

DISTRIBUTED SOURCE CODING FOR IMAGE AND
VIDEO APPLICATIONS

by

Ngai-Man Cheung

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

May 2008

Copyright 2008

Ngai-Man Cheung

Dedication

To my wife Tao.

Acknowledgements

I would like to thank my advisor Prof. Antonio Ortega for his guidance and suggestions throughout the course. I will always remember his enthusiasm about research discussion. And thanks for being critical of my work, which at the end I do feel grateful. I would also like to thank Prof. C.-C. Jay Kuo and Prof. Aiichiro Nakano for being the faculty members in my dissertation committee, and Prof. Sanjit Mitra and Prof. Zhen Zhang for being the faculty members in my qualifying exam committee. It is indeed a great privilege to have their valuable comments and feedbacks on my work.

I would like to thank my mentors Dr. Yiliang Bao, Dr. Dake He, and Dr. Ashish Jagmohan for making my internship experience at Nokia Research Center and IBM Research pleasant and productive. I would also like to thank Dr. Sam Dolinar, Dr. Aaron Kiely, Dr. Hua Xie and Dr. Matthew Klimesh of NASA-JPL for helpful research discussion. Thanks also to all my group members and friends for their help, cares, encouragements and feedbacks.

I would like to thank my parents for their unconditional love and support. It is always my regret not being able to spend much time with you these years. I would like to thank my son Long. It has been pleasure seeing him growing up. Finally, I would like to express my deepest gratitude to my wife Tao. There were many times when I felt

tired and wanted to give up. Without your support and care I can never face all the difficulties. I feel indebted for having you sacrificing so much, and worrying so much, but you never feel tired of me. I am happy we can hang in there until the end, and life has been pleasant with you together.

Contents

Dedication	ii
Acknowledgements	iii
List Of Tables	viii
List Of Figures	ix
Abstract	xvi
1 Introduction	1
1.1 Motivation	1
1.2 Distributed Source Coding	6
1.2.1 Slepian-Wolf Theorem	6
1.2.2 Constructive Coding Algorithm	7
1.2.2.1 Coset Example	8
1.2.2.2 Main Ideas and General Steps	9
1.2.2.3 Practical Algorithms Using Error Correcting Codes	10
1.2.3 Properties of Distributed Source Coding	12
1.2.3.1 Encoding Requiring Correlation Information Only	12
1.2.3.2 Systems Robust to Uncertainty on Predictor	15
1.3 Hyperspectral Imagery	16
1.3.1 Review of Hyperspectral Imagery Compression Approaches	17
1.3.1.1 Inter-band Prediction Approaches	17
1.3.1.2 3D Wavelet Approaches	19
1.4 Contributions of This Research	20
2 Efficient Wavelet-based Predictive Slepian-Wolf Coding for Hyperspectral Imagery	23
2.1 Introduction	23
2.1.1 Motivation	23
2.1.2 Our Contributions and Related Works	24
2.2 System Overview	28
2.3 Codec Architecture	29
2.4 Estimating the Correlation and Encoder Complexity Comparison	36

2.4.1	Estimation of Predictor Coefficients and Correlation	38
2.4.1.1	Estimation of Predictor Coefficients	38
2.4.1.2	Estimation of Crossover Probability	39
2.4.2	Encoder Complexity Comparison	41
2.4.2.1	Comparison with Inter-band Prediction	42
2.4.2.2	Comparison with 3D Wavelet Approaches	45
2.5	Experimental Results of Baseline Codec	45
2.5.1	Rate-Distortion Comparison with 3D Wavelet Approaches	48
2.5.2	Rate-Distortion Comparison with 2D Wavelet Approaches	51
2.5.3	Preservation of Spectral Signature	56
2.6	Adaptive Coding Strategy	57
2.6.1	Refinement/Sign Bits Compression	59
2.6.2	Significance Maps Compression and Raw Bitplane Coding	62
2.6.3	Modeling	63
2.6.4	Adaptive Coding Scheme	65
2.7	Experimental Results of Adaptive Codec	66
2.8	Conclusions	67
3	Flexible Video Decoding: A Distributed Source Coding Approach	72
3.1	Introduction	72
3.1.1	Motivation	72
3.1.2	Flexible Decoding: Challenges	73
3.1.3	Our Contributions and Related Work	78
3.2	Flexible Decoding Based on DSC: Intuition	80
3.3	Theoretical Performance	82
3.4	Proposed Algorithms	85
3.4.1	Motion Estimation and Macroblock Classification	85
3.4.2	Direct Coefficient Coding (DCC)	89
3.4.3	Significant Coefficient Coding (SCC)	90
3.4.4	Bit-plane Compression	91
3.4.5	Model and Conditional Probability Estimation	92
3.5	Application Scenarios	94
3.6	Experimental Results and Discussion	95
3.6.1	Viewpoint Switching in Multiview Video Coding	95
3.6.2	Forward/Backward Video Playback	98
3.7	Conclusions and Future Work	99
4	Correlation Estimation for Distributed Source Coding Under Rate and Complexity Constraints Using Sampling-based Techniques	103
4.1	Introduction	103
4.1.1	Motivation	103
4.1.2	Correlation Information Models in Practical DSC Applications	104
4.1.3	Correlation Estimation in Distributed Source Coding	106
4.1.4	Our Contributions and Related Work	108
4.2	Single Binary Source: Rate Penalty Analysis	112

4.2.1	Problem Definition	113
4.2.2	Correlation Estimation	114
4.2.3	Rate Penalty Analysis	115
4.2.4	Experiment	116
4.3	Multiple Binary Sources: Rate Penalty Analysis and Sample Allocation	118
4.3.1	Problem Definition	119
4.3.2	Optimal Sample Allocation Strategy	121
4.3.3	Rate Penalty Analysis - Multiple Sources	124
4.3.4	Experiments	124
4.4	Continuous Input Source: Model-based Estimation	125
4.4.1	General Approach	127
4.4.2	Experiments	128
4.5	Model-based Estimation on Structured Bit-planes	131
4.5.1	Model-based Estimation for Refinement/Sign Bit-planes	134
4.5.2	Model-based Estimation for Wavelet-based Applications	135
4.5.2.1	Estimation with Adequate Samples	136
4.5.2.2	Estimation without Adequate Samples	137
4.6	Hyperspectral Image Compression Experiments	139
4.6.1	DSC-based Hyperspectral Image Compression	140
4.6.2	Sample Allocation Experiments	141
4.6.3	Model-based Estimation Experiments	143
4.7	Extensions to Other Correlation Models	146
4.7.1	Rate Penalty Model for Non-equiprobable Input	146
4.7.2	Model-based Estimation with Previously Decoded Bit-planes as Side-information	149
4.7.3	Model-based Estimation with Continuous Side Information in Joint Decoding	151
4.8	Conclusions	153
5	Conclusions and Future Work	155
Appendix A		
	Derivation of Bounds on the Estimation Error	158
Appendix B		
	Derivation of Optimal Sample Allocation	160
Bibliography		162

List Of Tables

3.1	Comparison of DSC-based low-complexity encoding and flexible decoding.	80
4.1	Kolmogorov-Smirnov (K-S) tests statistics for ΔH : Mobile DC, QP=24. Numbers in parenthesis indicate cases that K-S statistics are larger than the 1% significance cutoff value, 0.1608, and therefore do not pass K-S tests. Those are cases when $n \cdot p \geq 4$ does not hold. Note that bit position 6 corresponds to a more significant bit-plane.	118

List Of Figures

1.1	Compression using CLP. Encoder computes the difference between the input source X and the predictor Y and communicates the difference (prediction residue) to the decoder.	2
1.2	Slepian-Wolf theorem.	7
1.3	An example using coset to illustrate the main ideas of constructive DSC algorithms.	9
1.4	Practical coding algorithms using error correcting codes.	11
1.5	Decoding error due to inaccurate correlation information.	14
1.6	Robustness property of DSC.	16
1.7	Hyperspectral image.	17
1.8	Mean square residuals after simple image alignment and subtraction.	18
2.1	The proposed parallel encoding system. Each processor compresses one spatial image at a time.	29
2.2	Block diagram of SW-SPIHT.	31
2.3	Bit-plane coding in SW-SPIHT.	31
2.4	Encoding using proposed system.	33
2.5	Example of conditional entropy of different bit-planes.	36
2.6	MSE under first-order linear predictor for a typical spectral band.	39

2.7	Example of estimation of crossover probability. Note that for all bit-planes it is possible to achieve an accurate estimate using a small percentage of data in a bit-plane.	41
2.8	Encoding using inter-band prediction.	43
2.9	Rate-distortion curves of SW-SPIHT and predictive 3D-SPIHT - Cuprite.	48
2.10	Rate-distortion curves of SW-SPIHT and predictive 3D-SPIHT - Moffet Field.	49
2.11	Rate-distortion curves of SW-SPIHT and predictive 3D-SPIHT - Lunar Lake.	49
2.12	Inter-band SNR fluctuation under 3D-SPIHT.	51
2.13	Rate-distortion curves of SW-SPIHT and predictive 2D-SPIHT - Cuprite.	52
2.14	Rate-distortion curves of SW-SPIHT and predictive 2D-SPIHT - Moffet Field.	53
2.15	Rate-distortion curves of SW-SPIHT and predictive 2D-SPIHT - Lunar Lake.	53
2.16	Rate-distortion curves of SW-SPIHT and SPIHT (Site: Cuprite, View: SC01, Band: 40).	54
2.17	Rate-distortion curves of SW-SPIHT and SPIHT (Site: Cuprite, View: SC01, Band: 133).	55
2.18	Rate-distortion curves of SW-SPIHT and SPIHT (Site: Cuprite, View: SC01, Band: 190).	55
2.19	Classification performance (Site: Lunar Lake, View: SC02).	56
2.20	Bit extraction.	57
2.21	Coding configurations.	58
2.22	Probability that a refinement/sign bit being zero or one.	60
2.23	Events of refinement/sign bits crossover.	61
2.24	Bias in significance maps.	62

2.25	Raw bit-planes crossover.	63
2.26	Modeling results.	69
2.27	Adaptive coding scheme.	69
2.28	Coding performance: Cuprite. Correlation information is estimated by a model-based approach discussed in Chapter 4.	70
2.29	Coding performance: Lunar. Correlation information is estimated by a model-based approach discussed in Chapter 4.	71
3.1	Multiview video applications - viewpoint switching may require a compression scheme to support several different decoding orders: (a) users stay in the same view during playback; (b), (c) users switch between adjacent views during playback.	74
3.2	Forward and backward frame-by-frame playback.	74
3.3	Robust video transmission using multiple decoding paths.	74
3.4	Problem formulation for flexible decoding. Either one of the candidate predictors Y_0, Y_1, \dots, Y_{N-1} will be present at the decoder, but encoder does not know which one.	76
3.5	Compression of input source X : (a) CLP coding; (b) DSC from the virtual channel perspective; (c) DSC approach to the flexible decoding problem.	82
3.6	Source networks: (a) Source network of Slepian-Wolf [21,63]. Csiszar and Korner [20,21] suggest an achievable rate $H(X Y)$ for communicating X is universally attainable. (b) Source networks of flexible decoding with predictor uncertainty. The same universal codes can be used to communicate X in any of these networks at a rate $H(X Y_i)$	85

3.7	Theoretical performances of intra coding, CLP and DSC in a flexible decoding scenario: Multiview video coding as in Figure 3.1. (a) Previously reconstructed frames of neighboring views are used as predictor candidates following the depicted order; (b) Entropy of the quantized DCT coefficients (as an estimate of the encoding rate) vs. number of predictor candidates. The results are the average of 30 frames using Akko&Kayo view 28-th. (c) Entropy of each frame in the case of three predictor candidates. Note that we assume the sources (i.e., quantized DCT coefficients of images, X , or that of residues, Z_i) consist of independent elements and estimate the coding performances by $H(X)$ and $H(Z_i)$, whereas in practice there could be some correlation exist between the elements and that could be exploited to reduce coding rates.	86
3.8	Theoretical performances of intra coding, CLP and DSC in a flexible decoding scenario: Robust video transmission as in Figure 3.3. (a) Past reconstructed frames are used as predictor candidates following the depicted order; (b) Entropy of the quantized DCT coefficients (as an estimate of the encoding rate) vs. number of predictor candidates. The results are the average of 30 frames using Coastguard. (c) Entropy of each frame in the case of three predictor candidates. Note that we assume the sources (i.e., quantized DCT coefficients of images, X , or that of residues, Z_i) consist of independent elements and estimate the coding performances by $H(X)$ and $H(Z_i)$, whereas in practice there could be some correlation exist between the elements and that could be exploited to reduce coding rates.	87
3.9	Proposed encoding algorithm to encode an macroblock M of the current frame.	88
3.10	Encoding macroblock M using DSC.	89
3.11	Estimate the conditional probability $p(b(l) Y_i, b(l + 1), b(l + 2), \dots)$	94
3.12	Different coding structures in multiview video application. Shaded video frames are those that needed to be decoded to enable the depicted decoding path. (a) Simulcast; (b) New coding structure enabled by the proposed tools, where “S” denotes those video frames that can be decoded using any one of the predictor candidates. Note that in simulcast, decoder needs to decode some extra video frames that users did not request, e.g., the first three frames in the v -th view in the depicted example. In client-server applications, bitstream corresponding to these extra frames is also needed to be sent to the client.	96
3.13	Simulation results of multiview video coding: Akko&Kayo. The results are the average of the first 30 frames (at the switching points) from 3 different views (views 27th-29th) arbitrarily chosen from all the available views. . .	98

3.14	Simulation results of multiview video coding: Ballroom. The results are the average of the first 30 frames (at the switching points) from 3 different views (views 3rd-5th) arbitrarily chosen from all the available views. . . .	99
3.15	Drifting experiment using Akko&Kayo view 28th: (a) CLP; (b) DSC. GOP size is 30 frames. Note that with DSC, the PSNR are almost the same in the switching and non-switching cases.	100
3.16	Scaling experiment: in this experiment, the temporally and spatially adjacent reconstructed frames are used as predictor candidates following the depicted order.	100
3.17	Scaling experiment using sequence Akko&Kayo view 29th. PSNR of different schemes are comparable - Intra: 35.07dB, CLP: 34.78dB, DSC:34.79dB. 101	101
3.18	Simulation results of forward/backward video playback: Coastguard. Results are reported for the average of the first 30 frames. Note that H.263 inter-frame coding cannot support flexible decoding - the results are shown here for reference only.	101
3.19	Simulation results of forward/backward video playback: Stefan. Results are reported for the average of the first 30 frames. Note that H.263 inter-frame coding cannot support flexible decoding - the results are shown here for reference only.	102
3.20	Drifting experiment using Stefan sequence: (a) CLP; (b) DSC. The figure shows the PSNR of the reconstructed frames in the first GOP. Note that with DSC, the PSNR are almost the same in the switching and non-switching cases.	102
4.1	(a) An example of Slepian-Wolf coding exploiting binary correlation. Boxes “ Q ” and “ Q^{-1} ” denote quantization and inverse quantization respectively. Boxes “ B ” and “ B^{-1} ” denote binarization and the inverse respectively. (b) An example of Slepian-Wolf coding exploiting continuous correlation. 107	107
4.2	Apply sampling to distributed video coding: when encoding the current frame, we randomly sample n macroblocks of the current frame to undergo motion estimation, with n being much smaller than the total number of macroblocks. By doing so, we would obtain n pairs of samples (X, Y) , where X is the current macroblock and Y is the corresponding motion-compensated predictor. From these n sample pairs the encoding rate can be estimated. Note that here the sampling cost associated with each data sample is not primarily due to data exchange but the computation in motion search.	109

4.3	Encoding rate function $H(\cdot)$. The same Δp_l would have a larger impact to the estimated rate if the true p_l is small. The optimal sample allocation takes into account this characteristic.	122
4.4	Reduction in rate penalty (bits) using the optimal sample allocation, with $L = 4$, $K_l = 16384$, $n_T = 128$ or 1024 (i.e., 0.20% or 1.56% of total respectively), $N_E = 100$, and $\bar{p} = 0.25$	126
4.5	(a) Crossover probability estimation for raw bit-planes. A_i are the events that lead to occurrence of crossover between X and Y at significance level l . (b) Refinement bit-plane crossover probability estimation: probability of crossover and X_i is already significant. (c) Sign bit-plane crossover probability estimation: probability of sign crossover and X_i becomes significant.	129
4.6	Comparing estimation accuracy between the direct approaches and the model-based approach.	132
4.7	$Pr(U(l, x))$ when (a) $\lfloor \frac{ x }{2^l} \rfloor$ is odd ; (b) $\lfloor \frac{ x }{2^l} \rfloor$ is even.	139
4.8	The DSC-based hyperspectral image compression with direct correlation estimation.	141
4.9	DSC-based hyperspectral image compression with model-based estimation.	142
4.10	Sample allocation experiments - (a) <i>Lunar</i> (reflectance data), (b) <i>Moffet</i> (radiance data), using 0.25% total sample. An adaptive sample allocation scheme using the proposed optimal sample allocation strategy in Section 4.3 with a priori information from previous encoded band is compared with the even sample allocation. Here “Exact” denote the cases when the exact correlation information is used to determine the coding rate, i.e., no rate penalty.	144
4.11	Coding efficiency comparison: (a) <i>Cuprite</i> (radiance data); (b) <i>Lunar</i> (reflectance data). The model-based system is compared with the original system, which uses all samples in direct estimation and exact crossover probabilities to determine the encoding rate, i.e. no rate penalty. Coding performances of several other wavelet systems are also presented for reference.	147
4.12	(a) Encoding of bit-plane $b_X(l)$ with both the previously decoded bit-planes $b_X(l + 1), \dots, b_X(l + m)$ and that of the correlated source $b_Y(l), b_Y(l + 1), \dots, b_Y(l + m)$ as SI. (b) Joint p.d.f. between the input and all the SI.	151

4.13	Entropy/conditional entropy of the video bit-planes data used in the experiment in Section 4.2.4, i.e., X and Y are the quantized DCT coefficients in a current frame and the corresponding quantized coefficients in the motion-compensated predictors in the reference frame, respectively, using the 2nd AC coefficients of <i>Mobile</i> (720×576 , 30 fps), at QP= 12: (a) Without SI, i.e., intra coding (“No SI”); (b) Using only the corresponding bit-plane as SI, i.e., $m = 0$ (“SI: Corr. bit-plane”); (c), (d), (e): Using corresponding and one, two or three previously decoded bit-planes as SI, i.e., $m = 1, 2$ or 3 respectively. The results suggest further improvements with $m > 1$ could be negligible.	152
4.14	Extension of model based estimation: The events leading to the occurrence of $\langle b_X(l+m) \dots b_X(l+1)b_X(l) \rangle = i$ and $\langle b_Y(l+m) \dots b_Y(l+1)b_Y(l) \rangle = j$ correspond to the region $A_{i,j}$ in the sample space of X and Y	152
4.15	The events leading to the occurrence of $\langle b_X(l+m) \dots b_X(l+1)b_X(l) \rangle = i$ correspond to regions A_i , subset of X	154

Abstract

Many video compression schemes (e.g., the recent H.264/AVC standard) and volumetric image coding algorithms are based on a closed-loop prediction (CLP) framework. While CLP based schemes can achieve state-of-the-art coding efficiency, they are inadequate in addressing some important emerging applications such as wireless video, multiview video, etc, which have new requirements including low complexity encoding, robustness to transmission error, flexible decoding, among others. In this research we investigate new video and image compression algorithms based on distributed source coding (DSC), and we demonstrate the proposed algorithms can overcome some of the deficiencies in CLP based systems while achieving competitive coding performance.

The first part of this thesis discusses our work to explore DSC principles for designing hyperspectral imagery compression algorithms, with an eye toward an efficient and parallel encoder implementation with modest memory requirement. Using DSC tools allows encoding to proceed in “open loop”, and this facilitates parallel compression of spectral bands in multi-processors configurations. We demonstrate that our proposed DSC techniques can be adaptively combined with set partitioning of wavelet coefficients to exploit spatial and spectral correlation. Our latest results show the proposed algorithm

can achieve a comparable coding efficiency to a simple 3D wavelet codec developed at NASA-JPL.

The second part of this thesis investigates DSC based coding algorithms to address the flexible decoding problem in video applications. In these, the encoder needs to compress a current frame under uncertainty on the predictor available at decoder. Flexible decoding is relevant in a number of applications including multiview video, frame-by-frame forward and backward video playback, robust video transmission, etc. The proposed algorithm incorporates novel macroblock mode switching and significance coding within the DSC framework. This, combined with a judicious exploitation of correlation statistics, allows us to outperform other competing solutions.

The third part of this thesis proposes solution to address the correlation estimation problem in DSC, which is an important subject for practical DSC applications. We formulate the rate-constrained correlation estimation problem in a DSC framework, and propose information exchange strategies that minimize the rate penalty due to inaccurate estimation. We also propose a novel model-based method for correlation estimation in the context of DSC. We demonstrate that the model-based estimation can achieve accurate estimation with minimal computational and data exchange requirements.

Chapter 1

Introduction

1.1 Motivation

In the past decade, both academia and industry have devoted substantial research and standardization efforts to the development of multimedia compression algorithms. Some well-known examples are the MPEG video coding standards developed by ISO/IEC and the H.26X video coding standards developed by the ITU. Digital video and image compression have become central technologies in a variety of applications, including consumer electronics (e.g., DVD, digital still camera), the Internet (e.g., JPEG-compressed pictures, streaming video), distance learning, surveillance and security, and remote sensing.

Conventionally, video compression standards (e.g., the recent H.264/AVC standard) and many volumetric image coding algorithms are based on a *closed-loop prediction* (CLP) framework [80, 82]. In CLP systems, encoder would compute the difference between the input source and a predictor available at both the encoder and decoder, and communicate this difference, or *prediction residue*, to the decoder (Figure 1.1). Compression schemes based on CLP have demonstrated state-of-the-art coding performance.

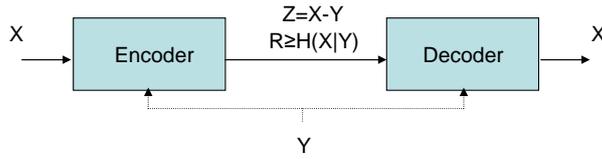


Figure 1.1: Compression using CLP. Encoder computes the difference between the input source X and the predictor Y and communicates the difference (prediction residue) to the decoder.

CLP based algorithms, however, could be inadequate to address the requirements of several important emerging applications. For example, CLP schemes are vulnerable to transmission error, and it is non-trivial to communicate CLP compressed data over lossy channels as occurred in wireless video applications [66, 75, 87]. Moreover, CLP systems may lack the decoding flexibility required by some emerging applications such as multiview video [35, 70]. Furthermore, CLP coding algorithms are inherently sequential, and it is difficult to achieve parallel encoding of slices in the cases of volumetric image compression [51].

It is the purpose of this research to investigate novel video and image compression algorithms to address the aforementioned issues in conventional compression framework based on CLP. Specifically, we propose hyperspectral image and video compression algorithms based on distributed source coding (DSC). We demonstrate the proposed DSC based algorithms can overcome some of the deficiencies in conventional schemes, and achieve competitive compression efficiency. One of the central problems in DSC is to estimate, during encoding, the correlation information between the input and the predictor (*side information*) available only at the decoder [30, 85]. Therefore, we also propose different correlation estimation strategies based on sampling techniques.

