that is, we set the missing coefficients in the high-frequency subbands to zero in our experiments. For optimal error concealment, the recovery of the missing coefficients in the high-frequency subbands is essential.

Many algorithms have been proposed to interpolate the coefficients in the high-frequency subband for 2-D image. In [86], a spatial-domain error concealment ap-
proach was applied using linear least-squared error (LLSE) estimation, where the missing blocks are interpolated from the surrounding coefficients. In [87], a deterministic least-squared model for wavelet transformed images was introduced. The interpolation was performed with block classification to preserve the edges. In [21], a maximum a posterior (MAP) technique was applied using a Markov random field prior in each subband.

Although these error concealment approaches have shown to be efficient in the reconstruction of the high-frequency subbands, they may still cause reduced presentation quality since they rely on completely on the inherent redundancy present in the signal. To further improve the quality of the reconstructed video, we need to extend the proposed video coder to the high-frequency subbands. We will examine the performance of our proposed system in conjunction with existing error concealment techniques geared toward recovery of high-frequency subband coefficients. Hence, there are two important design issues: (1) how to generate the additional redundancy and (2) how to make the best use of the added redundancy to reconstruct the the missing coefficients in the high-frequency subbands. We need to consider both the source code efficiency and the error concealment performance. We will work on this topic in the future.

6.2.2 Block-Based Hybrid Video Coding

In addition to SPIHT, some block-based hybrid video coding standards (such as MPEG-2 [88], MPEG-4 [89], H.263 [90], and H.264 [91,92]) are widely employed in
modern video compression system. The key feature of block-based hybrid video coding is that block-based motion-compensated prediction (MCP) is applied in conjunction with the discrete cosine transform (DCT). In general, the block-based hybrid video coding has two basic coding modes: intra-frame coding and inter-frame coding. In the intra-frame coding mode, a frame are divided into blocks of pixels. Each block is transformed by the DCT, and the resulting transformed coefficients are quantized and entropy coded using variable length coding (VLC). In the inter-frame coding mode, block-based hybrid video coding mainly consists of two parts: motion estimation and prediction error coding. More specially, a video frame is first partitioned into blocks of pixels. Each block is temporally predicted from a best matching block of equal size in a previously coded frame, called reference frame. The process to find the best matching block in the reference frame is referred to as motion estimation. The difference in motion between the current block and its matching block in the reference frame is defined as the motion vector (MV). The use of temporal prediction is based on the fact that the consecutive frames in a video sequence will be substantially similar except for changes caused by objects moving within the frames. Thus, coding the prediction error block will require fewer bits to code than the original block in the video frame. However, the prediction error block still contains considerable amount of spatial redundancy. To reduce such spatial redundancy, the prediction error block is encoded in a manner similar to the one in the intra-frame mode, and sent to the decoder along with the MV. Although block-based hybrid video coding achieves high compression efficiency, it is highly susceptible to transmission errors. It is due to the fact that the reconstruction of a frame depends on the successful reconstruction of its
reference frame. If one frame is lost or corrupted due to transmission errors, it will not only affect the reconstruction quality of the current frame but it will also propagate to the following frames even when all of the following frames are received correctly. Without effective control of temporal error propagation, reconstruction quality can become seriously degraded.

In this work, we design a domain-partitioning based MD coding framework in which the missing coefficients can be initially estimated from the inherent redundancy and further refined from the additional redundancy. We apply this framework to ERC-SPIHT-coded video. There is naturally great interest in carrying over this coding framework into the block-based hybrid video coding realm. Several design issues need to be solved, such as (1) how to generate the additional redundancy and (2) how to make the best use of the added redundancy to reconstruct the the missing coefficients in conjunction with the inherent redundancy. We will study this topic in future work.
Bibliography