Hyperspectral imagery is usually highly correlated, in some cases within each spectral band, but in particular across neighboring frequency bands [38, 51]. In Chapter 2 we propose to use DSC to exploit this correlation with an eye to a parallel encoding implementation with modest memory requirement [8, 14, 67]. We apply DSC principles to hyperspectral images by encoding individual spectral bands under the assumption that these bands are correlated. As will be discussed, using DSC tools allows the encoder to operate in “open loop” without requiring having access to decoded versions of (spectrally) neighboring bands, and this facilitates parallel encoding of spectral bands in multi-processor architectures. We first compute the parameters of a linear predictor to estimate the current spectral band from a neighboring one, and estimate the correlation between these two bands (after prediction). Then a wavelet transform is applied and a bit-plane representation is used for the resulting wavelet coefficients. We observe that in typical hyperspectral images, bit-planes of same frequency and significance located in neighboring spectral bands are correlated. We exploit this correlation by using low-density parity-check (LDPC) based Slepian-Wolf codes [43, 46]. The code rates are chosen based on the estimated correlation. We demonstrate that set partitioning of wavelet coefficients, such as that introduced in the popular SPIHT algorithm, can be combined with our proposed DSC techniques so that coefficient significance information is sent independently for all spectral bands, while sign and refinement bits can be coded using adaptive combinations of DSC and zerotree coding. Our latest results suggest that coding efficiency comparable to that of a simple 3D wavelet codec developed at NASA-JPL can be achieved by our proposed algorithm.

In Chapter 3 we investigate compression techniques to support *flexible video decoding* [10–12, 16]. In these, encoders generate a *single* compressed bit-stream that can be decoded in *several* different ways, so that users or decoders can choose among several available decoding paths. Flexible decoding has several advantages, including improved accessibility of the compressed data for emerging applications (e.g., multiview video) and enhanced robustness for video communication. Flexible decoding, however, makes it difficult for compression algorithms to exploit temporal redundancy: when the decoder can choose among different decoding paths, the encoder no longer knows deterministically which previously reconstructed frames will be available for decoding the current frame. Therefore, to support flexible decoding, encoders need to operate under uncertainty on the decoder predictor status. We propose to address flexible decoding based on DSC. The main advantage of a DSC approach to flexible decoding is that the information communicated from the encoder to the decoder (namely, the parity bits) is independent of a specific predictor. By “decoupling” the compressed information from the predictor, we will demonstrate that, theoretically and experimentally, DSC can lead to a solution that compares favorably to, in terms of coding efficiency, one based on conventional CLP approach, where multiple prediction residues are sent, one for each possible predictor available at the decoder. The main novelties of the proposed algorithm are that it incorporates different macroblock modes and significance coding within the DSC framework. This, combined with a judicious exploitation of correlation statistics, allows us to achieve competitive coding performance. Experimental results using multiview video coding and forward/backward video playback suggest the proposed DSC-based solution can outperform flexible decoding techniques based on CLP coding.

Central to practical DSC applications is the correlation information between the input and side information, which usually has to be estimated at the encoder in order to determine the encoding rate [30,85]. Coding efficiency depends strongly on the accuracy of this correlation estimation. While error in estimation is inevitable, the impact of estimation error on compression efficiency has not been sufficiently studied for the DSC problem. In Chapter 4, we study correlation estimation subject to rate and complexity constraints, and its impact on coding efficiency in a DSC framework for practical distributed image and video applications [9,15]. We focus, in particular, on applications where binary correlation models are exploited for Slepian-Wolf coding and sampling techniques are used to estimate the correlation, while extension to other correlation models will also be briefly discussed. In the first part of Chapter 4 we investigate the compression of binary data. We first propose a model to characterize the relationship between the number of samples used in estimation and the coding rate penalty, in the case of encoding of a single binary source. The model is then extended to scenarios where multiple binary sources are compressed, and based on the model we propose an algorithm to determine the number of samples allocated to different sources so that the overall rate penalty can be minimized, subject to a constraint on the total number of samples. The second part of Chapter 4 studies compression of continuous-valued data. We propose a model-based estimation for the particular but important situations where binary bit-planes are extracted from a continuous-valued input source, and each bit-plane is compressed using DSC. The proposed model-based method first estimates the source and correlation noise models using continuous-valued samples, and then uses the models to derive the bit-plane statistics analytically. We also extend the model-based estimation to the cases when bit-planes

are extracted based on the significance of the data, similar to those commonly used in wavelet-based applications. Experimental results demonstrate the effectiveness of the proposed algorithms.

The rest of this chapter gives a brief review of related topics and summarizes the contributions of this thesis.

1.2 Distributed Source Coding

DSC addresses the problem of compression of correlated sources that are not co-located. The information-theoretic foundations of DSC were laid out in the 1970s in the pioneering works of Slepian and Wolf [63], and Wyner and Ziv [84]. Driven by its potential to some emerging applications (sensors network, wireless video, etc), DSC has attracted much attention recently.

In this section we briefly review some information-theoretic results of DSC most relevant to our applications. We also discuss the basic ideas of constructive DSC algorithms, and properties of DSC essential to practical applications.

1.2.1 Slepian-Wolf Theorem

Here we illustrate a particular case of the Slepian-Wolf theorem which is most relevant to practical DSC applications, often referred as (lossless) source coding with decoder side-information. Consider the set-up in Figure 1.1, where we try to losslessly compress an i.i.d. random source $X^n = \{X_1, X_2, \dots, X_n\}$ with another correlated i.i.d. random source $Y^n = \{Y_1, Y_2, \dots, Y_n\}$ available at *both* the encoder and decoder. That is, $\{X_i, Y_i\}_{i=1}^n$ i.i.d. $\sim p(x, y)$, and X and Y are discrete random variables. In this case, we can use CLP (e.g.,

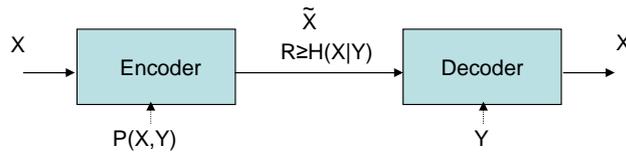


Figure 1.2: Slepian-Wolf theorem.

DPCM) to compress X^n , i.e., we use Y^n to predict X^n and then encode the prediction residue. The theoretical lower bound of the lossless encoding rate is $H(X|Y)$.

In distributed source coding, instead, we consider the situation where Y^n is available *only* at the decoder (Figure 1.2). This situation appears to pose a more difficult coding problem than that of Figure 1.1. However, Slepian and Wolf [63] have shown that theoretically we can achieve the same lower bound as in the previous case, i.e., $H(X|Y)$, even the actual realization of Y^n that will be available at the decoder is not known during encoding. The Slepian-Wolf theorem thus suggests that efficient encoding is indeed possible even when the encoder does not have precise knowledge of the side-information Y^n available at the decoder.

1.2.2 Constructive Coding Algorithm

The Slepian-Wolf theorem states that, theoretically, the best achievable rates are the same with or without Y^n at the encoder. However, practically, how can we compress X^n when Y^n is available only at the decoder, and achieve the same performance as when Y^n is available at both the encoder and decoder? Most constructive algorithms for Slepian-Wolf coding are based on channel coding [1, 27, 43, 56, 57, 83]. We will first illustrate the idea

with an example using *cosets*, and then describe the mechanism to perform Slepian-Wolf coding with error correcting codes.

1.2.2.1 Coset Example

Consider an example when we try to represent a discrete random variable X distributed uniformly in $[0,255]$. In this case intra coding would require 8 bits to represent X . Suppose there is another correlated random variable Y available, and the correlation between X and Y is such that $Y - X$ is distributed uniformly in $[-4,4)$. We can exploit this correlation to reduce the bit-rate to represent X . If Y is available at both encoder and decoder, we can use CLP to communicate X , i.e., we encode the residue $Y - X$. Since $Y - X$ is distributed uniformly in $[-4,4)$, only 3 bits are required to represent X .

In situations when Y is not available at the encoder we can convey X using the following algorithm (Figure 1.3) [85]. We partition the space of all reconstruction levels into different groups or *cosets*. In this particular example, we partition the space into 8 cosets: $\{A, B, C, \dots, G, H\}$, and each coset includes several reconstruction levels separated by the same distance. As will be discussed, the number of cosets is determined by the correlation between X and Y . To communicate X , we transmit the label of the coset which X belongs to (C in Figure 1.3), which would require 3 bits (since there are 8 cosets). At the decoder, we receive the coset label and observe the side-information Y . The coset label suggests X can be any one of the members in the coset. To disambiguate this information, we select the one coset member that is closest to Y and declare it as the reconstructed value for X . By doing so, one can check that we can always recover X

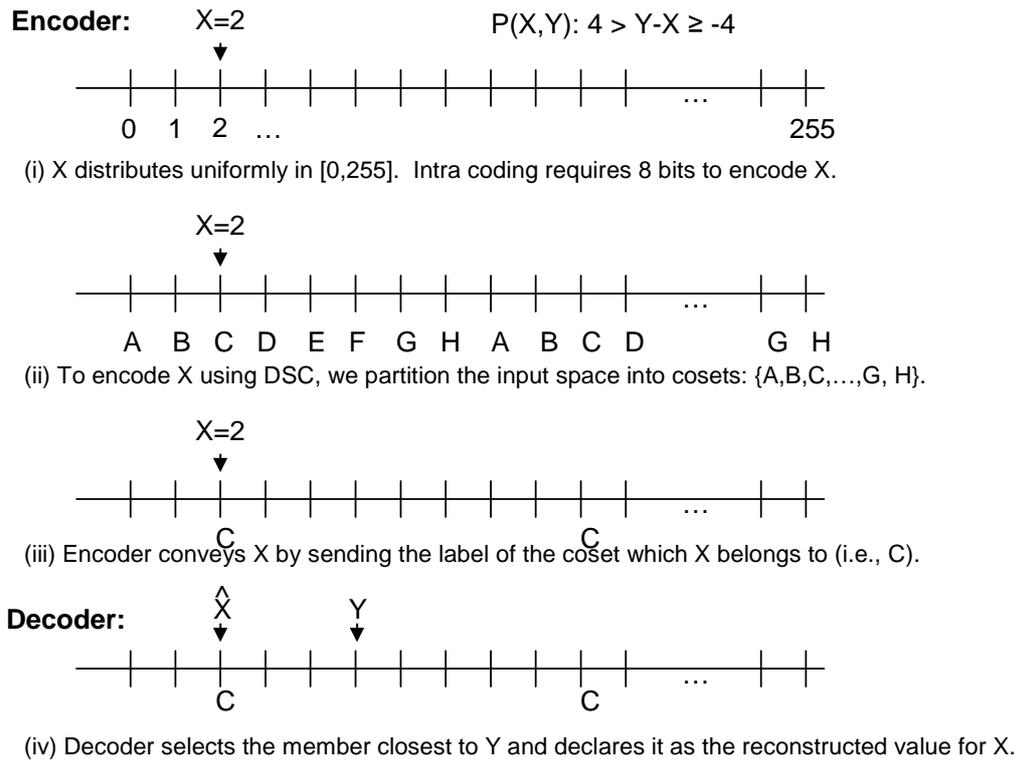


Figure 1.3: An example using coset to illustrate the main ideas of constructive DSC algorithms.

exactly as long as X and Y obey the correlation structure. Therefore, we can communicate X using 3 bits, same as CLP, with Y available only at the decoder.

1.2.2.2 Main Ideas and General Steps

As illustrated by the coset example, the main ideas of DSC are that the encoder sends ambiguous information to achieve rate savings, and the decoder would disambiguate the information using the side information. In the coset example, the ambiguous information is the coset label representing a group of reconstruction levels. As long as the members are sufficiently apart within the coset, we would be able to recover the original value

by selecting the member which is identified by the coset label and is closest to the side information.

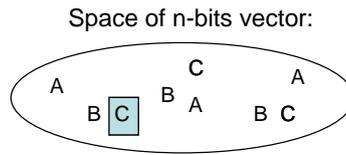
The general coding steps of DSC can be summarized as follows:

1. Partitioning the input space into cosets;
2. Members in the coset are separated by some minimum distance;
3. Encoder sends the coset label to decoder;
4. Decoder selects the coset member closest to the side-information, and declares it as the reconstructed value.

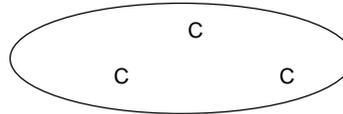
1.2.2.3 Practical Algorithms Using Error Correcting Codes

Practical coding algorithms employing error correcting codes follow the same basic ideas and general steps discussed above [28, 58, 85]. For example, to encode an n -bit binary vector \mathbf{X} with a linear (n, k) binary error correcting code defined by the *parity matrix* \mathbf{H} , we compute $\mathbf{S} = \mathbf{X}\mathbf{H}^T$, where the $(n - k)$ -bit \mathbf{S} is the *syndrome*. The syndrome would serve the same function as the coset label in the coset example. From coding theory, of all the 2^n possible \mathbf{X} , 2^k of them will have the same syndrome, thus we can partition the input space of \mathbf{X} into $2^{(n-k)}$ cosets according to the syndrome (Figure 1.4). Moreover, the minimum *Hamming distance* between members is the same for each coset.

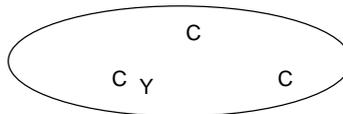
To communicate \mathbf{X} , we would send the syndrome of the coset which \mathbf{X} belongs to. At the decoder, we receive the syndrome and observe the side-information \mathbf{Y} . To disambiguate the information, we select the one closest to \mathbf{Y} in the coset (Figure 1.4). Here



(i) 2^{n-k} cosets, each identified by coset label (syndrome).



(ii) Encoder sends the syndrome of the coset which X belongs to.



(iii) Decoder selects the member closest to Y .

Figure 1.4: Practical coding algorithms using error correcting codes.

we achieve compression by sending the $(n - k)$ -bit \mathbf{S} instead of the original n -bit source \mathbf{X} , and the encoding rate is $\frac{n-k}{n}$, which should be no less than the Slepian-Wolf limit.

Note that in order to achieve performance close to the Slepian-Wolf limit, error correction code based algorithms would need to use very large block lengths, e.g., n is of the order of 10^5 [85]. These may not be suitable for practical image and video applications, where correlation between symbols is non-stationary and may vary significantly between small blocks. Recent work has proposed to address distributed compression based on arithmetic coding, which may achieve good coding performance with much smaller block sizes, e.g., n is of the order of 10^3 [29].

1.2.3 Properties of Distributed Source Coding

In this section we will highlight some of the properties of DSC that are most useful for some emerging applications.

1.2.3.1 Encoding Requiring Correlation Information Only

The most notable property of DSC is that encoding requires, in addition to the input source itself, only the correlation information between the source and its side-information. In particular, the exact realization of the side-information is not needed during encoding, as illustrated by the coset example. This property makes DSC useful for application scenarios where encoders do not have access to the predictors. An example of this scenario could be compression of sensor data [55, 85]. Consider a dense sensor field where individual sensors acquire and transmit information to a central node for processing. Since the information could be highly correlated, the sensors may opt to coordinate with their neighbors and remove the redundancy in the collected information before sending it to the central node, so that lower transmission rates can be achieved. However, the conventional CLP approach would incur substantial local communication between the neighboring sensors, since the exact value of a predictor (Y) available at one node needs to be communicated to a neighboring node in order to encode X . Using DSC, communication between sensors could be avoided if the correlation information is available at the sensors, leading to significant reduction in energy consumption. Practically, correlation information may need to be estimated, and some aspects of this correlation estimation are studied in this research [9, 15] (Chapter 4).

Another example of a scenario where predictors could be inaccessible during encoding is low complexity video encoding. Conventional video coding standards (e.g. MPEG, H.26x) follow the CLP approach, and to encode a block in the current frame, encoder would use *motion estimation* to find the optimal predictor block in the previous reconstructed frames. The prediction residue is then computed, compressed and communicated to the decoder. Motion estimation is a well-established technique to exploit the temporal correlation between neighboring frames to achieve high compression ratio, but it incurs substantial computational complexity, leading to asymmetric coding frameworks where encoders are more complex than decoders. This poses challenges to some emerging applications such as video sensor networks or mobile video, where encoders need to operate under power constraints and low complexity encoding would be more suitable. In contrast to CLP, DSC requires only correlation information during encoding, and if encoder could acquire such information with minimal computation, low complexity encoding can be achieved [4, 28, 59].

It should be noted that the coding efficiency of DSC depends strongly on the accuracy of the correlation information, and decoding error may occur if the correlation is not known accurately. We illustrate this with the coset example discussed in Section 1.2.2.1. Recall in this example the “correct” correlation between the source and the side-information is $4 > Y - X \geq -4$. Suppose the encoder has obtained inaccurate correlation information between the input data, say, $2 > Y - X \geq -2$, i.e., encoder thought X and Y were *more* correlated. Based on the inaccurate correlation information, the encoder determines the Slepian-Wolf limit, $H(X|Y)$, to be 2 bits, and uses four cosets to communicate X (Figure 1.5). Following the same coding steps, decoder would receive the

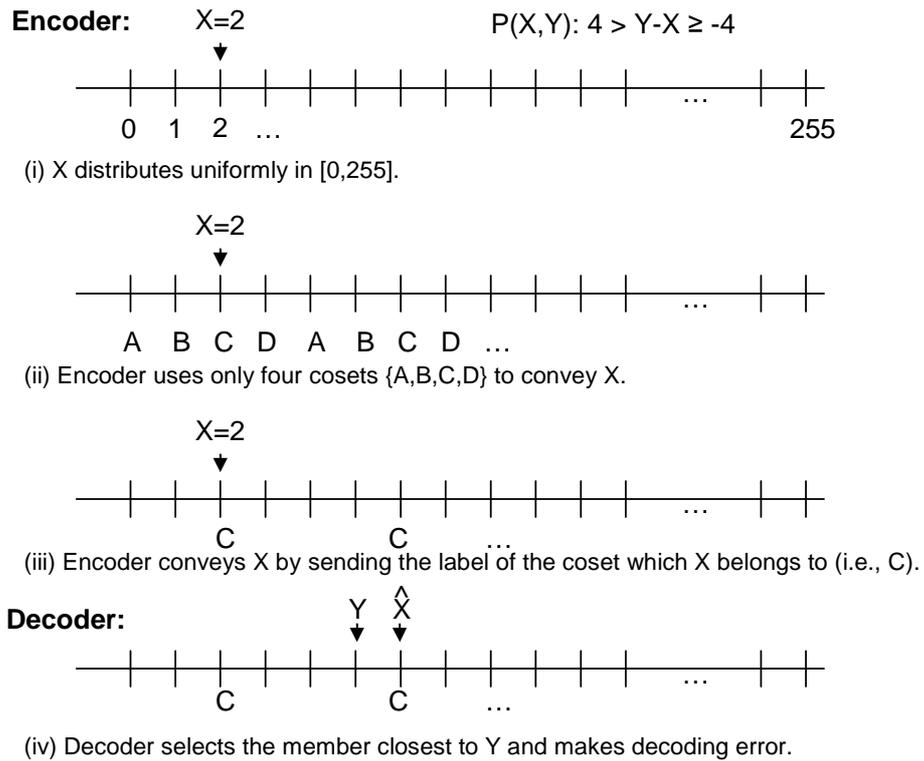


Figure 1.5: Decoding error due to inaccurate correlation information.

coset label, and to disambiguate this coset information decoder would select the member closest to Y and declare it as the reconstructed value of X . However, in this case, the reconstructed value could be different from the original one, as illustrated in Figure 1.5, and decoding error could occur. Therefore, in DSC, if the encoder over-estimates the correlation a decoding error can occur. On the other hand, if the encoder under-estimates the correlation, we can still reconstruct X correctly at decoder, but we suffer a penalty in terms of coding performance.

1.2.3.2 Systems Robust to Uncertainty on Predictor

Another property of DSC is that DSC systems tend to be more robust to error in the predictor, or to uncertainty on the predictor. As illustrated in the coset example, in DSC, the compressed information (i.e., coset label or syndrome) is computed based on the correlation information, instead of directly from the predictor as in CLP. Therefore, DSC is an approach to exploit the correlation without using the predictor directly in the encoding. By “de-coupling” the predictor from the encoding process and from the compressed information, DSC systems can be robust to errors on or uncertainties about the predictor. For example, it is possible to reconstruct the source exactly even when the side-information is corrupted by errors. This can be illustrated by the coset example (Figure 1.6). Recall that at the decoder we would select the member closest to the side-information Y as the reconstructed value for X . Suppose Y is corrupted by noise N . If the noise power is less than some limit, we can still reconstruct X exactly as illustrated in Figure 1.6. This property can be exploited in a DSC based video communication system, where X would be the current frame and Y would be the reference frame [60, 62, 78]. In this case even there is transmission error in the reference frame, it is still possible to reconstruct the current frame exactly and prevent any error propagation. This is in contrast to CLP systems such as MPEG/H.26x video coding schemes, where error would propagate until a video frame (or block) is intra-coded. Note that DSC exploits inter-symbol correlation (between X and Y) when encoding X , and thus could achieve better coding efficiency than intra-coding of X .

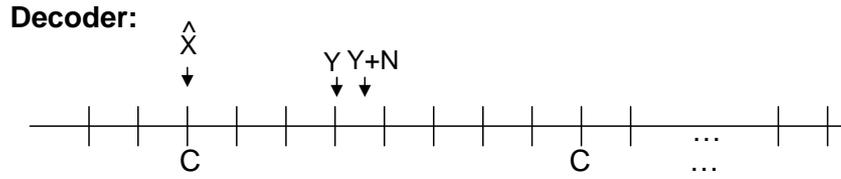


Figure 1.6: Robustness property of DSC.

1.3 Hyperspectral Imagery

In Chapter 2 we propose DSC-based compression algorithms for hyperspectral imagery. We first briefly review the basics of hyperspectral image compression in this section.

Hyperspectral images consist of hundreds of spatial images each being acquired at a particular frequency (*spectral band*) (Figure 1.7). Therefore, in hyperspectral data sets, the pixel values along the spectral direction depict the spectra of the captured objects, and this spectral information can be used in classification applications, e.g., identification of surface materials. The raw data size of hyperspectral images is non-trivial. For example, the images captured by AVIRIS (Airborne Visible/Infrared Imaging Spectrometer, operated by NASA) include 224 spectral bands, and a single hyperspectral image could contain up to 140 Mbytes of raw data [40]. Therefore, efficient compression is necessary for practical hyperspectral imagery applications. In addition, hyperspectral images are usually captured by satellites or spacecrafts that use embedded processors with limited resources, so encoding complexity is an important issue in hyperspectral image compression. Furthermore, due to the large amount of raw data, high speed encoding is desirable, and one approach to speed up encoding could be to perform parallel compression on several processors. This will be one of the topics investigated in this research.

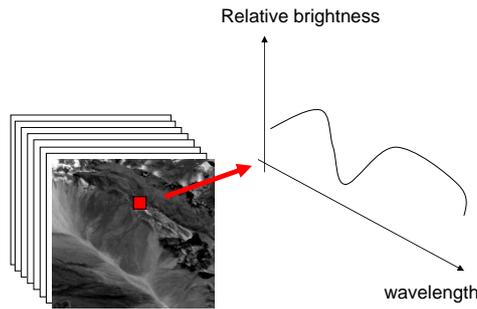


Figure 1.7: Hyperspectral image.

1.3.1 Review of Hyperspectral Imagery Compression Approaches

In a hyperspectral dataset many spectral bands are highly correlated. This is shown in Figure 1.8, where image mean-square residuals after simple alignment are shown for two different views on a site. Neighboring bands tend to be correlated and the degree of correlation varies relatively slowly over a broad range of spectral regions. Thus, exploiting inter-band correlation using, for example, inter-band prediction followed by 2D compression [51] or 3D wavelet decompositions [69], has proven to be a popular approach to compress hyperspectral images¹.

1.3.1.1 Inter-band Prediction Approaches

In inter-band prediction approaches, a band is predicted using previously encoded bands and the resulting prediction residuals are encoded using standard image coding techniques (transformation followed by quantization and entropy coding). Since, typically, the prediction residue has a much lower energy than the original band, encoding the residue

¹As will be illustrated later, it is easy to modify an algorithm that exploits cross-band correlation so that it operates independently in each frame when correlation is low, as is the case in some spectral regions in Fig. 1.8.

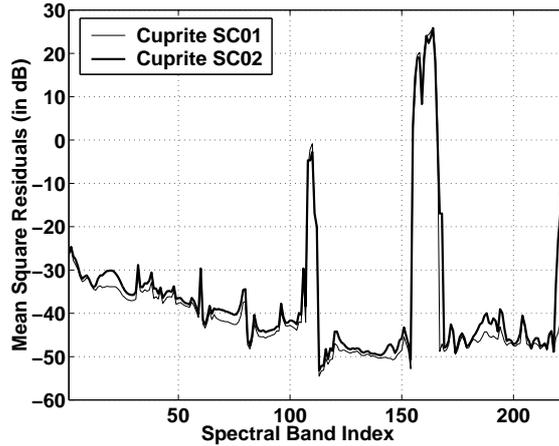


Figure 1.8: Mean square residuals after simple image alignment and subtraction.

usually requires fewer bits than encoding the original band. Inter-band prediction approaches are analogous to the standard MPEG/H.26x compression algorithms for video, but motion estimation/compensation are not necessary since co-located pixels in different spectral bands represent the same ground object (at different frequencies).

Inter-band prediction approaches can achieve high compression ratio with moderate memory requirements. However, there are several drawbacks. First of all, inter-band prediction methods need to generate exact copies of the decoded bands at the encoder, so encoders need to perform decoding as well, and decoding complexity could be significant, e.g., comparable to encoding complexity. [51] has proposed using only full bit-planes to form the predictors at the encoder and decoder. This could avoid bit-plane decoding at the encoder. However, since this approach does not utilize fractional bit-plane information in reconstruction, the predictors in general have worse qualities compared to that of conventional inter-band prediction methods, leading to degradation in coding performance. Second, inter-band predictive methods are inherently serial, since each band is encoded based on a predictor obtained from previously *decoded* bands. Therefore, it is

difficult to scale up the processing speed of an inter-band predictive encoder to handle the high data rate generated by hyperspectral imaging instruments. Furthermore, it is difficult to achieve efficient rate scalability. This is because bit-rate scaling by on-the-fly truncation of the bit-stream during transmission may lead to a different reconstruction at the decoder, resulting in drifting.

1.3.1.2 3D Wavelet Approaches

3D wavelet methods, including 3D-SPECK, 3D-SPIHT [69], or 3D-ICER [38, 39] developed by NASA Jet Propulsion Laboratory, provide alternatives to predictive techniques. 3D wavelet methods can also exploit inter-band correlation by performing filtering across spectral bands, with the expectation that most of the signal energy will be concentrated in low pass subbands (corresponding to low spatial and “cross band” frequencies).

As an example to illustrate 3D wavelet approaches, in 3D-ICER [38, 39], a modified 3D Mallat decomposition is first applied to the image cube. Then mean values are subtracted from the spatial planes of the spatially low-pass sub-bands to account for the “systematic difference” in different image bands [39]. Then bit-plane coding is applied to the transform coefficients, with each coefficient bit adaptively entropy-coded based on its estimated probability-of-zero statistics. The probability is estimated by classifying the current bit to be encoded into one of several contexts according to the significance status of the current coefficient and its spectral neighbors. A different probability model is associated to each context.

While 3D wavelet methods can achieve good compression efficiency with excellent scalability, a main disadvantage is that they lead to complex memory management issues. A naive implementation would consist of loading several spectral bands in memory so as to perform cross-band filtering, leading to expensive memory requirements. More sophisticated approaches are possible, e.g., loading simultaneously only subbands corresponding to a given spatial frequency in various spectral bands, but these approaches have the drawback of requiring numerous iterations of memory access.

1.4 Contributions of This Research

In this research we propose novel image and video coding algorithms based on DSC in order to address some of the deficiencies in conventional compression framework. We also address the correlation estimation problem in practical DSC applications. The main contributions of this research include the following:

1. *Wavelet-based Slepian-Wolf coding for hyperspectral imagery* [8, 14, 67].
 - We exploit the DSC principle to design hyperspectral imagery compression with an eye to an efficient parallel encoder implementation. We combine set partitioning of wavelet coefficients with our proposed DSC techniques to achieve competitive coding performance. Our proposed system can achieve a comparable coding efficiency to a simple 3D wavelet codec developed at NASA-JPL.
 - We propose adaptive coding strategies that optimally combine DSC with intra coding for wavelet-based DSC applications. Experimental results demonstrate

that these adaptive strategies can lead to up to 4dB improvement compared with a non-adaptive system.

2. *Flexible video decoding* [10–13, 16].

- We investigate compression techniques to support flexible video decoding based on DSC. The proposed algorithm incorporates different macroblock modes and significance coding within the DSC framework, and combined with a judicious exploitation of correlation statistics the proposed algorithm can outperform flexible decoding techniques based on conventional CLP coding.
- We study the information-theoretical achievable rate bound for the flexible decoding problem under predictor uncertainty.

3. *Sampling-based correlation estimation for DSC* [9, 15, 17].

- Within the DSC framework, we propose models to characterize the relationship between the number of samples used in estimation and the coding rate penalty.
- In compression of multiple binary sources and under constraints on the total number of samples, we propose an algorithm to determine the number of samples allocated to different sources so that the minimum overall rate penalty can be achieved.
- We propose model-based estimation for distributed coding of continuous valued input sources. Experimental results, including some based on real image data, demonstrate the effectiveness of the proposed algorithms.

The rest of the thesis is organized as follows. We discuss hyperspectral image compression in Chapter 2. Chapter 3 discusses how to address flexible decoding based on DSC. Chapter 4 studies the correlation estimation problem in DSC. Finally, Chapter 5 concludes the research and discusses future work.

Chapter 2

Efficient Wavelet-based Predictive Slepian-Wolf Coding for Hyperspectral Imagery

2.1 Introduction

2.1.1 Motivation

In this chapter we propose novel compression algorithms for hyperspectral imagery that facilitate parallel and low complexity encoding, while achieving competitive compression performance. Our proposed techniques use wavelet-based encoding to enable lossy to lossless, scalable encoding of the spectral bands. This is combined with distributed source coding techniques [63], which are used to exploit the inter-band correlation. As discussed in Chapter 1.2, Slepian and Wolf [63] proved that two correlated sources can be optimally encoded even if the encoder only has access to the two sources separately. This counter-intuitive result permits in principle significant complexity and communication overhead reductions in parallel encoding configurations, while preserving the ability to optimally compress the data (approaching the same performance as conventional schemes based on predictive framework) by exploiting the redundancy in the correlated spatial images

at adjacent spectral bands. These advantages are particularly important for hyperspectral imagery compression, where low complexity, high speed encoding is most needed. Other applications of Slepian-Wolf coding include data aggregation in sensor networks (e.g., [55, 85]) and video coding (e.g., [28, 59]). In the video coding applications, the correlated sources are successive video frames. In this work, correlated sources will be successive bands of hyperspectral imagery.

2.1.2 Our Contributions and Related Works

Our proposed scheme, set-partitioning in hierarchical trees with Slepian-Wolf coding (SW-SPIHT), is an extension of the well-known SPIHT algorithm [61]. SW-SPIHT first uses an iterative set-partitioning algorithm to extract bit-planes. Bit-planes at the same bit position in neighboring bands are shown to be correlated. Once the first spectral band, which is encoded independently, is available to the joint decoder, bit-planes can be extracted from it and successive bit-planes at corresponding subbands and significance levels from the second spectral band can be decoded. All bit-planes other than those from the first spectral band are encoded independently using an LDPC based Slepian-Wolf code [43, 45] and jointly decoded by a sum-product decoding algorithm. As an example of coding performance, for the NASA AVIRIS hyperspectral images data set, at medium to high quality, our baseline SW-SPIHT can achieve up to 5dB gain compared to 2D-SPIHT on individual bands, while an adaptive SW-SPIHT codec to be discussed in Section 2.6 can achieve up to 8dB gain compared to 2D-SPIHT and a comparable performance as a simple 3D wavelet system developed at NASA-JPL [8].

Note that when all bit-planes are encoded SW-SPIHT can also provide lossless compression. In many hyperspectral imaging applications preserving the spectral signature is important (e.g., the spectral signature may be used for classification and preserving classification rates becomes important [69]). SW-SPIHT provides flexibility in the choice of operating points, so that the rate can be selected in order to preserve the spectral signature. A detailed analysis is presented in Sec. 2.5.3, which demonstrates that SW-SPIHT can provide a more uniform distortion profile across bands than 3D wavelet techniques. This is shown to be advantageous in terms of preserving the spectral signature.

To the best of our knowledge we were the first to propose the application of DSC techniques in the context of hyperspectral imagery [14,67], while applying DSC for lossless hyperspectral image compression was also proposed in [6,47]. Another key novelty of our work is that we combine (i) DSC techniques operating on binary data and (ii) bit-plane successive refinement encoding based on set partitioning, a technique that has been broadly used in wavelet-based image coding. These two techniques achieve good coding efficiency by exploiting different characteristics of the input data, namely, spatial and frequency localization of wavelet coefficient energy (set partitioning) and correlation across spectral bands (DSC). We show that by combining these techniques, so that DSC is applied when it provides the most gain, a better performance is achieved than if DSC were applied directly to “raw” bit-planes (i.e. complete bit-planes, rather than set-partitioned ones). More specifically, our proposed codec relies on standard set-partitioning techniques to signal the location of “significant” wavelet coefficients, while using DSC to encode signs and refinement bits.

Note that DSC techniques require the encoder to have information about the correlation between the source being encoded and side information available at the decoder. In our application, the side information, i.e., neighboring bands, is actually available at the encoder and thus correlation can be estimated exactly. However, to estimate this correlation accurately may involve a significant overhead, in terms of memory and complexity at the encoder. Thus, another important novelty in our work is that we take into account the cost involved in estimating inter-band correlation. In this chapter we discuss a direct approach to estimate correlation and demonstrate that this results in minimal losses in compression performance. In Chapter 4 we will discuss a model-based estimation, and illustrate how the model-based approach could facilitate parallel compression of spectral bands. The proposed hyperspectral image coding algorithm has potential advantages when compared with competing techniques that exploit cross-band correlation, such as inter-band predictive methods and 3D wavelet techniques.

In inter-band prediction approaches [51], a band is predicted using previously encoded bands and the resulting prediction residuals are encoded using standard image coding techniques. As discussed in Chapter 1.3.1, there are several drawbacks of inter-band prediction approaches, namely (i) high encoding complexity due to the requirement of replicating decoder reconstruction, (ii) inherently sequential encoding, and (iii) difficult to achieve rate scalability. Our proposed DSC based algorithms can, however, address these shortcomings. First, DSC requires only access to correlation statistics, and these statistics can be reliably estimated with low complexity from uncoded data, as will be shown. Second, a DSC approach has the potential to enable parallel encoding with multiple processors. Specifically, once the inter-band correlations have been estimated, each

processor can in principle proceed independently. While correlation estimation requires data exchange across bands, this process could be much simpler than encoding/decoding, as we will discuss in more detail. This inherent parallelism can facilitate hardware implementations and greatly increases the on-board encoding speed. Third, our proposed algorithms facilitate scalability. We apply DSC to bit-planes extracted from wavelet coefficient data. A given bit-plane in a given subband depends only on the same bit-plane in a neighboring spectral band. Thus, once hyperspectral data has been encoded, efficient rate scalability can be achieved by decoding all spectral bands up to the same bit-plane resolution level. Note that the rate scalability problem in hyperspectral imaging is analogous to that in video compression scenarios, for which DSC techniques have also been proposed recently [76, 77].

3D wavelet methods provide alternatives to predictive techniques. As discussed in Chapter 1.3.1, some drawbacks of 3D wavelet methods are that they lead to expensive memory requirements or complex memory management issues. In contrast, our proposed DSC-based algorithms only require storing in memory a single spectral band at a time, once correlation statistics are estimated. These lower memory requirements could potentially lead to lower power consumption at the encoder, since a substantial amount of off-chip memory access would be avoided. This is particularly important because off-chip memory accesses often consume up to one order of magnitude higher power than on-chip data accesses [54].

This chapter is organized as follows. We first present an overview of the proposed system in Section 2.2, the baseline codec in Section 2.3, and our prediction and estimation model in Section 2.4. Implementation and experimental results for the baseline codec are

described in Section 2.5. Section 2.6 discusses the adaptive codec. Section 2.7 presents the experimental results for the adaptive hyperspectral image codec. Section 2.8 concludes this chapter.

2.2 System Overview

Figure 2.1 shows an overview of the proposed encoding system. The proposed encoder consists of multiple processors, and each processor compresses one entire spatial image at a time, using the algorithms to be discussed in detail in Section 2.3. As discussed in Section 1.2, with a DSC approach encoding needs only the correlation information to exploit inter-band redundancy. In particular, if this correlation information is available, each encoding thread compressing one spatial image would be able to proceed in parallel, and parallel encoding of hyperspectral dataset can be achieved. Therefore, one key question is how to estimate the correlation information efficiently during encoding. It is, however, nontrivial to estimate this correlation information, for the following reasons:

- The spatial images of different spectral bands are resided at different processors, and the communication bandwidth between the processors could be limited;
- Data exchanges between the processors may impact parallel processing, as the processors may have to remain idle to wait for the data.

To address these constraints, in Sections 2.4 and 4.4 we will propose several techniques to achieve accurate estimation of correlation with small amounts of inter-processors communication overhead and minimal dependencies of different encoding threads.

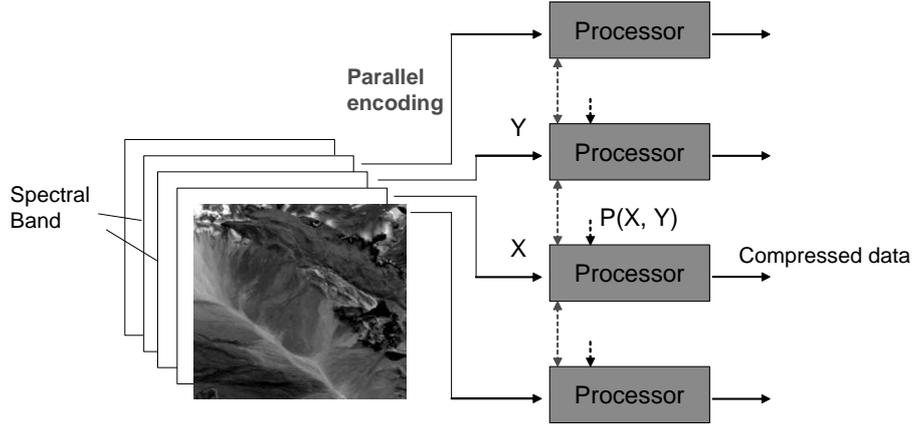


Figure 2.1: The proposed parallel encoding system. Each processor compresses one spatial image at a time.

2.3 Codec Architecture

Consider two hyperspectral bands, X and Y , and denote \hat{X} the reconstruction of band X at the decoder, which will be used to produce the side-information to decode Y ¹. This side information, \hat{X}' , is generated by linear prediction $\hat{X}' = \alpha\hat{X} + \beta$, where α and β will be estimated at the encoder.

Let us assume first that the correlation statistics are known to both the encoder and the decoder. In particular, assume that for every set of binary data to be encoded (e.g., a bit-plane or part of a bit-plane extracted from Y), we have access to the “crossover probabilities”, i.e., the probabilities that two bits in corresponding bit-plane positions of \hat{X}' and Y , respectively, are not equal. These crossover probabilities will tend to be different at each level of significance (i.e., crossover probability will tend to increase from MSB to LSB bit-planes). Section 2.4 will present techniques to estimate efficiently both

¹Note that, as will be discussed later, decoding is possible with many reconstructions of X at the decoder; as coarser versions of X are used, the reconstruction of Y will be correspondingly coarser. This facilitates rate scalability, i.e., multiple operating points can be achieved with a single embedded bitstream.

crossover probabilities and prediction parameters from input data; these techniques require processing a small fraction of pixels in spectral bands X and Y so that computation overhead is kept low.

In our work we use SPIHT [61], a well known wavelet-based image coding algorithm, as a starting point. Similar ideas could be applied to other image coding algorithms that achieve successive refinement of information by representing data in bit-planes. In each pass, SPIHT uses a significance test on wavelet coefficients to partition them into two sets: the significant set and the insignificant set. Bits corresponding to significance information are entropy coded and output by the encoder; they allow the decoder to update the list of coefficients in the significant set.

A block diagram of our proposed system is shown in Figure 2.2. Band X is encoded and decoded independently (i.e., without information from any other band) using a wavelet transform and SPIHT coding. The reconstructed band \hat{X} will then be used to form side-information to decode Y . As for band Y , the first step is again a wavelet transform $T(f, n)$ where f is the filter used in the transform and n is the number of transformation levels. Then SW-SPIHT successively updates the set of significant wavelet coefficients of Y at each pass. As shown in Figure 2.3, at the end of each iteration, a sign bit-plane, a refinement bit-plane and corresponding significance bits are generated. Sign bits and refinement bits are encoded using an LDPC-based Slepian-Wolf code and corresponding syndrome bits are output to the bitstream. However, significance bits are encoded independently using intra coding (in particular, zero tree coding in our implementation), i.e., exactly as they would have been coded in a standard SPIHT approach.

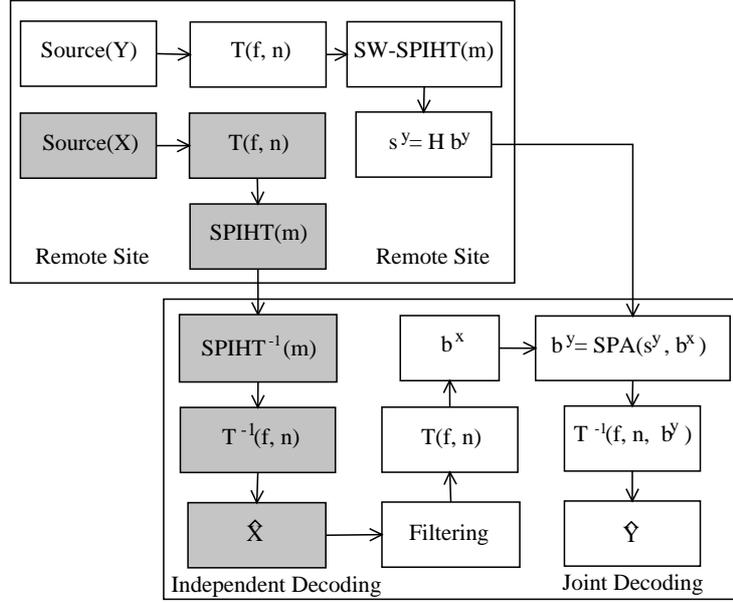


Figure 2.2: Block diagram of SW-SPIHT.

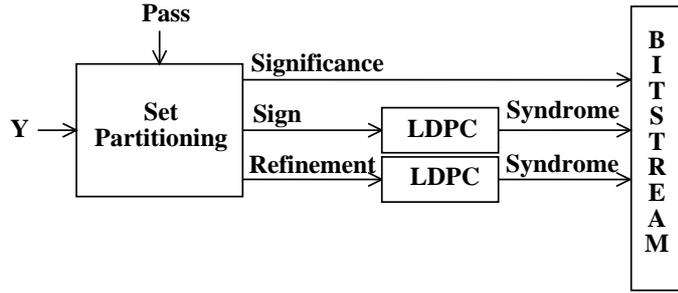


Figure 2.3: Bit-plane coding in SW-SPIHT.

In Section 2.6 we will analyze the conditions when intra coding of the significance information could be more efficient than exploiting the inter-band correlation, and propose an adaptive combination of the techniques.

In what follows, b^w , b_i^w , and $b_i^w(l)$ denote a bit-plane, the i -th bit-plane and the l -th bit of the i -th bit-plane of image W , respectively. Also in what follows, unless otherwise stated, bit-planes are sets of sign bits and refinement bits as generated after

set partitioning at a given level of significance. This is illustrated by Figure 2.3. The encoder comprises the following steps (see Figure 2.4).

E-1. Estimation of predictor coefficients α and β using a subset of information in X and Y ;

E-2. Application of the prediction coefficients to obtain wavelet transform coefficients of X' ;

E-3. Computation of wavelet transform of Y ;

E-4. At each iteration, set partitioning of the wavelet coefficients of Y to extract bit-planes b_i^y ($1 \leq i \leq m$);

E-5. Application of the significance tree of Y to the wavelet coefficients of X' to extract bit-planes b_i^x ($1 \leq i \leq m$);

E-6. Computation of \hat{p}_i , estimated crossover probability of the bit-plane pair (b_i^x, b_i^y) ($1 \leq i \leq m$) of X' and Y respectively;

E-7. Determination of the Slepian-Wolf coding rate based on the estimated crossover probability;

E-8. Generation of parity-check matrix for b_i^y ($1 \leq i \leq m$).

The compressed bitstream generated for Y includes, for each coding pass, the corresponding significance map and the syndromes generated for sign and refinement bit-planes. Note that in this algorithm it is not necessary to have access to an encoded version of X .

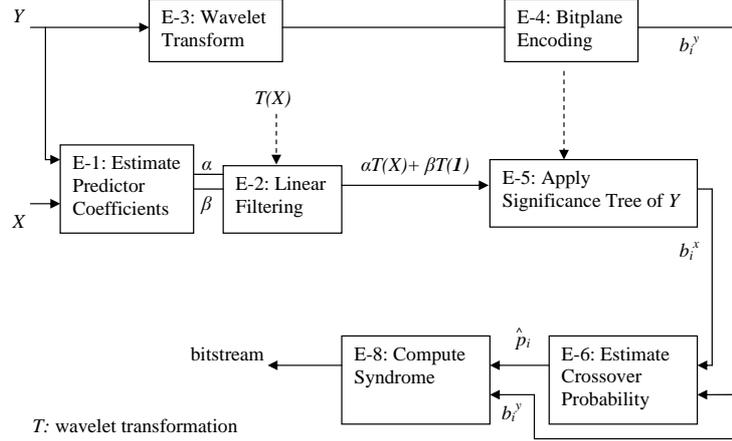


Figure 2.4: Encoding using proposed system.

Also, we will discuss in Section 2.4 how prediction coefficients and crossover probabilities can be estimated with low complexity.

At the decoder, the reconstructed \hat{X}' is transformed using $T(f, n)$, i.e., the same wavelet transformation used on Y at the encoder. Then the significance tree of Y (not X) is used to parse the wavelet coefficients of \hat{X}' in order to extract the bit-planes to be used as side information. Note that the significance tree is sent to the decoder directly (i.e., coded in “intra” mode) and thus will be available without requiring any side information. This is an important aspect of our algorithm because we have chosen to partition Y into sets before applying Slepian-Wolf coding techniques to some of the data. Thus, in order to produce the “right” side information for decoding we must apply the *same set partitioning* to \hat{X}' . The LDPC sum-product algorithm (SPA) is used to decode the bit-planes of Y given syndrome bits and side-information bit-planes from \hat{X}' .

When all bit-planes are decoded and coefficients have been refined to a desired quality level, the decoder applies the inverse wavelet transform $T^{-1}(f, n)$ to reconstruct \hat{Y} , an estimate of Y . Since Slepian-Wolf coding is used to code these bit-planes, they can be

transmitted with no or negligible information loss, as long as the correlation model is correct. Information loss would only occur if some of the crossover probabilities were underestimated. Note also that simple quality scalability can be achieved with our scheme; since any bit-plane in Y is encoded based on a single bit-plane in X , we can scale the rate by stopping the bit-plane refinement at the same level of significance in both X and Y . SW-SPIHT can also provide lossless compression for hyperspectral imagery when all bit-planes are coded, provided that an integer-to-integer wavelet transform [7] is used. Note that the least significant bit-planes tend to be uncorrelated from image to image and also have near maximum entropy; thus, in lossless applications, these bit-planes can be sent uncoded.

Crossover probabilities are used by the encoder to determine the compression rate. This rate determines which parity-check matrix should be used for a given bit-plane. In SW-SPIHT, irregular Gallager codes are used. A table is built offline that associates different crossover probabilities with random seeds for proper parity-check matrices. Once the crossover probability between a bit-plane and its corresponding side-information bit-plane is obtained, a proper parity-check matrix can be selected at run-time. To make sure the same parity-check matrix is used at the decoder, the random seed used by the encoder to generate the parity-check matrix is sent to the decoder. To match the exact bit-plane width, column puncturing and splitting is used on the parity-check matrix.

In summary our decoder comprises the following steps:

D-1. Application of prediction coefficients to obtain $\hat{X}' = \alpha\hat{X} + \beta$;

D-2. Transformation of \hat{X}' using the same wavelet transform used for Y at the encoder;

D-3. Application of the significance tree of Y to the wavelet coefficients of \hat{X}' to extract m bit-planes b_i^x ($1 \leq i \leq m$);

D-4. Computation of *a priori* probability $Pr(b_i^y(j) = 0 | b_i^x(j))$ for $b_i^x(j) = 0$ or 1 . using SPA.

Note that our proposed technique can be also extended to support multiple sources of side information. For example, if we consider encoding each bit-plane of the current band, n , which we denote X_n , using the corresponding bit-planes in the two previous bands, $n - 1$ and $n - 2$, denoted X_{n-1} and X_{n-2} , respectively, we could in theory achieve an encoding rate close to $H(X_n | X_{n-1} X_{n-2})$, and this would be smaller than that of using only single side information, $H(X_n | X_{n-i})$, $i = 1, 2$. This would require a minimal increase in complexity at the encoder (due to computation of additional prediction coefficients and crossover probabilities) but would lead an increase in decoder complexity. We tested this approach for the datasets considered in this chapter, and observed that the gains may not justify the additional complexity at the decoder except lossless or near lossless coding operation. For most bit-planes, using band $n - 1$ alone as side information already leads to significant compression gains, and a relative small conditional entropy, $H(X_n | X_{n-1})$. In our observation, the additional compression gain when using X_{n-2} as additional side information, i.e., $H(X_n | X_{n-1}) - H(X_n | X_{n-1} X_{n-2})$, tends to be relatively small. As an example, Figure 2.5 shows $H(X_n | X_{n-1})$ and $H(X_n | X_{n-1} X_{n-2})$ at different bit-planes of typical spectral bands. As shown in the figure, the reduction in coding rate achievable when using multiple bands as side information is only around 0.01 bits/sample in the more significant bit-planes, which for many lossy compression applications would not justify the

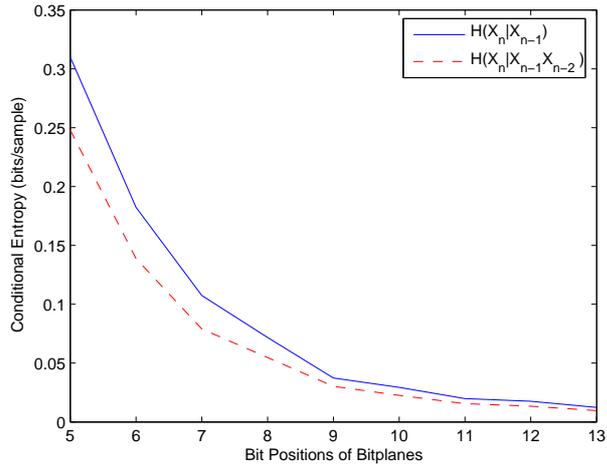


Figure 2.5: Example of conditional entropy of different bit-planes.

additional complexity at the decoder. As for the less significant bit-planes, the reduction in conditional entropy when using multiple bands as side information is larger (up to 0.05 bits/sample), so that in lossless or near-lossless scenarios multiple side information may be useful. Given that we are not focusing specifically in the near-lossless or lossless case, the rest of this chapter describes our design and experimental results based on a single band used as side information.

2.4 Estimating the Correlation and Encoder Complexity Comparison

The performance of DSC techniques depends strongly on the estimation of correlation and prediction parameters. In our system, we need to estimate two sets of parameters, namely, (i) the linear prediction coefficients, α and β , and (ii) the bit-plane crossover probabilities. In this section, we demonstrate that accurate estimation of correlation

parameters can be achieved using techniques involving a limited number of data transfers and computations. Because this estimation is accurate and requires low complexity, our proposed DSC techniques compare favorably with inter-band predictive approaches, which usually involve a substantial amount of data transfer (e.g., if a whole spectral band is predicted using another spectral band, then all pixels in the predictor image need to be fetched in order to generate a prediction residue). Reduction in the amount of data transfer is particularly important for applications operating in embedded environments, such as hyperspectral imagery compression in satellites. In these applications the encoder may only have enough internal memory to accommodate the current spectral band (since the application programs and operating systems may have occupied significant portions of the internal memory). In order to perform prediction, the system would need to fetch the relevant information from neighboring bands, which is likely to be stored in external memory. Such external memory accesses usually lead to substantial power consumption and delay. For example, while some sophisticated CPU/DSPs can handle multiple arithmetic operations in a single cycle, accessing external memory data may incur latency of the order of tens of cycles [64]. So it is desirable to reduce the total amount of data exchanged, which translates into reduction in overall system complexity.

In what follows we present low complexity techniques in estimating prediction coefficients and correlation. We also compare the encoder complexity of the proposed system with two competing techniques, namely those based on inter-band prediction and 3D wavelets.

2.4.1 Estimation of Predictor Coefficients and Correlation

The encoder can determine a rough level of correlation after it estimates α and β by computing an estimate of the residual energy after prediction. If this energy is above a certain threshold, the spectral band can be coded in intra mode (i.e., independently of other bands); with the coding mode reverting to DSC mode when the residual energy goes under the threshold. For example, Band 162 in Figure 1.8 can be coded in intra mode. Note that in real data sets we have considered, a majority of bands can be coded using DSC (e.g., 95% of bands in the Cuprite data set we use in our experiments).

2.4.1.1 Estimation of Predictor Coefficients

As discussed earlier, we use a linear predictor $X' = \alpha X + \beta$ to generate side-information for Y . The least-squares technique can be used to calculate α and β . In order to reduce the complexity (and data exchange requirements) of this process, we first down-sample the spectral bands and use only pixels in the down-sampled bands for estimation. As shown in Figure 2.6, with only 0.32% of pixels, the resulting predictor can achieve a prediction mean square error (MSE) within 0.05 of that of the optimal predictor (i.e., that computed using all pixels in X and Y). By using only a small fraction of data we reduce data exchange and computation in the least-squares calculation, without compromising the performance of the predictor (or its impact on the crossover probability estimation).

The overhead due to downsampling the data is usually negligible, as downsampling can be accomplished by incrementing the access position in data memory by a constant, and nowadays many CPU/DSPs have build-in hardware to support this operation and incur negligible overhead.

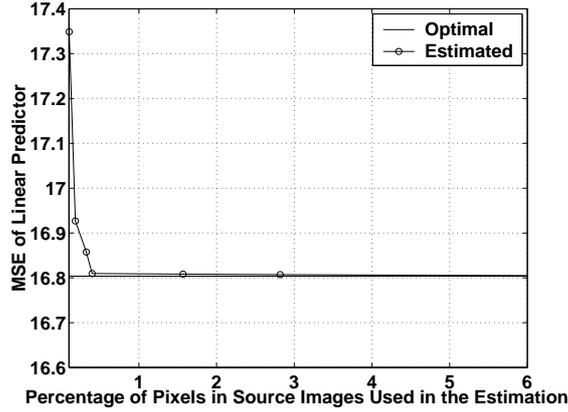


Figure 2.6: MSE under first-order linear predictor for a typical spectral band.

2.4.1.2 Estimation of Crossover Probability

We now consider estimation of the crossover probabilities at the encoder. These are needed to select an appropriate Slepian-Wolf coding rate at the encoder and to initialize the SPA at the decoder. In this section we first discuss a direct approach for correlation estimation, while in Chapter 4 we will discuss another model-based estimation. To achieve low cost estimation we propose that only a small portion of bit-plane data (generated by set partitioning) be exchanged between spectral bands. Note that, since set partitioning tends to “scramble” the ordering of coefficients, estimates of crossover probability after set partitioning are in general reliable². We use the upper bound of the 95% statistical

²Specifically, set partitioning orders the transform coefficients according to the bit levels they become significant and the zero trees they belong to [61].

confidence interval as our estimate. Specifically, the upper bound of the $(1 - \omega) \times 100\%$ confidence interval for a population proportion is given by [50]

$$\hat{p}_i = \frac{s_i}{n_i} + z_{\omega/2} \sqrt{p_i(1 - p_i)/n_i} \quad (2.1)$$

$$\approx \frac{s_i}{n_i} + z_{\omega/2} \sqrt{\frac{s_i}{n_i} \left(1 - \frac{s_i}{n_i}\right) / n_i} \quad (2.2)$$

Here \hat{p}_i is the estimate of the crossover probability of bit-plane pair (b_i^x, b_i^y) , n_i is the number of samples exchanged in estimating p_i , s_i is the number of exchanged samples for which crossover occurs, and $z_{\omega/2}$ is a constant that depends on the chosen confidence interval, e.g., $z_{\omega/2} = 1.96$ when we use a 95% confidence interval. Note that we choose the upper bound as the estimator to minimize the risk of decoding failure, at the expense of some encoding rate penalty. Statistically, with this estimation, we are $(1 - \omega) \times 100\%$ confident that the true crossover probability p_i is within $\frac{s_i}{n_i} \pm z_{\omega/2} \sqrt{p_i(1 - p_i)/n_i}$. Hence the estimation error, $\Delta p_i = \hat{p}_i - p_i$, is bounded by $0 \leq \Delta p_i \leq 2z_{\omega/2} \sqrt{p_i(1 - p_i)/n_i}$ with probability $1 - \omega$. In addition, it can be shown that (refer to Appendix A for details):

$$Pr(\Delta p_i < 0) = \omega/2 \quad (2.3)$$

$$Pr\left(\Delta p_i > 2z_{\omega/2} \sqrt{p_i(1 - p_i)/n_i}\right) = \omega/2, \quad (2.4)$$

which allows us to bound in a systematic way the probability of decoding error and the probability of incurring a large encoding rate penalty. Since the estimation process consists of simply counting occurrences of crossovers in small portions of two bit-planes, the overall estimation overhead is small.

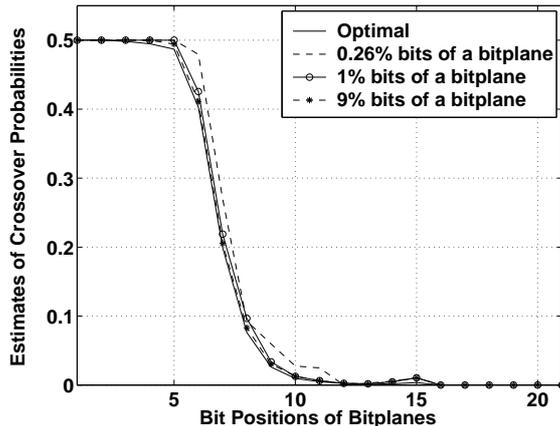


Figure 2.7: Example of estimation of crossover probability. Note that for all bit-planes it is possible to achieve an accurate estimate using a small percentage of data in a bit-plane.

As an example of the accuracy of crossover probability estimation using this low complexity technique, Figure 2.7 shows a typical estimation result using different percentages of data from a bit-plane. As an example, with 5% of bits exchanged the crossover probability estimate is within 0.003 of the actual crossover probability. Since we choose the compression rate to leave a margin of about 0.05 bits over the Slepian-Wolf limit (as estimated by $H(\hat{p}_i)$, since we assume the source model as in [43]), this estimation accuracy is sufficient. In addition, we also test this technique in our coding performance experiments (details in Section 2.5). There we use around 10% of data in a bit-plane for correlation estimation and our experimental results show that the estimates are accurate enough that no decoding errors occur.

2.4.2 Encoder Complexity Comparison

In this section we compare the encoder complexity of our proposed scheme to that of inter-band prediction and 3D wavelet approaches.

2.4.2.1 Comparison with Inter-band Prediction

Inter-band prediction approaches need to generate exact copies of the decoded bands at the encoder, so that the encoder needs to perform decoding as well. To encode the current band Y using neighboring band X for prediction, the inter-band encoder requires following steps (Figure 2.8):

- I-1.** Estimation of predictor coefficients α^* and β^* (in this case approximate techniques could also be used as long as the chosen parameters are communicated to the decoder);
- I-2.** Application of the prediction coefficients to obtain $\hat{X}' = \alpha^* \hat{X} + \beta^*$;
- I-3.** Computation of $Y - \hat{X}'$ to generate the residue;
- I-4.** Transformation of residue using the wavelet transform;
- I-5.** Set partitioning on the wavelet coefficients of residue. Output bitstream;
- I-6.** Inverse set partitioning;
- I-7.** Inverse transformation;
- I-8.** Adding \hat{X}' to the output of inverse transformation to generate \hat{Y} .

Comparing the encoding steps of our proposed scheme (Figure 2.4) with inter-band prediction approach (Figure 2.8), we can make the following observations:

- (i) Both schemes need to compute the wavelet transform and perform bit-plane encoding of the current band Y (Steps (E-3) and (E-4) in Figure 2.4, Steps (I-4) and (I-5) in Figure 2.8).

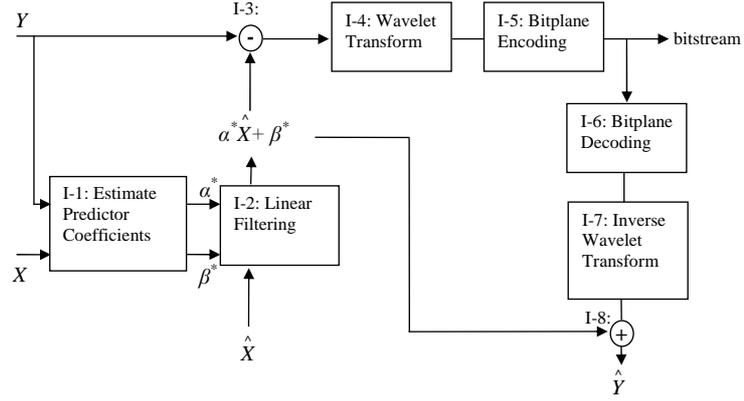


Figure 2.8: Encoding using inter-band prediction.

- (ii) The inter-band prediction approach has to perform an inverse wavelet transform (I-7). In our proposed scheme, we need wavelet coefficients of the linear predictor X' for correlation estimation. However, forward transformation is not necessary here since wavelet coefficients of X have been computed during the compression of previous band, and we can compute wavelet coefficients of X' simply by

$$T(X') = T(\alpha X + \beta) = \alpha T(X) + \beta T(\mathbf{1}) \quad (2.5)$$

where T denotes wavelet transformation, and $\mathbf{1}$ is a vector of ones. We pre-compute $\beta T(\mathbf{1})$ and use it for all the coefficients in a spectral band. Therefore, computing $T(X')$ in our proposed algorithm requires one multiplication and one addition per coefficient as suggested by (2.5), and this amount of computation is in general less than that of inverse transform (I-7) in inter-band prediction approach.

- (iii) In the inter-band approach, encoder needs to perform bit-plane decoding (I-6)³. In our proposed system we apply the significance tree of Y to the wavelet coefficients of X' to extract bit-planes (E-5), for crossover probability estimation. Note that in (E-5) we merely extract coefficients according to the significance tree of Y , and no significance test on partition is required, so this is similar to (I-6) in terms of complexity. We would like to emphasize that the complexity of our system can be further reduced by avoiding bit-plane extraction, since there are low complexity alternatives for correlation estimation, as will be discussed in Chapter 4.
- (iv) The inter-band prediction approach requires subtracting the predictor from the current band to compute the residue (I-3), and then adding back the predictor to the reconstructed residue (I-8). Since the subtraction/addition has to be performed on every pixel, the complexity here is of the order of the amount of data in one band. On the other hand, our proposed scheme needs only a small portion of data to estimate crossover probabilities (E-6). Also the complexity of generating syndrome (E-8) is linear (since the parity check matrix is sparse), and is of the order of the number of bit-planes we need to encode, which is usually small since in many lossy compression applications only high significant bit-planes are transmitted.

Based on the above comparisons, we conclude that our scheme requires lower encoding complexity than inter-band prediction approaches.

³ [51] has proposed using only full bit-planes to form the predictor. This could avoid bit-plane decoding at the encoder, but leads to performance degradation. In the general case when one wants to truncate at the middle of bit-plane, some decoding of the significance information or bookkeeping are necessary to determine the order of the wavelet coefficients.

2.4.2.2 Comparison with 3D Wavelet Approaches

3D wavelet approaches operate on multiple spectral bands at the same time. This usually incurs substantial external memory access overheads in storing intermediate results. For example, using 3D wavelet approaches, 3D wavelet coefficients need to be computed first, followed by set-partitioning of the 3D wavelet coefficients. Since the internal memory may not be able to accommodate several bands of 3D wavelet transform coefficients, they need to be transferred back-and-forth between external and internal memory. In contrast, our proposed scheme operates on each single spectral band independently once the inter-band correlation has been estimated, and wavelet transformation and bit-plane encoding of a single spectral band could be in general completed entirely in internal memory without incurring external memory access for storing intermediate data⁴. Hence the data access overheads in our scheme are much smaller than those involved in a 3D wavelet approach.

2.5 Experimental Results of Baseline Codec

We have implemented SW-SPIHT and applied it to the 16-bit hyperspectral images. Our C program implementations of the set partitioning and bit extraction algorithms are derived based on a MATLAB implementation of 2D-SPIHT [73]. The SPA we used for SW-SPIHT is based on the algorithm in Section III-A of [46], and is modified according to [68] for Slepian-Wolf decoding.

⁴As an example, storing the transform coefficients of a single $512 \times 512 \times 16$ -bit spectral band would need $512 \times 512 \times 24$ -bit buffer memory, or 768K bytes. Note that transform coefficients have larger dynamic ranges than the original pixel data.

In our experiments we use data sets originally comprising 224 spectral bands, each of size 614×512 pixels. Due to constraints of the implementation of our codecs (e.g., our implementation of wavelet transform can handle only power-of-two dimension data), in the experiments we compress 512×512 pixels in each band, and in total 192 bands starting from band number 33. Experimental results use SNR and PSNR for the comparison on individual frames and multiband SNR (MSNR) and multiband peak SNR (MPSNR) for the whole spectrum. These quantities are defined as follows:

$$\text{MSE} = E[(x - \hat{x})^2] \quad (2.6)$$

$$\text{SNR} = 10 \log_{10} \left(\frac{E[x^2]}{\text{MSE}} \right) \quad (2.7)$$

$$\text{PSNR} = 10 \log_{10} \left(\frac{(65535)^2}{\text{MSE}} \right) \quad (2.8)$$

$$(2.9)$$

where, $E(\cdot)$ is the expectation operator over pixels from an image band. x is the 16-bit value of a source pixel and \hat{x} is the 16-bit value of reconstructed pixel of x . Also,

$$\text{MSNR} = 10 \log_{10} \left(\frac{E[x^2]}{\text{MSE}} \right) \quad (2.10)$$

$$\text{MPSNR} = 10 \log_{10} \left(\frac{(65535)^2}{\text{MSE}} \right) \quad (2.11)$$

where, now $E(\cdot)$ is the expectation operator over pixels from *all* spectral bands. The rates for individual image band are measured in bits per pixel (bpp) and those for the whole spectrum are in bits per pixel per band (bpbpb).

The outline of this experimental study is given as follows: First, we provide a comparison in terms of rate-distortion performance between SW-SPIHT and predictive 3D-SPIHT. Second, we compare SW-SPIHT with predictive 2D-SPIHT. In these experiments, we use different scenes and sites from the NASA AVIRIS data set including Cuprite Radiance (SC01), Moffet Field Radiance (SC03) and Lunar Lake Reflectance (SC02).

In order to describe these alternative codecs and our implementations of them, we need following notations:

1. A denotes a general image band.
2. B_i denotes the i -th image band from the spectrum.
3. $\mathbf{1}$ is the vector with all 1 elements. The dimension is set as the number of pixels used by the least squares predictor.
4. $V(A)$ is the function to vectorize a fixed number of pixels from image band A .
5. For the predictor image bands A and source image band B_i , $\alpha(A, B_i)$ is the prediction slope coefficient and $\beta(A, B_i)$ is the prediction intercept coefficient.
6. $B'_i(A)$ denotes the band after regression using least squares prediction, and the design matrix is given by $X = (\mathbf{1}, V(A))$. Recall that the least squares coefficients are given as follows:

$$(\beta, \alpha)^\tau = (X^\tau X)^{-1} X V(B_i)$$

where, τ is the transpose operator.

7. Regression residuals of the least squares predictor of the i -th frame can be computed as $B_i - B'_i$.

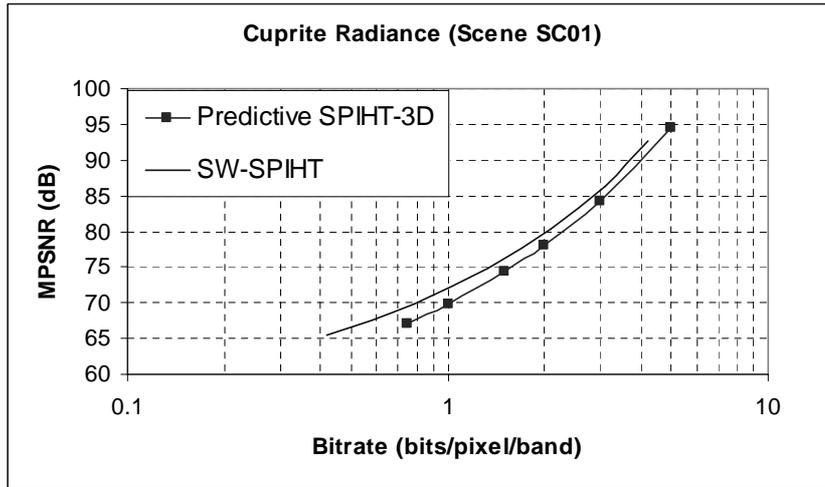


Figure 2.9: Rate-distortion curves of SW-SPIHT and predictive 3D-SPIHT - Cuprite.

2.5.1 Rate-Distortion Comparison with 3D Wavelet Approaches

Before presenting the rate-distortion comparison of SW-SPIHT with a predictive variant of 3D-SPIHT, we briefly describe these codecs and how we implemented them.

We modify 3D-SPIHT to adjust the bands taking into account their correlation. Thus, instead of operating on the original bands, (B_1, B_2, \dots) we apply the wavelet transform and encoding to a new set of bands, (B'_1, B'_2, \dots) , obtained as follows:

1. $B'_1 = B_1$.
2. For all $i > 1$, $B'_i = \alpha(B_i, B'_{i-1})B_i + \beta(B_i, B'_{i-1})$, and $\alpha(B_i, B'_{i-1})$ and $\beta(B_i, B'_{i-1})$ are directly encoded into the bitstream.

We use this predictive 3D-SPIHT approach so as to better “align” all spectral bands, so that wavelet transform can better exploit the inter-band correlation.

Figures 2.9 to 2.11 provide coding performance comparisons for the radiance data from the Cuprite and Moffet Field sites, and the reflectance data from the Lunar Lake

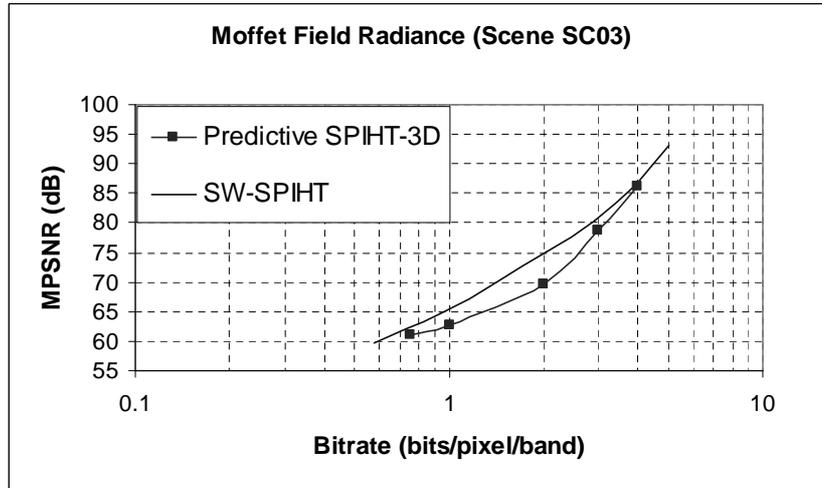


Figure 2.10: Rate-distortion curves of SW-SPIHT and predictive 3D-SPIHT - Moffet Field.

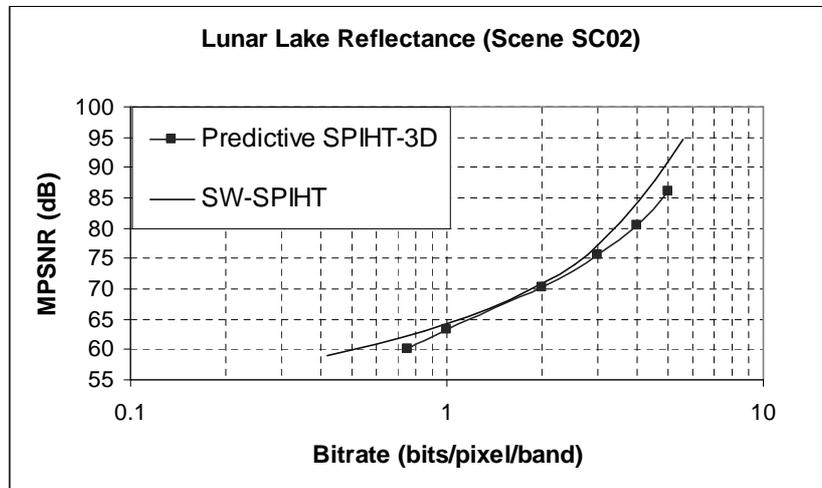


Figure 2.11: Rate-distortion curves of SW-SPIHT and predictive 3D-SPIHT - Lunar Lake.

site (log scale is used for the rate to facilitate the comparison at low bit-rates). To obtain the results of our predictive 3D-SPIHT, we first align the spectral bands as discussed in the previous paragraph and compress B'_1, B'_2, \dots with an implementation of 3D-SPIHT available in the public domain [23, 24]. It can be seen that SW-SPIHT performs competitively, with some gain over 3D-SPIHT at most rate regions. In addition SW-SPIHT has moderate memory requirements for encoding. It should be noted that the performance of 3D-SPIHT can be improved by applying entropy coding (e.g., arithmetic coding) on the output bits. Similarly, we can improve SW-SPIHT by applying entropy coding on the significance bits information. Also note that results for 3D-SPIHT without prediction (not included here) are close to predictive 3D-SPIHT with a marginal loss at low bit-rates.

It is well known that wavelet set-partitioning based codecs can precisely control the bit-rate. In other words, the SNR can be kept at a required level when the bit-rate is allowed to change. However this only holds for global SNR, and not necessarily for different parts of the encoded stream. In the case of 3D-SPIHT, the SNR of individual spectral bands can actually fluctuate significantly for a given target global SNR (variations of up to 5dB are possible, see Fig. 2.12 for an example). Another salient feature of SW-SPIHT is that it allows targeting individual band SNRs, so that fluctuations across bands can be kept very small (e.g., within 1 dB). Note that these variations are undesirable, as they could destroy the spectral signatures that are of primary interest in analysis of hyperspectral imagery. Refer to Section 2.5.3 for an example of how SW-SPIHT is better at preserving these spectral signatures.

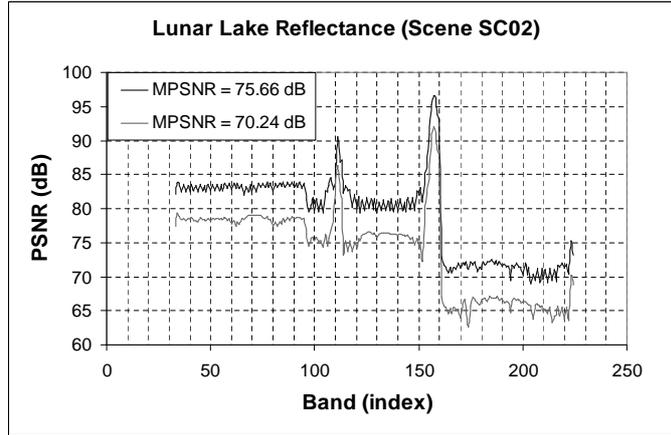


Figure 2.12: Inter-band SNR fluctuation under 3D-SPIHT.

2.5.2 Rate-Distortion Comparison with 2D Wavelet Approaches

We have also compared SW-SPIHT with two other 2D wavelet based codecs. The first is the standard 2D-SPIHT codec that operates independently on all spectral bands, without cross-band prediction. The second is the predictive 2D-SPIHT codec, which operates as follows:

1. The first image band B_1 is encoded as is.
2. \hat{B}_{i-1} , reconstruction of image band B_{i-1} , is used to obtain a predictor for B_i , B'_i .
3. 2D-SPIHT codec is applied to $B_i - B'_i$ for all $i > 1$; if the residual energy is above a certain threshold then B_i is encoded directly.
4. Prediction coefficients $\alpha(\hat{B}_{i-1}, B_i)$ and $\beta(\hat{B}_{i-1}, B_i)$ are sent as overhead.

Note that the predictor used in predictive 2D-SPIHT is the preceding image band, and this is different from the predictor used in the predictive 3D-SPIHT codec. The 2D-SPIHT we used to compress the images (in the intra-band codec) or the residues (in the

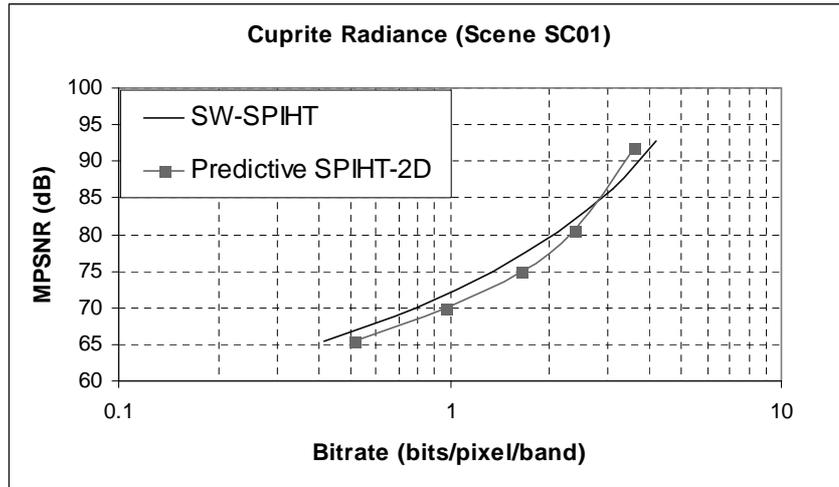


Figure 2.13: Rate-distortion curves of SW-SPIHT and predictive 2D-SPIHT - Cuprite.

predictive codec) is the same MATLAB implementation from which our SW-SPIHT was derived [73].

Figures 2.13 to 2.15 provide comparisons based on the radiance data from the Cuprite and Moffet Field sites, and the reflectance data from the Lunar Lake site. For Cuprite, SW-SPIHT achieves some gain at middle range bit-rates, but suffers marginal loss at high bit-rates. The coding performance of predictive 2D-SPIHT improves at high bit-rates thanks to the better quality reconstruction used as predictor. For Moffet Field and Lunar Lake sites, SW-SPIHT achieves marginal gain consistently, demonstrating competitive rate-distortion performance ⁵.

⁵Note that there are several significant differences between our system and predictive 2D-SPIHT. Our scheme applies set-partitioning to the transform coefficients of the original band and DSC is used to further compress the sign/refinement bits after set-partitioning. In contrast, predictive 2D-SPIHT applies the wavelet transform to the prediction residue and then uses set-partitioning as the only “entropy coding” tool (i.e., sign/refinement bits are not compressed; this is true for 3D-SPIHT as well). Since, typically, the prediction residue has much smaller energy than the original band, set-partitioning on the residue can lead to fewer bits required to represent the significance tree, as well as fewer refinement and sign bits, as compared to set-partitioning on the original band. This gain is more significant at low rates. Moreover, while our proposed algorithms ignore the noise symbols dependency (we model the correlation noise as a simple i.i.d. source), predictive 2D-SPIHT exploits the spatial correlation that might remain in

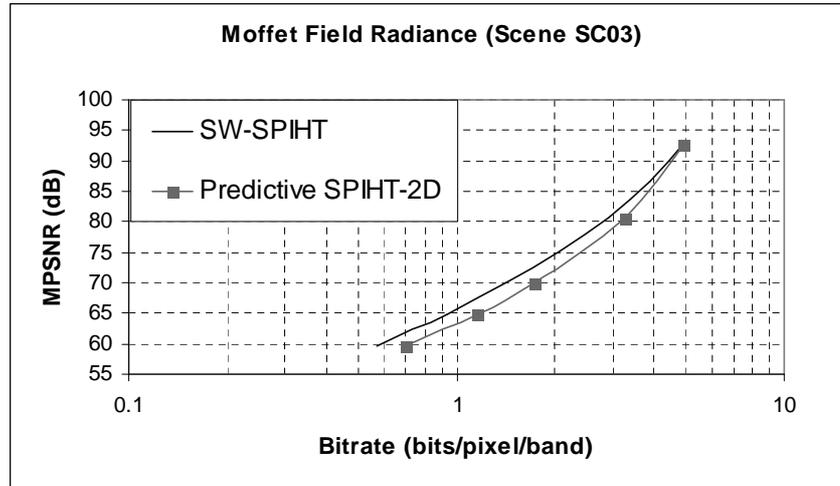


Figure 2.14: Rate-distortion curves of SW-SPIHT and predictive 2D-SPIHT - Moffet Field.

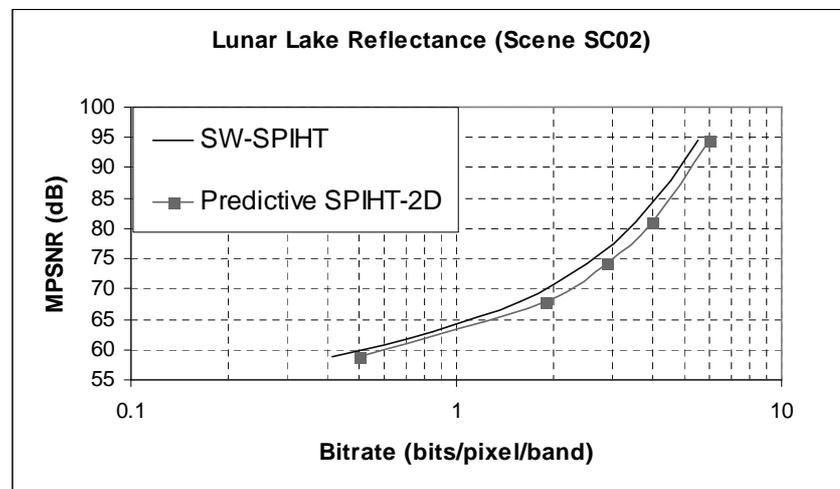


Figure 2.15: Rate-distortion curves of SW-SPIHT and predictive 2D-SPIHT - Lunar Lake.

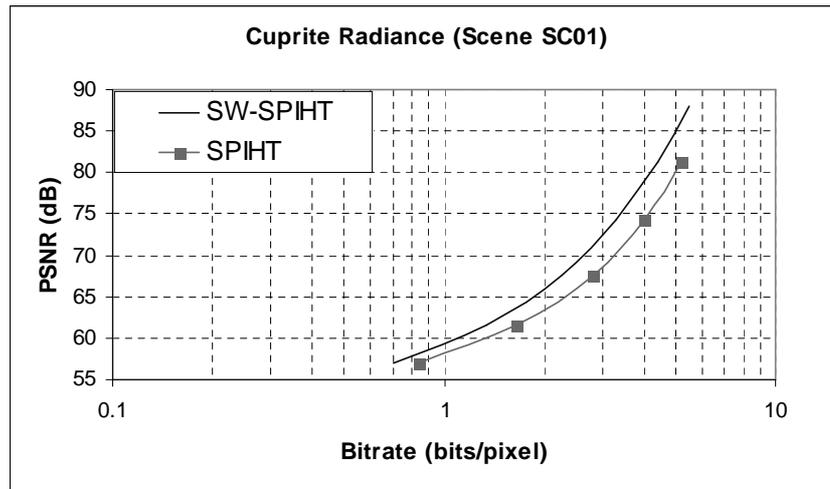


Figure 2.16: Rate-distortion curves of SW-SPIHT and SPIHT (Site: Cuprite, View: SC01, Band: 40).

Figures 2.16 to 2.18 compare the performance of SPIHT and SW-SPIHT on individual image bands. We selected three pairs of bands from different spectral regions where the levels of correlations are different, as also shown in Figure 1.8. We did not select bands in spectral regions where the predictor sees large surges in mean square residuals, since these bands have low correlation and intra-coding is used instead. As shown in the figures, SW-SPIHT outperforms SPIHT significantly, with up to 5dB gain in some rate regions. There are some variations in the PSNR gain due to variations of the energy among these images and correlations between images in these pairs.

the residue, leading to some additional coding gain. Note that some advanced 3D wavelet methods (e.g., 3D-ICER developed in NASA-JPL [38]) also utilize this noise symbols dependency through modified 3D decomposition to improve performance.

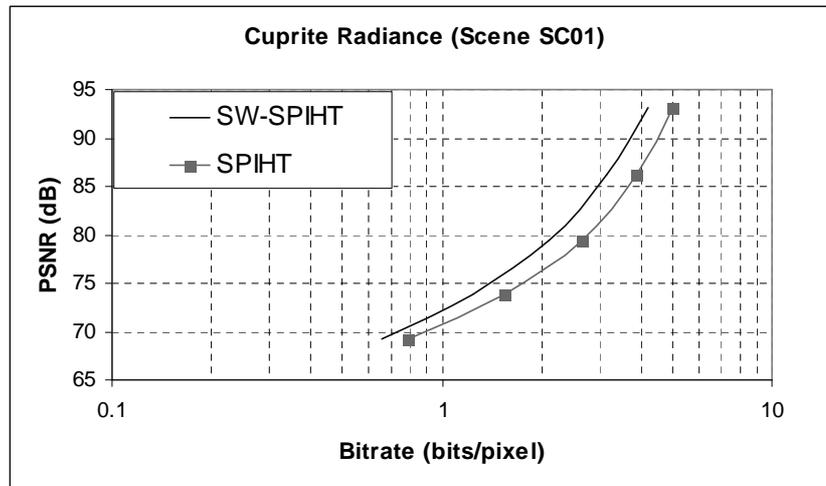


Figure 2.17: Rate-distortion curves of SW-SPIHT and SPIHT (Site: Cuprite, View: SC01, Band: 133).

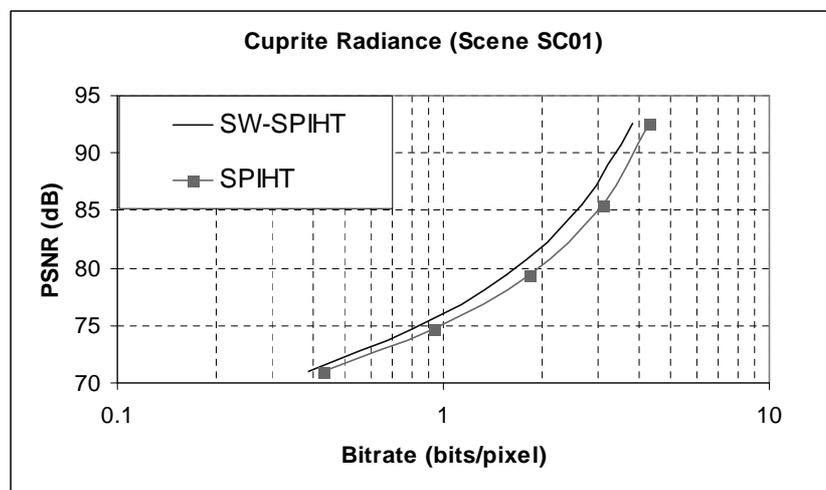


Figure 2.18: Rate-distortion curves of SW-SPIHT and SPIHT (Site: Cuprite, View: SC01, Band: 190).

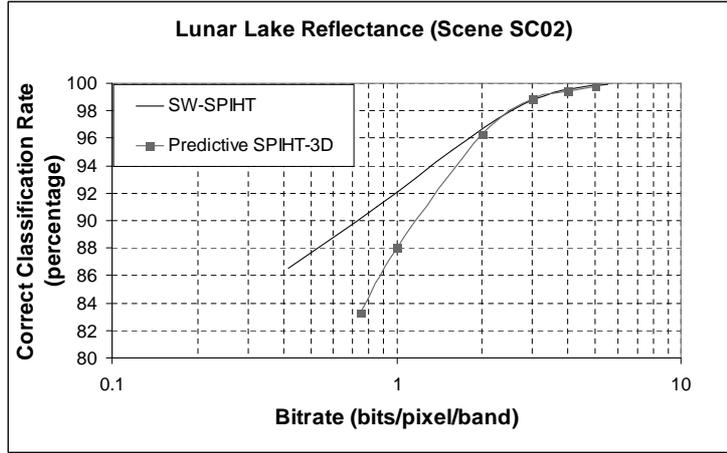


Figure 2.19: Classification performance (Site: Lunar Lake, View: SC02).

2.5.3 Preservation of Spectral Signature

As mentioned earlier, SW-SPIHT allows encoding with very consistent quality across bands, a property that cannot be guaranteed with 3D-SPIHT. To illustrate the potential advantages of SW-SPIHT in terms of signature preservation, we have also assessed its performance in a remote sensing classification application. We tested our system with the Spectral Angle Mapper algorithm (SAM) [18], which is a well-known algorithm designed to measure the similarity between the unknown test spectra and the reference spectra. Similar to the set-up in [69], we assume the classification results of the original image are correct, and measure the number of pixels of the reconstructed image which have the same classification results as the original image pixels. Figure 2.19 provides a comparison in terms of classification performance. As shown in the figure our proposed approach outperforms 3D-SPIHT in general. This is because our approach can keep the variation of SNR small across bands. As a result, spectral signatures can be better preserved.

2.6 Adaptive Coding Strategy

In Section 2.3 we have discussed a compression algorithm that applies DSC to sign and refinement bits, and intra coding to significance information. We now discuss a coding scheme which adaptively applies DSC or intra coding to bit-plane data according to bit-plane statistics and correlation, with the algorithm discussed in Section 2.3 being a particular case when switching does not occur, as will be discussed. Experiment results suggest the adaptive coding strategy can lead to considerable improvements to the hyperspectral image system. To justify the adaptive coding strategy, we first analyze the coding gains of intra coding/DSC tools under different bits extraction scenarios. Figure 2.20 each column corresponds to a wavelet coefficient, and we extract and encode the bits bit-plane by bit-plane starting from the most significant bit-plane. As is usually done in wavelet image compression, we extract a sign bit only when the corresponding coefficient becomes significant. The extracted sign bits can be encoded by DSC or intra coding (e.g., arithmetic coding). As will be discussed in detail in the next section, it is generally more efficient to encode sign bits using DSC.

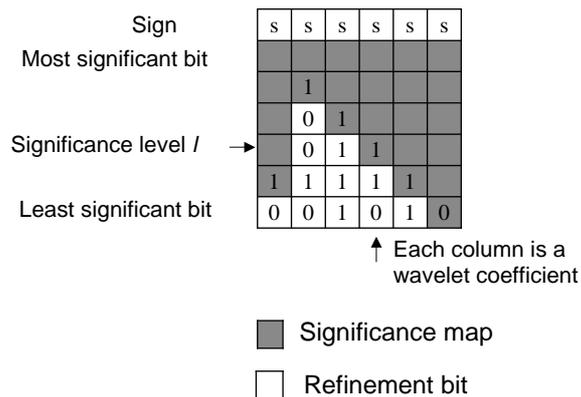


Figure 2.20: Bit extraction.

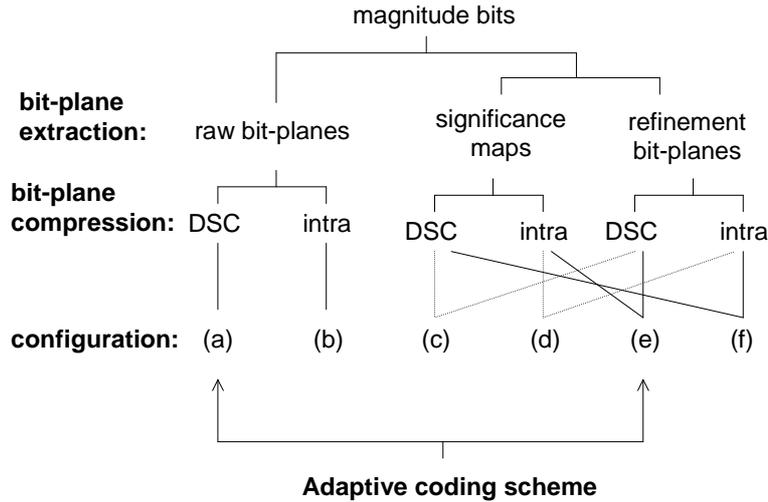


Figure 2.21: Coding configurations.

The magnitude bits can be extracted in two different ways. We can extract and encode all the magnitude bits of the same significance level (i.e., a raw bit-plane) in one single pass. Alternatively, we can partition the magnitude bits into significance maps and refinement bits, and encode them separately using DSC or intra coding (zerotree coding in our case). Therefore, it is possible to compress the magnitude bits using several different *coding configurations*, each representing a possible combination of DSC and intra coding under different bit extraction scenarios (Figure 2.21). Our goal is to select and appropriately combine some of these configurations so that the optimal overall coding performance can be achieved. In what follows we will first quickly dismiss several configurations and then examine the remaining ones in more detail. As will be discussed, the optimal coding strategy involves judicious application of configurations (a) and (e) depending on bit significance levels and wavelet subbands, leading to the proposed adaptive coding scheme.

In Figure 2.21, both configurations (b) and (d) utilize only intra coding. However, (d) has been found to be more efficient, as it exploits the differences in the zero-th order statistics of the significance maps and refinement bits. In addition, there exist efficient methods to jointly encode the bits and convey the overhead for classifying the bits (e.g., set-partitioning in SPIHT). Therefore, we eliminate (b). We also eliminate (c) and (f) as both utilize DSC to encode significance maps, which could potentially result in a vulnerable system. This is because significance maps carry important structural information about the positions of significance and refinement bits. While a single error in the significance maps could lead to incorrect decoding of all the remaining bits, DSC usually has a small but non-zero probability of decoding failure. This is true in particular in our application, where it is infeasible to adopt the feedback architecture proposed in the literature [28] due to long delay.

In the following sections, we discuss sign bits compression by DSC or intra coding, and magnitude bits compression using configurations (a), (d) and (e). Note that the difference between (d) and (e) is in terms of compression of refinement bits and will be discussed in Section 2.6.1, while the differences between (a) and (e) are in the extraction and compression of significance bits, which will be discussed in Section 2.6.2.

2.6.1 Refinement/Sign Bits Compression

It is well known that the refinement bits of wavelet coefficients are almost random (i.e., marginal probability of the bits is close to 0.5), and this can be shown by inspecting the distributions of the wavelet coefficients. Wavelet coefficients can be modeled by Laplacian distribution. Figure 2.22(a) shows how to estimate, from the p.d.f. of X_i (coefficients in

the i th subband), the probability that a refinement bit being zero and the probability that a refinement bit being one. As shown in the figure, the probabilities are almost the same, and hence intra coding cannot achieve much compression for refinement bits in general. Similarly, when a coefficient becomes significant, the probabilities of the coefficient being positive and being negative are almost the same (Figure 2.22(b)). Therefore, sign bits are almost random, with marginal probability close to 0.5, and intra coding in general cannot achieve much compression for sign bits as well⁶.

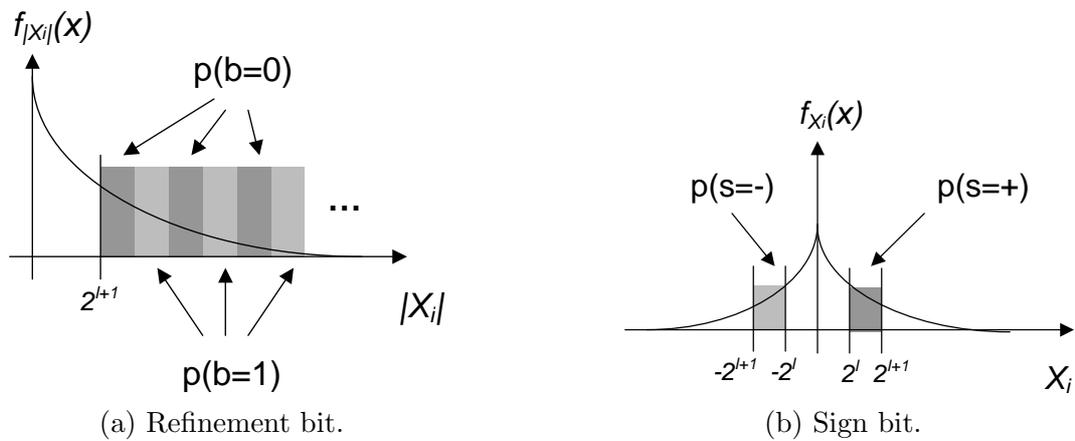


Figure 2.22: Probability that a refinement/sign bit being zero or one.

On the other hand, the source X_i and SI Y_i are correlated (Y_i are i th wavelet sub-band coefficients of the previous spectral band after linear prediction), and samples of (X_i, Y_i) concentrate mostly near the diagonal in a scatter plot. Therefore, it is possible to compress refinement/sign bits by exploiting inter-band correlation. For refinement bits, the events of crossover correspond to the regions A_j in the sample space of X_i and Y_i in Figure 2.23(a). Usually there are only a few samples that occur inside the off-diagonal

⁶Note that some wavelet-based image codecs, e.g., JPEG 2000, assign different probability models to refinement and sign bits depending on some information of the neighboring coefficients, and achieve some rate reduction through these conditional codings [71].

crossover regions. Therefore, the crossover probability is usually small, and substantial compression of refinements bits can be achieved by DSC ⁷.

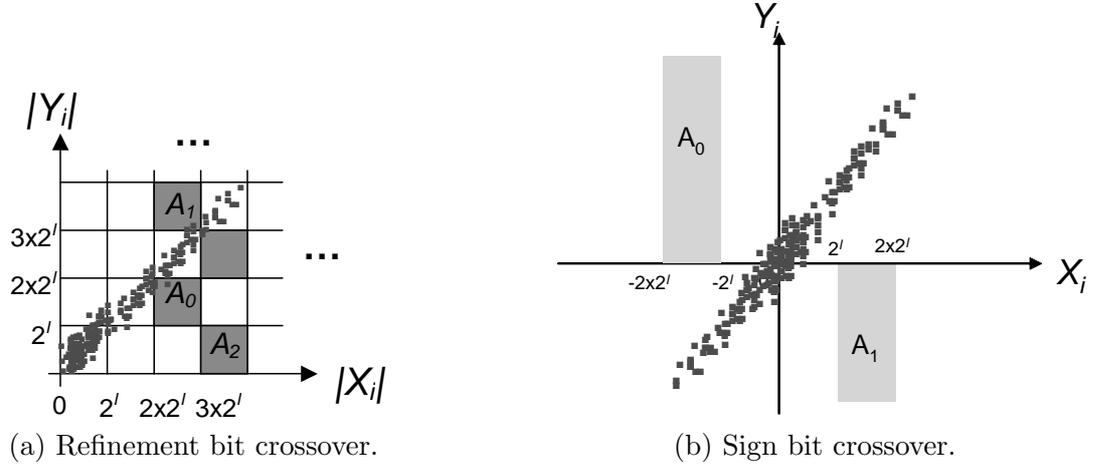


Figure 2.23: Events of refinement/sign bits crossover.

Similarly, for the sign bits, the crossover events correspond to the regions A_j in the sample space of X_i and Y_i in Figure 2.23(b). The probability of sign crossover is usually small, except for the lowest significance levels (small l), when the crossover regions are near the origin (A_j starts at $|X_i| = 2^l$) and there are more samples occurring inside the sign crossover regions.

Therefore, in our system, we compress sign and refinement bits with DSC to exploit inter-band correlation. This eliminates configuration (d) for magnitude bits compression, and only (a) and (e) are left for further evaluation.

⁷Under some assumptions, the theoretical compression rate of DSC for binary source is given by $H(p)$, where $p \leq 0.5$ is the crossover probability between the source and SI [43]. Therefore, the compression efficiency of DSC increases as the crossover probability becomes smaller.

2.6.2 Significance Maps Compression and Raw Bitplane Coding

The significance map is biased toward zero in general. This can be verified from the distribution of the coefficients (Figure 2.24(a)). Therefore, intra coding can lead to effective compression for significance maps. However, the bias would decrease for the lower significance levels (when l is small) as shown in Figure 2.24(b). Accordingly, intra coding would become less efficient when coding the least significant bit-planes. In addition, the bias would decrease as the variance of the coefficients increases (see Figure 2.24(b); a more rigorous mathematical justification will be given later). Therefore, for low-pass subbands and higher level of decomposition subbands, intra coding may not be very efficient.

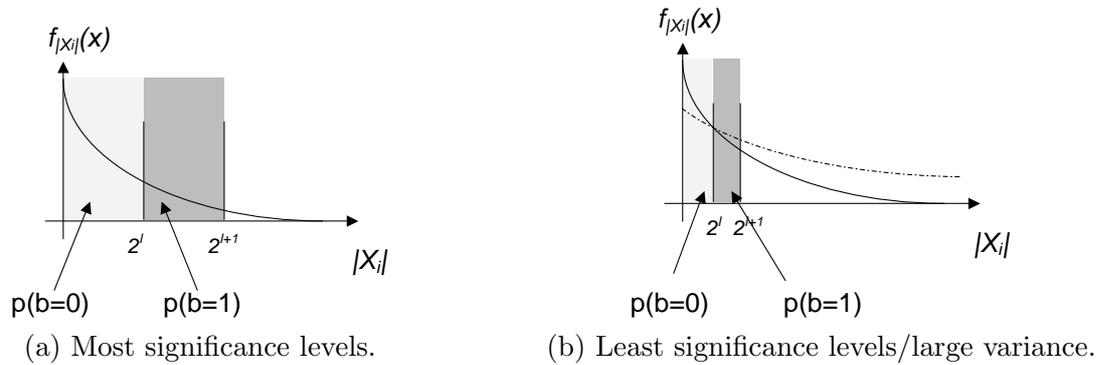


Figure 2.24: Bias in significance maps.

Alternatively, instead of partitioning into significance and refinement bits, we can extract the magnitude bits as raw bit-planes and apply DSC to exploit inter-band correlation. For raw bit-plane, the events of bit crossover correspond to the regions A_j in Figure 2.25(a). The probability of raw bit crossover is usually small, as there are only a few samples (X_i, Y_i) that occur inside A_j . Therefore, DSC can achieve compression. On the other hand, when the significance level l is small, the area of each crossover region decreases (A_j are square regions with length 2^l), and they become more evenly distributed

over the sample space (Figure 2.25(b)). As a result, more samples would occur within the crossover regions, and DSC would also become less efficient as we encode least significant bit-planes.

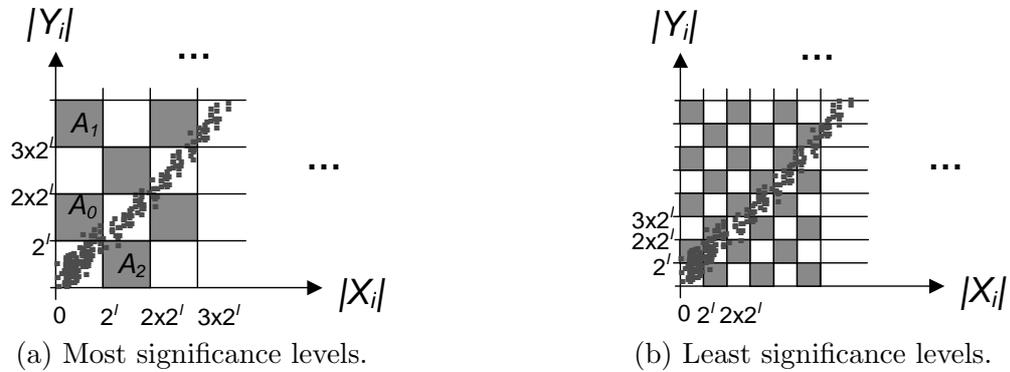


Figure 2.25: Raw bit-planes crossover.

To summarize, both intra coding of significance information and inter-band coding of raw bit-planes can potentially achieve considerable compression for magnitude bits, and configurations (a) and (e) are promising candidates. However, it is unclear which one we should use in different situations to achieve the optimal coding performance. Therefore, we propose modeling techniques to precisely analyze their performance for different source characteristics and correlation levels.

2.6.3 Modeling

The modeling techniques estimate the number of *coded* bits for configurations (a) and (e). Recall in configuration (a) we encode the entire raw bit-plane using DSC, whereas in (e) we partition the bits into refinement bits and significance maps and encode them separately using DSC and intra coding respectively (Figure 2.21). To determine the optimal coding strategy, we estimate and compare the number of coded bits at each significance level l

and for each wavelet subband i . The estimated number of coded bits for configuration (a) is given by:

$$N_i \times (H(p_{raw}(l, i)) + m), \quad (2.12)$$

where N_i is the number of uncoded raw bits at a given significance level in wavelet subband i (hence N_i is equal to the number of coefficients in subband i), and $p_{raw}(l, i)$ is the estimated crossover probability of the raw bit-plane at significance level l for coefficients in wavelet subband i . $H(p_{raw}(l, i))$ is the theoretical compression rate using DSC, and we add a margin m to account for the performance of practical systems. The estimates for $p_{raw}(l, i)$ can be derived by integrating the joint p.d.f. $f_{X_i Y_i}(x, y)$ over the crossover regions A_j as shown in Figure 2.25. Details will be provided in Chapter 4.4, where we discuss how to estimate crossover probabilities for the purpose of determining the encoding rate.

The estimated number of coded bits for configuration (e) is given by:

$$N_{ref}(l, i) \times (H(p_{ref}(l, i)) + m) + N_{signif}(l, i) \times \gamma(l, i), \quad (2.13)$$

where $N_{ref}(l, i)$ and $N_{signif}(l, i)$ are the number of uncoded refinement bits and significance map bits at significance level l in subband i respectively, $p_{ref}(l, i)$ is the estimate for refinement bits crossover probability, and $\gamma(l, i)$ is the compression ratio achieved by intra coding. $H(p_{ref}(l, i)) + m$ is the compression ratio achieved by a practical DSC scheme, and we can estimate $p_{ref}(l, i)$ as in Chapter 4. We model the significance map bits as an i.i.d. binary source and estimate $\gamma(l, i)$ by $H(p_0(l, i))$, where $p_0(l, i)$ is the probability that significance map bits being zero. Assume the wavelet coefficients in subband i are

Laplacian distributed with parameter β_i , i.e., $f_{X_i}(x) = \frac{1}{2}\beta_i e^{-\beta_i|x|}$, following from the definitions of refinement bits and significance maps, $N_{ref}(l, i)$, $N_{signif}(l, i)$ and $p_0(l, i)$ can be estimated by

$$N_{ref}(l, i) = N_i \exp(-\beta_i 2^{l+1}) \quad (2.14)$$

$$N_{signif}(l, i) = N_i(1 - \exp(-\beta_i 2^{l+1})) \quad (2.15)$$

$$p_0(l, i) = \frac{1 - \exp(-\beta_i 2^l)}{1 - \exp(-\beta_i 2^{l+1})} \quad (2.16)$$

We found that $H(p_0(l, i))$ is a good estimate for the compression efficiency of zerotree coding.

2.6.4 Adaptive Coding Scheme

We compare (2.12) and (2.13) to determine the optimal coding configuration at each significance level and for each wavelet subband. Figure 2.26 shows the number of coded bits for different significance levels for coding configurations (a) and (e) in two wavelet subbands. The numbers are estimated using (2.12) and (2.13). As shown in the figures, at the most significant levels, both schemes can achieve substantial compression, but by compressing the significance map using intra coding it is possible to achieve better coding gain. On the other hand, in the middle significance levels, coding the entire raw bit-plane with DSC can achieve better results. As for the least significant bit-planes, both schemes cannot achieve much compression, as the bits there are equally likely and do not have much correlation with the corresponding SI.

Based on these modeling results, we propose an *adaptive coding scheme*: when coding the most significant bit-planes at the beginning, we partition the magnitude bits into refinement bits and significance maps, and apply DSC and intra coding (zerotree coding) respectively (i.e., configuration (e)). Later, in the middle significance levels, we switch to compress the entire raw bit-planes using DSC (i.e., configuration (a)) (Figure 2.27). We use (2.12) and (2.13) to determine the significance level at which configurations switching should occur. Note that for different subbands, switching could occur at different significance levels. Switching would occur earlier (at a higher significance level) for high decomposition level subbands as intra coding is less efficient there ⁸. Intuitively, in high decomposition level subbands, coefficients would become significant earlier, and zerotree coding would become inefficient as much partitioning would be needed.

2.7 Experimental Results of Adaptive Codec

This section presents the experimental results of the adaptive DSC-based hyperspectral image codec. We use (2.12) and (2.13) to determine the significance level at which configurations switching occurs for each subband. We compare with the non-adaptive scheme in Section 2.3 [67], which uses zerotree coding for significance maps for all significance levels (i.e. configuration (e) only). We use the NASA AVIRIS image data-sets in the experiment [38]. The original image consists of 224 spectral bands, and each spectral band

⁸Recall $p_0(l, i)$ is the probability that significance map bits being zero. We can rewrite (2.16) as $p_0(l, i) = \frac{1-\tau}{1-\tau^2}$, with $\tau = \exp(-\beta_i 2^l)$, hence $0 \leq \tau \leq 1$. It can be shown that $p_0(l, i)$ decreases monotonically with increasing τ , with $p_0(l, i) = 1$ when $\tau = 0$ and $\lim_{\tau \rightarrow 1} p_0(l, i) = 0.5$. According, intra coding would become less efficient for significance maps when τ is large, i.e., when l is small (at the least significance levels) and when β_i is small (when coefficients distributions have large variances, in low-pass subbands and high decomposition level subbands).

consists of 614×512 16-bits pixels. In the experiment, we compress 512×512 pixels in each band. Figures 2.28 and 2.29 show some of the results in compressing images *Cuprite* (radiance data) and *Lunar* (reflectance data). Here $\text{MPSNR} = 10 \log_{10}(65535^2/\text{MSE})$, where MSE is the mean squared error between all the original and reconstructed bands. As shown in the figures, the adaptive coding scheme can provide considerable and consistent improvements in all cases, with up to 4dB gain at some bit-rates.

We also compare the DSC-based systems with several 3D wavelet systems (3D ICER) developed by NASA-JPL [38]. As shown in Figure 2.28, the DSC-based system with adaptive coding is comparable to a simple 3D wavelet system (FY04 3D ICER) in terms of coding efficiency. The simple 3D wavelet system uses the standard dyadic wavelet decomposition and a context-adaptive entropy coding scheme to compress coefficients bits. However, there still is a performance gap when comparing the DSC-based systems to a more recent and sophisticated version of 3D wavelet (Latest 3D ICER). The more recent 3D wavelet developed in NASA-JPL exploits the spatial correlation remained in the correlation noise [38]. This could be one direction to improve the DSC-based systems, which currently use a simple i.i.d. model for correlation noise and ignore the dependency between correlation noise symbols. We also compare the DSC-based systems with 2D SPIHT, and the DSC-based systems can achieve 8dB gains at some bit-rates.

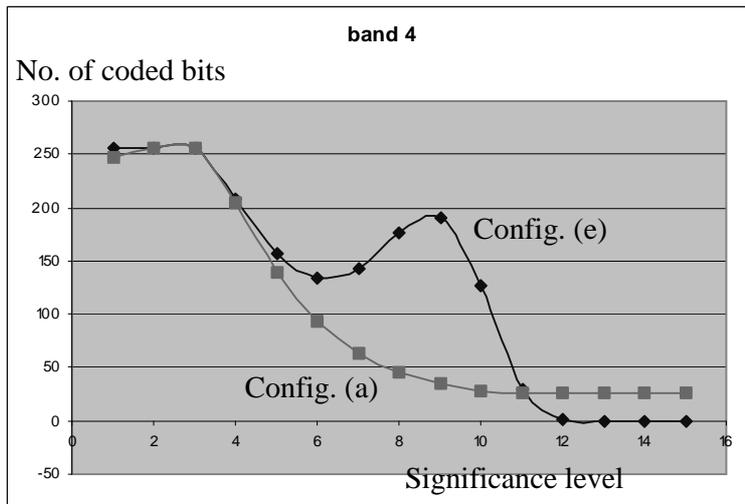
2.8 Conclusions

In this chapter, we have demonstrated a viable approach for compression of hyperspectral imagery. A novel scheme called SW-SPIHT is proposed. Our scheme can facilitate

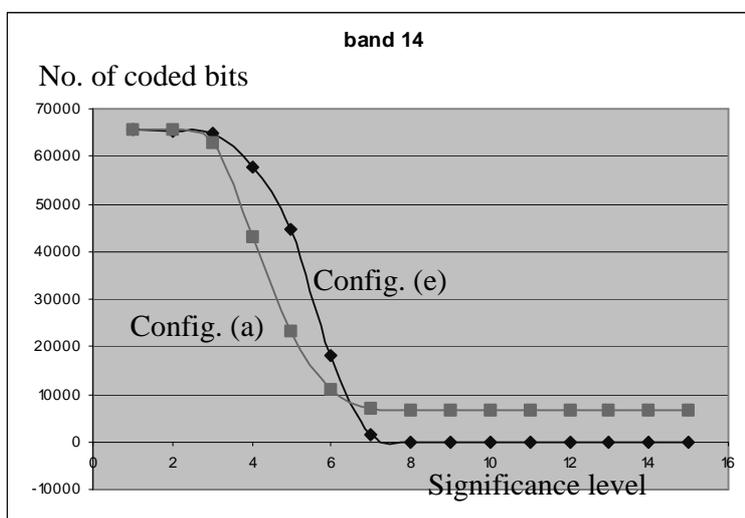
parallel encoding with modest memory requirement. As for coding performance, we have compared our scheme with several existing techniques. Experimental results show that our scheme can achieve competitive coding efficiency. In particular, our scheme is comparable to a simple 3-D wavelet codec developed by NASA-JPL in terms of compression performance. Furthermore, our scheme can preserve spectral signatures and obtain good classification performance.

Acknowledgments

The authors would like to thank Sam Dolinar, Aaron Kiely and Hua Xie of NASA-JPL for helpful discussions and for providing data for comparison. The authors would also like to thank Nazeeh Aranki of NASA-JPL for providing access to the AVIRIS hyperspectral images.



(a) The 4th subband.



(b) The 14th subband.

Figure 2.26: Modeling results.

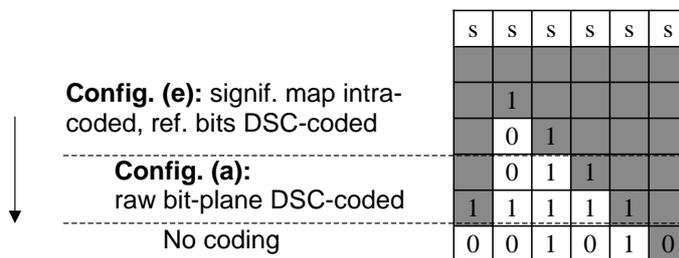


Figure 2.27: Adaptive coding scheme.

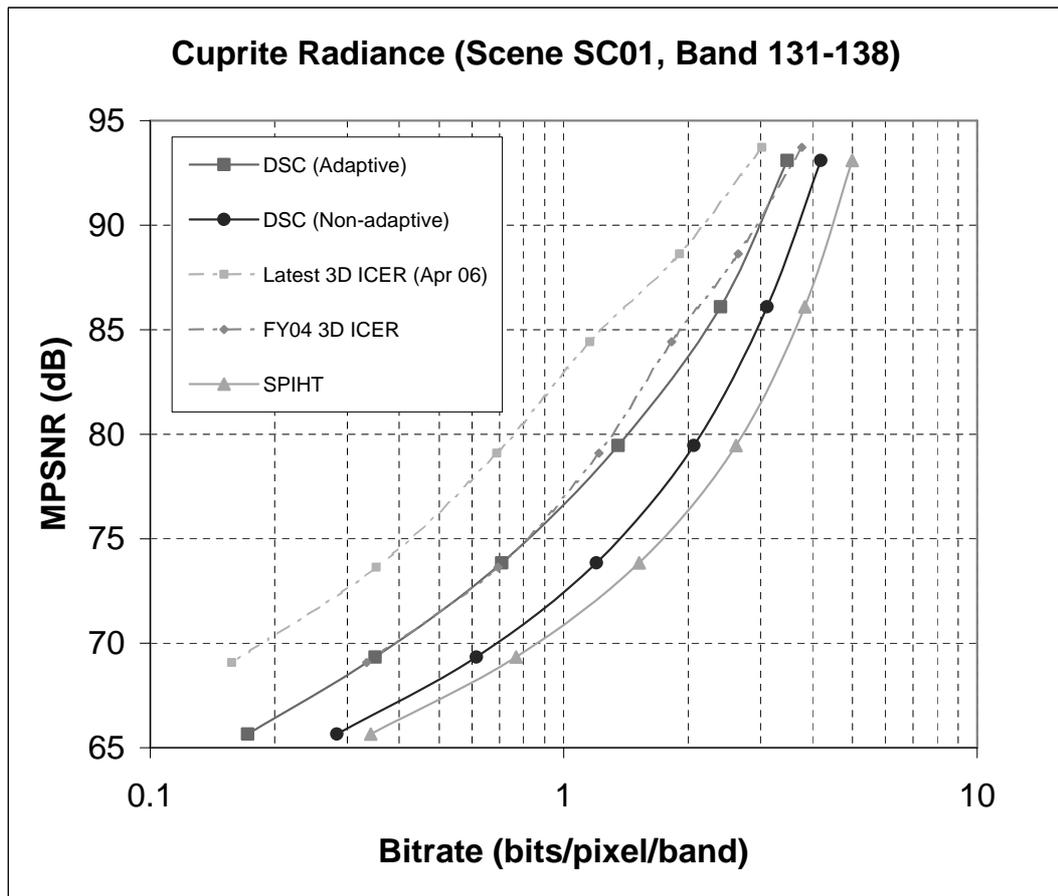


Figure 2.28: Coding performance: Cuprite. Correlation information is estimated by a model-based approach discussed in Chapter 4.

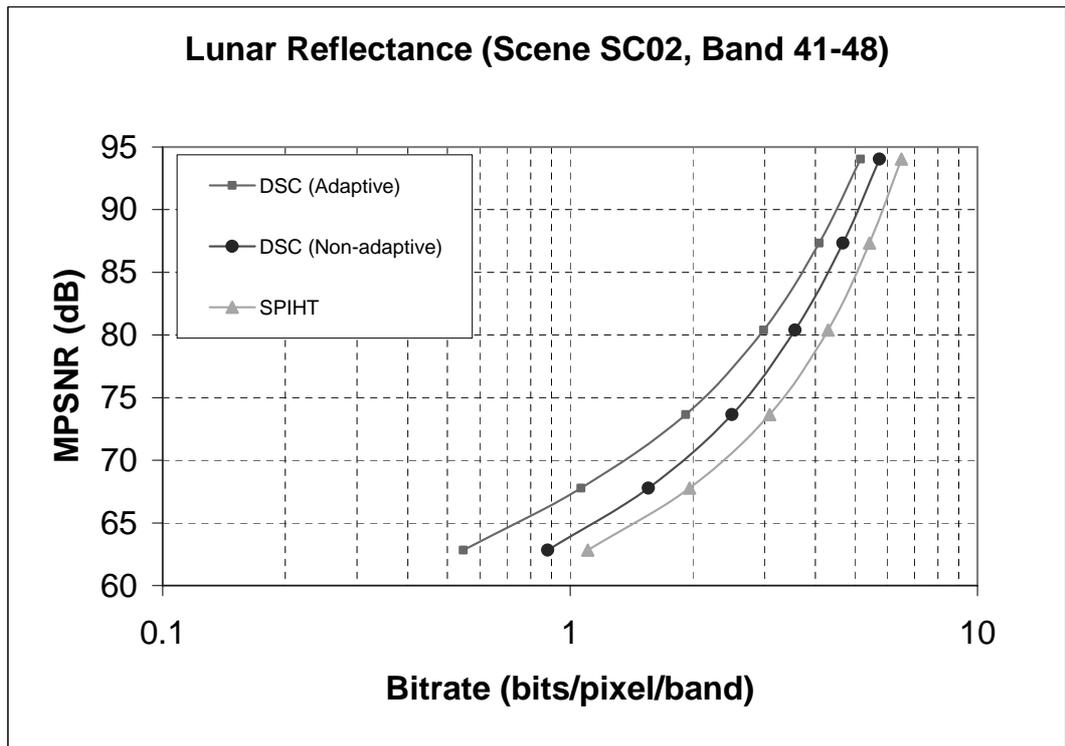


Figure 2.29: Coding performance: Lunar. Correlation information is estimated by a model-based approach discussed in Chapter 4.

Chapter 3

Flexible Video Decoding: A Distributed Source Coding Approach

3.1 Introduction

3.1.1 Motivation

In this chapter we investigate video compression algorithms to support *flexible decoding* for a number of emerging applications [11]. Flexibility in this context means that video frames can be decoded in several different *orders*, while still exploiting redundancy between successively *decoded* frames (e.g., temporal or cross-view redundancy)¹. The decoding order is decided only at the time of decoding, so that a choice among several available decoding paths can be made depending on the users' preferences or the operating conditions. We focus on coding tools to generate a *single* compressed bit-stream that can be decoded in *several* different ways, i.e., we assume it is not possible to request at decoding time (via feedback) coded data matching the chosen decoding order.

¹A trivial approach to enable flexible decoding would be to encode every frame independently, as an Intra frame.

Flexible decoding can be useful for several applications. Notably, it improves the *accessibility* of the compressed data, which is important for several emerging applications and for some novel imagery datasets [53]. For example, some multiview video applications such as free viewpoint TV [35, 70] aim to enable users to play back different views and to switch between different views during playback. In order to facilitate these free viewpoint switchings, it is desirable for the compressed multiview video data to be decodable in several different orders, corresponding to different view switching scenarios [10] (Figure 3.1). As another example, new video applications which support forward and backward frame-by-frame playback can benefit from compression schemes that allow both forward and backward decoding [16] (Figure 3.2).

Moreover, flexible decoding can be used to achieve more *robust* video communications, in applications where some reference frames may be corrupted during transmission. If a compression scheme can support multiple decoding paths the decoder would be able to recover the current frame using any of several possible error-free references (Figure 3.3) [79].

3.1.2 Flexible Decoding: Challenges

State-of-the-art video coding algorithms exploit redundancy between neighboring frames to achieve compression [82]. Flexible decoding makes it difficult to exploit this kind of interframe redundancy because decoders can choose different decoding paths, each leading to a different set of previously decoded frames. Thus at the time of encoding there will be uncertainty about which frames can be used to predict the current frame (as there is no guarantee that those same frames will be available at decoding time).

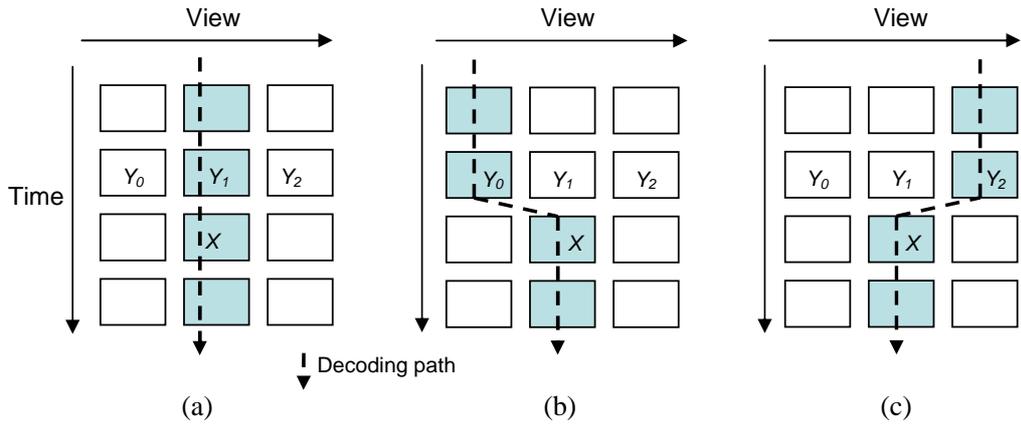


Figure 3.1: Multiview video applications - viewpoint switching may require a compression scheme to support several different decoding orders: (a) users stay in the same view during playback; (b), (c) users switch between adjacent views during playback.

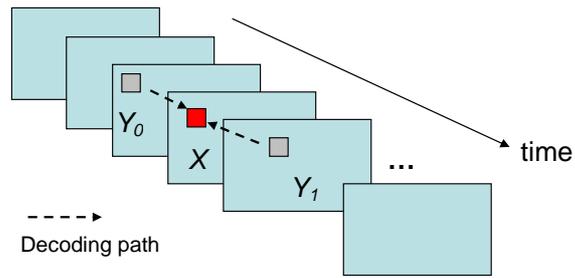


Figure 3.2: Forward and backward frame-by-frame playback.

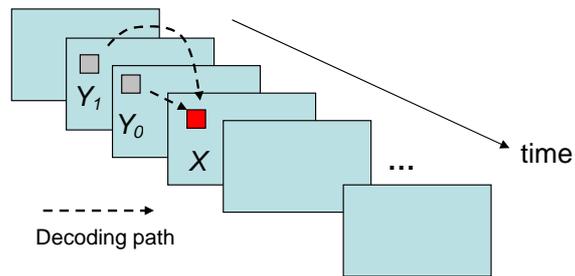


Figure 3.3: Robust video transmission using multiple decoding paths.

For example, in multiview video applications, depending on whether the user continues requesting the same view as in Figure 3.1(a), or switches views as in Figure 3.1(b) or Figure 3.1(c), either the previous reconstructed frame of the same view (Y_1) or that of another view (Y_0 or Y_2) would be available as predictor for decoding the current frame X . However, since it is up to the users to choose among different decoding paths, the encoder would not know exactly which reconstructed frames will be available for decoding X . Similarly, in a forward/backward video playback application, either the “past” or the “future” reconstructed frame will be available at the decoder to serve as the predictor, depending on whether the data is being played back in the forward or backward direction (Figure 3.2). Since users can choose to play back in either direction, the encoder would not know which reconstructed frame will be available at the decoder. Similar scenarios can also arise in low delay video communication, where decoder feedback could be infeasible. In these cases the encoder may not have any information regarding which reference frames have arrived at the decoder error-free and would be available for decoding the current frame (Figure 3.3). In short, flexible decoding, while desirable, results in uncertainty on the predictor status at decoder.

Figure 3.4 depicts the general formulation of the flexible decoding problem. When compressing an input source X (the current video frame), the encoder has access to a number of correlated sources Y_0, Y_1, \dots, Y_{N-1} (previously decoded video frames) to serve as predictors for encoding X . Here each Y_k is associated with a possible decoding path. However, of these predictor candidates, only *one* will be available at the decoder depending on the decoding path it takes. Crucially, since the encoder does not have any information regarding the chosen decoding path, it *does not* know which Y_k will be used

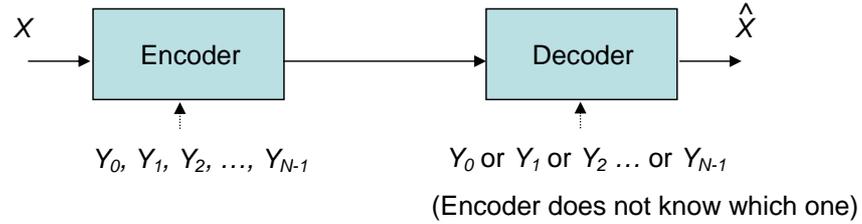


Figure 3.4: Problem formulation for flexible decoding. Either one of the candidate predictors Y_0, Y_1, \dots, Y_{N-1} will be present at the decoder, but encoder does not know which one.

at the decoder. Our goal is to investigate coding algorithms such that the encoder can operate under this kind of uncertainty about predictor status at the decoder.

In order to support flexible decoding within a conventional closed-loop prediction (CLP) framework, e.g., motion-compensated predictive (MCP) video coding systems such as MPEG or H.26X, the encoder may send all the possible prediction residues $\{Z_i; i = 0 \text{ to } N - 1\}$ to the decoder, where $Z_i = X - Y_i$ (following the notations in Figure 3.4), so that X can be recovered no matter which Y_i is available at the decoder. Each Z_i would correspond to a P-frame in these video coding standards. Note that it is indeed necessary for the encoder to communicate all the N possible prediction residues to the decoder. This is because, in CLP, a prediction residue would be “tied” to a specific predictor. For example, if Y_k is the available predictor at the decoder, then we can only use Z_k during the decoding process to recover X without causing significant mismatch. Therefore, in the cases of predictor uncertainty, the encoder would need to send multiple prediction residues. Thus there are two potential issues with the CLP approach. First, coding performance is degraded because multiple prediction residues are included in the bitstream. Specifically, the overhead to support flexible decoding tends to increase with the number of candidate predictors (or the number of possible decoding paths). Second,

this approach may cause drifting. This is because, in practical video compression systems, the encoder would send the quantized versions of Z_i , \hat{Z}_i , to the decoder. Therefore, the reconstructed sources $\hat{X}_i = \hat{Z}_i + Y_i$ would be slightly different when different Y_i are used as predictors. Drifting may occur when \hat{X}_i is used as reference for decoding future frames.

The H.264 video compression standard has defined SP- and SI-frames to support functionalities such as random access or error recovery that were originally supported by I-frames [37]. Essentially SP-frames follow the CLP coding approach we just discussed, but with modifications such that \hat{X}_i can be identically reconstructed from different Y_i 's using its corresponding Z_i (here Z_i corresponds to a primary or secondary SP-frame). This is achieved by using a different prediction loop from that in conventional P-frames (e.g., SP-frames compute the prediction residue w.r.t. the quantized reconstruction in the transform domain, whereas P-frames would compute it w.r.t. the original image in the pixel domain [37]). However, this causes some penalty in coding performance, and the compression efficiency of SP-frames is in general worse than that of P-frames [37]. To support flexible decoding, different SP-frames bits (each corresponding to a different Y_i) need to be generated and sent to the decoder, similar to CLP coding, and therefore, H.264 SP-frames would incur a comparable amount of overhead as that in CLP coding. It should be noted that most H.264 SP-frame applications assume the availability of feedback from the decoder (e.g., Zhou et al. [87]), so that the encoder does know which predictor is available at the decoder and transmits only one of the Z_i . In short, H.264 SP-frames could be inefficient to support flexible decoding when there is no feedback.

3.1.3 Our Contributions and Related Work

In this chapter, we propose to apply distributed source coding (DSC) [28, 56, 63, 84, 85] to address the flexible decoding problem, where the encoder has access to all predictors, Y_k , which will play the role of side information (SI) at the decoder, but there is uncertainty as to which one will be used for decoding. One of the main challenges for DSC-based applications has proven to be achieving competitive compression efficiency [28]. To address this challenge, our proposed algorithm incorporates novel macroblock modes and significance coding into the DSC framework. This, along with careful exploitation of correlation statistics, allows us to achieve significant performance improvements. Using multiview video and forward/backward video playback as examples, we demonstrate that the proposed algorithm can outperform, in terms of coding efficiency, techniques based on CLP coding such as those as discussed above. Moreover, the proposed algorithm incurs only a small amount of drifting. In particular, DSC-coded macroblocks lead to the same reconstruction no matter which predictor candidate Y_k is used.

DSC has been studied extensively for enabling low-complexity video encoding, e.g., Puri and Ramchandran [59], Aaron et al. [4]. However, there are significant differences between low-complexity encoding and flexible decoding, as summarized in Table 3.1, which will lead us to a different solution. DSC has also been proposed to provide random access in compression of image-based rendering data/light fields in Jagmohan et al. [36] and Aaron et al. [2], and of multiview video data in Guo et al. [32]. This prior work, however, assumes that the encoder has knowledge of predictor status at decoder, notably through using feedback, while in our case the encoder needs to operate with unknown predictor

status. Recent work by Wang et al. [79] has proposed a DSC-based approach to address the problem of robust video transmission by allowing a video block to be decoded using more than one predictor blocks. While the general philosophy is similar to ours, different assumptions are made. In particular, this work assumes the encoder knows the probability that each predictor will be used, as determined by the packet erasure probability (whereas we assume all predictors are equally-likely to be used). This information is exploited to reduce the coding rate. In addition, the specific tools used are different from those proposed here. Recent work by Naman and Taubman [52] has proposed to enhance decoding flexibility and accessibility using intra coding and conditional replenishment. Some information theoretic aspect of flexible decoding has also been studied independently by the recent work of Draper and Martinian [22], which we will briefly discuss. However, there is no practical coding algorithm proposed in this work. Reverse playback of video specifically focusing on MPEG coding algorithms was discussed in Wee and Vasudev [81], Lin et al. [42] and Fu et al. [25]. Our previous work Cheung et al. [16] has also proposed to apply DSC to enable forward/backward video playback. The proposed algorithm in the present chapter is, however, considerably different and significantly more efficient. Among the key improvements are the introduction of macroblock modes and significance coding, a different approach to exploit the correlation between source and side-information, a different way to partition the input symbols and estimate the source bit's conditional probability, and a minimum MSE dequantization.

This chapter is organized as follows. In Section 3.2 we discuss how DSC can address flexible decoding. A comparison of theoretically achievable performances is provided in Section 3.3. In Section 3.4 we present the proposed compression algorithm. Section 3.5

Table 3.1: Comparison of DSC-based low-complexity encoding and flexible decoding.

	DSC-based low complexity video encoding [4, 59]	DSC-based flexible video decoding
Key objective	Low complexity video encoding for mobile video, video sensors, etc.	Generate a single bitstream to support multiple decoding paths for forward and backward video playback, multi-view video, video transmission, etc.
Encoding complexity	Most target applications require low complexity, real-time encoding.	Not primary issue. Most target applications may use offline encoding.
Encoder access to the side information	SI not accessible by encoder due to complexity constraint.	Encoder has access to all the SI candidates. However, the exact one to be used at decoder is unknown to encoder.

discusses briefly some application scenarios. Section 3.6 presents the experimental results and Section 3.7 concludes the work.

3.2 Flexible Decoding Based on DSC: Intuition

In conventional CLP coding, the encoder computes a prediction residual $Z = X - Y$, between source X and predictor Y , and communicates Z to the decoder (Figure 3.5(a)). DSC approaches the same compression problem taking a “virtual communication channel” perspective [28, 85]. Specifically, X is viewed as an input to a channel with *correlation noise* Z , and Y as the output of the channel (Figure 3.5(b)). Therefore, to recover X from Y , encoder would send *parity information* to the decoder. That is, in DSC, the encoder would communicate X using parity information. Significantly, the parity information is independent of a specific Y being observed: the parity information is computed entirely

from X taking into account the statistics of Z ². In particular, what matters in the DSC approach is, analogous to data communication, the *amount* of parity information corresponding to the statistics of Z . Thus the decoder will be able to recover X as long as a sufficient amount of parity information has been communicated. In short, in DSC, the information communicated from encoder to decoder is independent of a specific Y , in contrast to CLP, where the encoder would communicate to the decoder the prediction residue, which is tied to a specific Y .

To understand how DSC can tackle flexible decoding with N predictor candidates, consider N virtual channels, each corresponding to a predictor candidate Y_i (Figure 3.5(c)). Each channel is characterized by the correlation noise $Z_i = X - Y_i$. In order to recover X from any of these channels, the encoder would need to send an amount of parity sufficient for *all* the channels. In particular, the encoder would need to transmit enough parity information to allow decoding of the *worst-case* Z_i . Doing so, X can be recovered no matter which Y_i is available at the decoder. Note that the encoder only needs to know the statistics of all the Z_i to determine the amount of parity information, and this is feasible since X and all Y_i are accessible at encoder in our problem formulation. In particular, the encoder does not need to know which Y_i is actually present at decoder. Since parity information is independent of a specific Y_i , the same parity information generated based on the worst case Z_i can be used to communicate X no matter which Y_i is available at the decoder.

²As will be discussed in Section 3.4, we encode the bit-plane representation of X using DSC, and parity information is computed by XOR-ing a subset of bits in the bit-planes of X .

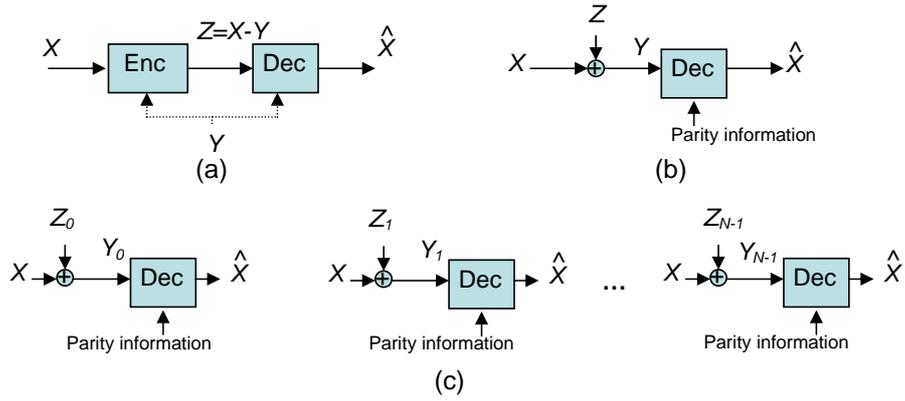


Figure 3.5: Compression of input source X : (a) CLP coding; (b) DSC from the virtual channel perspective; (c) DSC approach to the flexible decoding problem.

3.3 Theoretical Performance

In this section we compare the theoretical performance of CLP and DSC in a flexible decoding scenario. We consider the compression of a discrete i.i.d. scalar source X under the scenario depicted in Figure 3.4. The predictor candidates Y_i , $i = 0$ to $N - 1$ are discrete i.i.d. scalar sources such that $X = Y_i + Z_i$, $Y_i \perp Z_i$, where Z_i are the discrete i.i.d. scalar prediction residues. As previously discussed, in CLP the overhead to address flexible decoding increases with the number of predictors N , while in DSC the overhead depends mainly on the worst-case correlation noise. Specifically, in the CLP approach, all the residues $\{Z_i; i = 0 \text{ to } N - 1\}$ would have to be sent to the decoder, which theoretically would require an information rate

$$R_{CLP} = \sum_{i=0}^{N-1} H(Z_i). \quad (3.1)$$

On the other hand, in the DSC approach, the information rate required to communicate X with Y_i at the decoder is $H(X|Y_i)$, and using $X = Y_i + Z_i$ and $Y_i \perp Z_i$, we have

$H(X|Y_i) = H(Z_i)$. Under the scenario of side-information uncertainty, we would need to communicate X at a rate³

$$R_{DSC} = \max_i H(X|Y_i) = \max_i H(Z_i). \quad (3.2)$$

It is clear that the encoder needs to communicate at least $\max_i H(X|Y_i)$ bits so that X can be recovered for whichever Y_i available at decoder. To show that $\max_i H(X|Y_i)$ is indeed achievable, we use the *source networks* approach proposed by Csiszar and Korner [20, 21]. A source network is a graphical abstraction of a multiterminal source coding problem involving information sources, encoders and destinations located at its vertices. Each encoder operates on the input messages from the sources connected to it, and the resulting codeword is made available to all destinations connected to the encoder through noiseless communication channels. Each destination must be able to reproduce accurately the messages from certain specified sources based on the received codewords. In particular, the source networks in Csiszar and Korner [21] focus on (i) discrete memoryless sources, (ii) graphs in which no edge joins two sources or two encoders or two destinations, and (iii) graphs in which the destinations are required to reproduce the messages of the specified sources with small probability of error (i.e., lossless data compression). Figure 3.6(a) shows an example of source network representing the Slepian-Wolf problem. For certain subclass of source network, Csiszar and Korner [21] derived the exponential error bounds which are tight in a neighborhood of the boundary of the achievable rate region. In addition, these bounds were shown to be *universally attainable*, i.e., they are

³To be more precise, R_{CLP} and R_{DSC} are the best (minimum) achievable rates.

attainable by encoders and decoders not depending on the source statistics. Subsequently, Csiszar [20] further showed that the error exponents can be attained universally using linear codes.

To model the predictor uncertainty in flexible decoding, we use N source networks each corresponding to a different predictor candidate as depicted in Figure 3.6(b). Following from the result in Csiszar and Korner [20,21] that the Slepian-Wolf's achievable rate region is universally attainable, the *same* codes can be used to communicate X in any of these source networks at an achievable rate $H(X|Y_i)$. Therefore, at a rate of $\max_i H(X|Y_i)$, the codes can be used to communicate X regardless of which Y_i is available at the decoder. From (3.1) and (3.2), we have $R_{CLP} \geq R_{DSC}$. Therefore, the DSC approach can potentially achieve better coding performance. Figures 3.7 and 3.8 depict for some empirical data $H(X)$ (i.e., intra coding), $\sum_i H(Z_i)$ (i.e., R_{CLP}) and $\max_i H(Z_i)$ (i.e., R_{DSC}) in different applications and show how these quantities could vary with the number of predictors.

Note that some theoretical aspect of flexible video decoding was also studied by Draper and Martinian [22]. In particular, (3.2) was independently proved in [22], using a different approach based on some extension of the random binning arguments. [22] also discussed the improved achievable error exponents compared to those of conventional Slepian-Wolf problem.

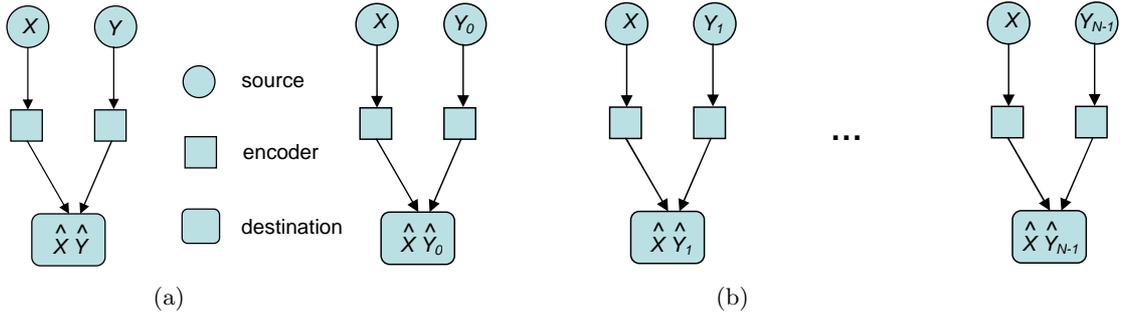


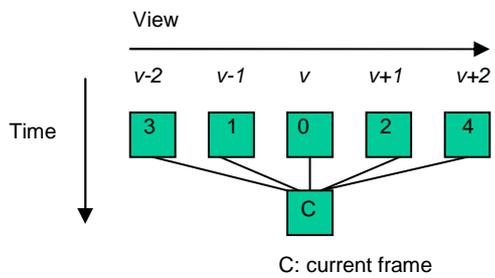
Figure 3.6: Source networks: (a) Source network of Slepian-Wolf [21, 63]. Csiszar and Korner [20, 21] suggest an achievable rate $H(X|Y)$ for communicating X is universally attainable. (b) Source networks of flexible decoding with predictor uncertainty. The same universal codes can be used to communicate X in any of these networks at a rate $H(X|Y_i)$.

3.4 Proposed Algorithms

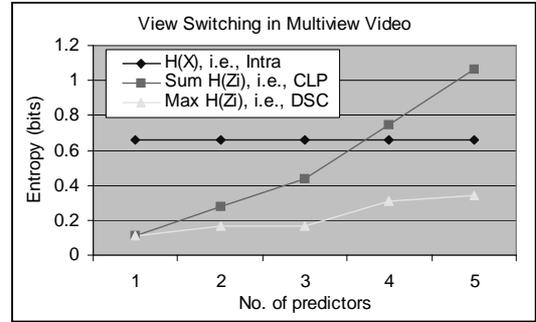
Figure 3.9 depicts the proposed video encoding algorithms to address flexible decoding based on DSC [11]. These are described next.

3.4.1 Motion Estimation and Macroblock Classification

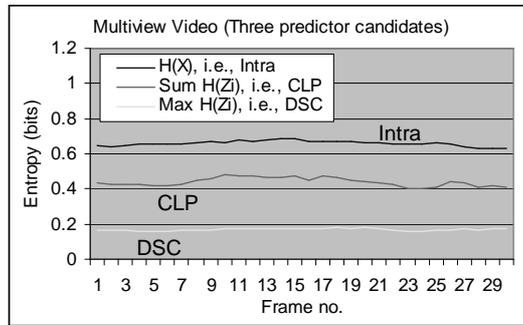
Each macroblock (MB) M in the current frame first undergoes standard motion estimation (and disparity estimation in the case of multiview video application) w.r.t. each candidate reference frame f_i , and the corresponding motion information (one per reference frame, f_i) is included in the bitstream, i.e., the encoder sends N motion vectors to the decoder. Denote A_i the best motion-compensated predictor for M obtained in f_i . If the difference between M and A_i is sufficiently small, M may be classified to be in skip mode w.r.t. f_i (Figure 3.9). In that case, since the encoder can skip some prediction residues and the encoder does not need to communicate all N residues, the overhead of



(a)



(b)



(c)

Figure 3.7: Theoretical performances of intra coding, CLP and DSC in a flexible decoding scenario: Multiview video coding as in Figure 3.1. (a) Previously reconstructed frames of neighboring views are used as predictor candidates following the depicted order; (b) Entropy of the quantized DCT coefficients (as an estimate of the encoding rate) vs. number of predictor candidates. The results are the average of 30 frames using Akko&Kayo view 28-th. (c) Entropy of each frame in the case of three predictor candidates. Note that we assume the sources (i.e., quantized DCT coefficients of images, X , or that of residues, Z_i) consist of independent elements and estimate the coding performances by $H(X)$ and $H(Z_i)$, whereas in practice there could be some correlation exist between the elements and that could be exploited to reduce coding rates.

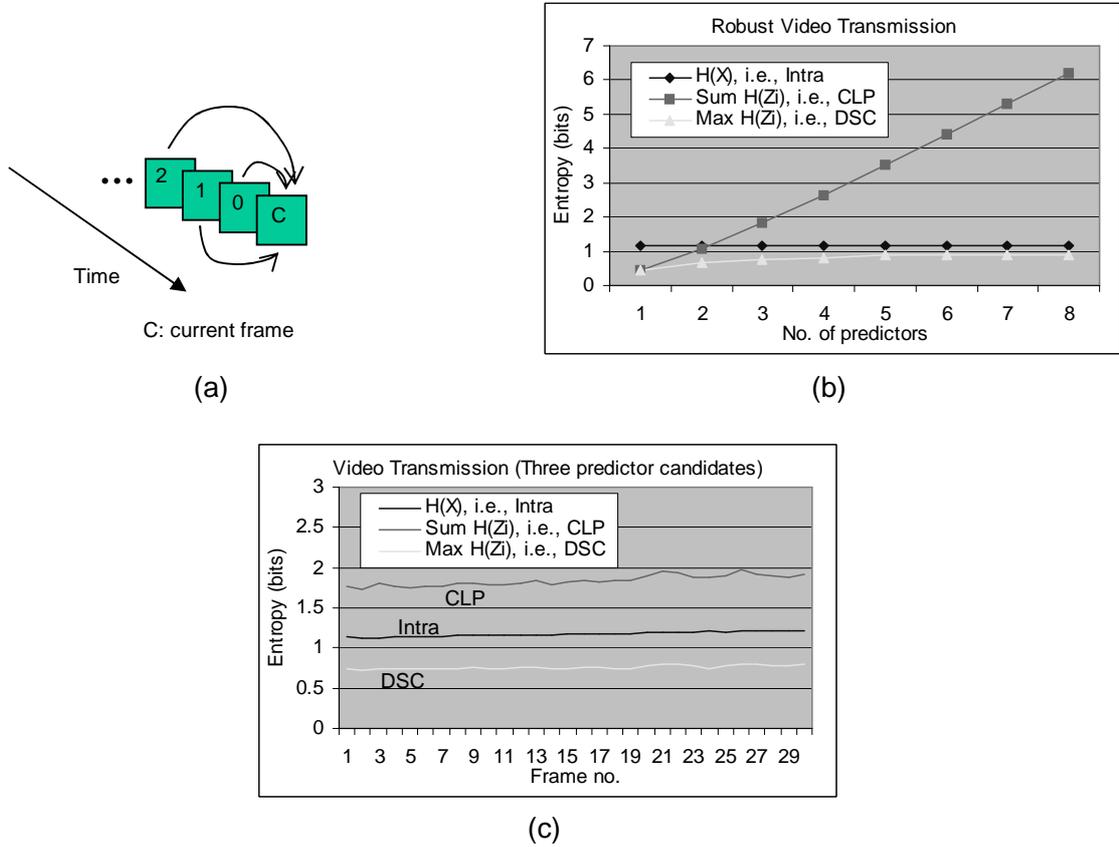


Figure 3.8: Theoretical performances of intra coding, CLP and DSC in a flexible decoding scenario: Robust video transmission as in Figure 3.3. (a) Past reconstructed frames are used as predictor candidates following the depicted order; (b) Entropy of the quantized DCT coefficients (as an estimate of the encoding rate) vs. number of predictor candidates. The results are the average of 30 frames using Coastguard. (c) Entropy of each frame in the case of three predictor candidates. Note that we assume the sources (i.e., quantized DCT coefficients of images, X , or that of residues, Z_i) consist of independent elements and estimate the coding performances by $H(X)$ and $H(Z_i)$, whereas in practice there could be some correlation exist between the elements and that could be exploited to reduce coding rates.

including multiple prediction residues using CLP could be small. Specifically, in our current implementation, each macroblock M can be in either skip mode or a non-skip mode. A macroblock will be encoded using the skip mode when, out of all of the N residue blocks between M and A_i , $0 \leq i \leq N - 1$, at least one of the residue blocks is a zero block after quantization (i.e., all the quantized transform coefficients in the block are zero). In skip mode M is encoded using conventional CLP coding (similar to standard H.26X algorithms) w.r.t. the candidate reference frames which do not have skipping. However, majority of the macroblocks will be classified into the non-skip mode and be encoded using DSC following the steps discussed in the next section.

Note that choosing between CLP and DSC for a given macroblock can be achieved using rate-distortion (RD) based mode selection (as in H.264): The RD costs of CLP and DSC are computed and the one achieving the minimum RD cost is selected. Such RD optimized mode decision algorithm can achieve a better coding performance, at the expense of requiring higher encoding complexity. In our comparison with H.263 (Section 3.6) we did not use this RD optimized mode decision. As will be discussed, we implemented our proposed algorithms mainly based on H.263 coding tools.

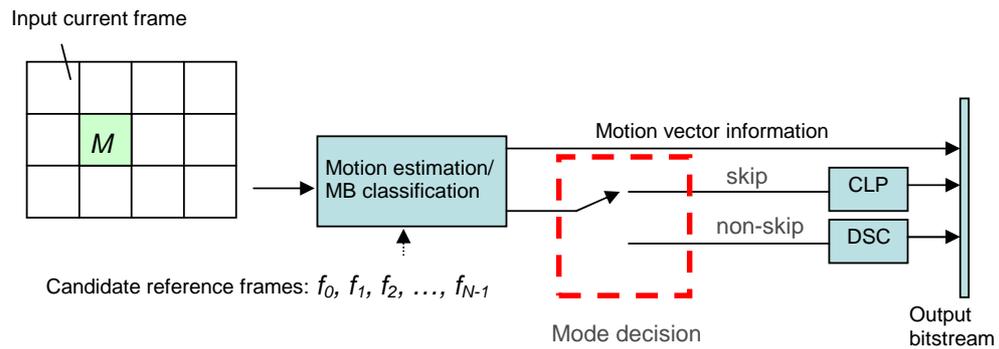


Figure 3.9: Proposed encoding algorithm to encode an macroblock M of the current frame.

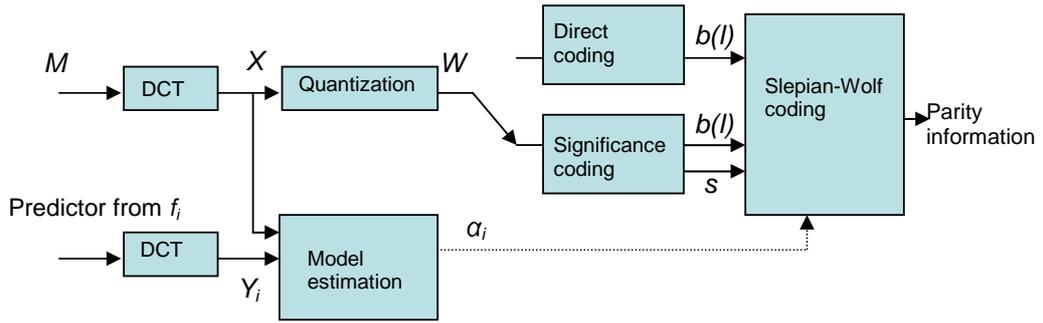


Figure 3.10: Encoding macroblock M using DSC.

3.4.2 Direct Coefficient Coding (DCC)

For those macroblocks M to be encoded with DSC, we first apply standard 8×8 DCT to the pixel data to obtain the vector of transform coefficients. Denote X the DCT coefficient. We then quantize X to obtain the quantization index W (Figure 3.10). This is similar to intra-frame coding in standard H.26X algorithms. Denote Y_i the DCT coefficient in A_i corresponding to W (recall A_i is the best motion-compensated predictor from each f_i). We compress W by exploiting its correlation with the worst case Y_i , so that it can be recovered with any Y_i that may be present at the decoder. Specifically, based on a correlation model between W and Y_i (to be discussed in Section 3.4.5), the encoder can estimate the coding rates needed to communicate W when Y_i is available at the decoder. Then the encoder communicates W by sending an amount of parity information equal to the maximum of these estimated coding rates. Since both W and Y_i are available at the encoder in our problem, the correlation model can be readily estimated.

The quantized values of the K lowest frequency DCT coefficients (along a zig-zag scan order) are encoded with a *direct* coefficient coding (DCC), and for the rest we use

a *significant* coefficient coding (SCC). In DCC, we form the k -th frequency coefficient vector by grouping together the k -th ($0 \leq k \leq K - 1$) frequency quantized coefficients, W_k , from all the 8×8 blocks in a frame (except those in skip modes). Then each of these vectors is converted into a raw bit-plane representation, and the bit-planes are passed to a Slepian-Wolf (SW) coder, where inter-frame correlation is exploited to compress the bit-planes losslessly. Note that DCC would lead to L_k bitplanes for the k -th frequency coefficients vector, where $L_k = \lceil \log_2(\max |W_k| + 1) \rceil$.

3.4.3 Significant Coefficient Coding (SCC)

The quantized values of the k -th highest frequency coefficients, $k \geq K$, are encoded using SCC. Specifically, we first use a *significance bit* s to signal if the quantized value of a coefficient is zero ($s = 0$) or not ($s = 1$), so that distributed coding is used to communicate only the value of non-zero coefficients. The significance bits of all the k -th frequency coefficients in the frame (one for each 8×8 block, except those in skip modes) are grouped together to form a significance bit-plane to be compressed by the SW coder. On the other hand, the non-zero coefficients are grouped together to form coefficient vectors where all the DCT frequencies are combined, as we found that the correlation statistics of non-zero coefficients are similar at different frequencies.

SCC is introduced as an alternative to DCC to reduce the number of source bits to be handled in SW coding. Specifically, assume DCC leads to L_k bitplanes for the k -th frequency coefficient vector. Therefore, each k -th frequency coefficient contributes L_k source bits in DCC, regardless of whether the coefficient is zero or not. On the other hand, with SCC, a zero coefficient contributes one source bit (significance bit), while a

non-zero coefficient contributes approximately $1 + L_k$ bits. If p_k is the probability that the k -th frequency coefficient will be zero, then the expected number of source bits using SCC is

$$1 \times p_k + (1 + L_k) \times (1 - p_k), \quad (3.3)$$

and SCC can lead to rate savings (compared with DCC) if the expected number of bits using SCC, i.e., (3.3), is less than that of DCC, i.e., L_k , or equivalently if

$$p_k > \frac{1}{L_k} \quad (3.4)$$

holds. Therefore, SCC can achieve rate savings when coefficients are likely to be zero. In the experiment, we use $K = 3$ (where SCC starts) determined using (3.4) and some statistics of the video sequences.

3.4.4 Bit-plane Compression

Bit-planes extracted from the K coefficient vectors produced in DCC along with those produced in SCC are compressed by a SW coder, starting from the most significant bit-planes. Denote a bit in the bit-plane at l -th level of significance by a binary r.v. $b(l)$, where $l = 0$ corresponding to the least significant level. That is, $b(l)$ is the l -th significant bit in the quantization index W . A binary r.v. $b(l)$ is to be compressed using Y_i and decoded bits $b(l + 1), b(l + 2), \dots$ as side information. Specifically, this is performed by a low density parity check (LDPC) based SW encoder, which computes the *syndrome* bits from the original bit-planes and sends them to the decoder [43].

3.4.5 Model and Conditional Probability Estimation

SW decoding needs the conditional probability $p(b(l)|Y_i, b(l+1), b(l+2), \dots)$ estimated from SI to aid recovering $b(l)$. The probability can be estimated as follows. The encoder estimates the conditional p.d.f. $f_{X|Y_i}(x|y_i)$ for each coefficient vector and for each candidate predictor. Assuming a system model $X = Y_i + Z_i$, and under the assumption of independence of Y_i and Z_i , we have

$$f_{X|Y_i}(x|y_i) = f_{Z_i}(x - y_i) \quad (3.5)$$

We assume Z_i is Laplacian distributed, i.e., $f_{Z_i}(z_i) = \frac{1}{2}\alpha_i e^{-\alpha_i|z_i|}$, and estimate the model parameters α_i at the encoder using maximum likelihood estimation (MLE) and send them to the decoder. Note that in the flexible decoding problem, the encoder can access all the candidate SIs. Therefore, the model parameters can be readily estimated. This is not the case in typical DSC applications, where there are constraints on accessing side-information at the encoder making model estimation a non-trivial problem [15].

Given all the model parameters α_i , the decoder can estimate the conditional probability for any particular Y_i available at decoder using the following procedure (Figure 3.11). Denote \tilde{W} the numerical value of the concatenation of the sequence of the decoded bits $b(l+1), b(l+2), \dots$, i.e., $\tilde{W} = b(l+1) \times 2^0 + b(l+2) \times 2^1 + \dots$. Given the decoded bits, the quantization index W can range only from $\tilde{W} \times 2^{l+1}$ to $\tilde{W} \times 2^{l+1} + 2^{l+1} - 1$. When

$W \in [W_r, W_s]$, $b(l) = 0$, and when $W \in [W_t, W_u]$, $b(l) = 1$, where W_r, W_s, W_t, W_u are given by (in the cases when $\tilde{W} \geq 0$):

$$\begin{aligned}
W_r &= \tilde{W} \times 2^{l+1}; \\
W_s &= \tilde{W} \times 2^{l+1} + 2^l - 1; \\
W_t &= \tilde{W} \times 2^{l+1} + 2^l; \\
W_u &= \tilde{W} \times 2^{l+1} + 2^{l+1} - 1.
\end{aligned} \tag{3.6}$$

Equations for $\tilde{W} < 0$ are similar. Therefore, the decoder can estimate the probabilities that $b(l)$ will be zero and one by integrating $f_{X|Y_i}(x|y_i)$ over the intervals $[X_r, X_s]$ and $[X_t, X_u]$ respectively, where $[X_r, X_s]$ is the inverse quantization mapping of $[W_r, W_s]$, and $[X_t, X_u]$ is that of $[W_t, W_u]$.

Note that each Y_i exhibits different levels of correlation with respect to $b(l)$. Therefore, the l -th significant bitplane comprised of bit $b(l)$ and extracted from the current video frame would require a different number of syndrome bits to be recovered, when a different candidate reference frame f_i is available at the decoder. Denote R_i this number of syndrome bits. To ensure that the l -th significant bitplane can be recovered with any of the candidate decoder reference f_i , the encoder could send $R = \max R_i$ syndrome bits to the decoder. By doing so, each bit-plane and hence bit $b(l)$ can be exactly recovered no matter which candidate reference is available at the decoder. Therefore, W can be losslessly recovered and X reconstructed to the same value when any of the Y_i is used as predictor. This eliminates drifting in DSC-coded macroblocks.

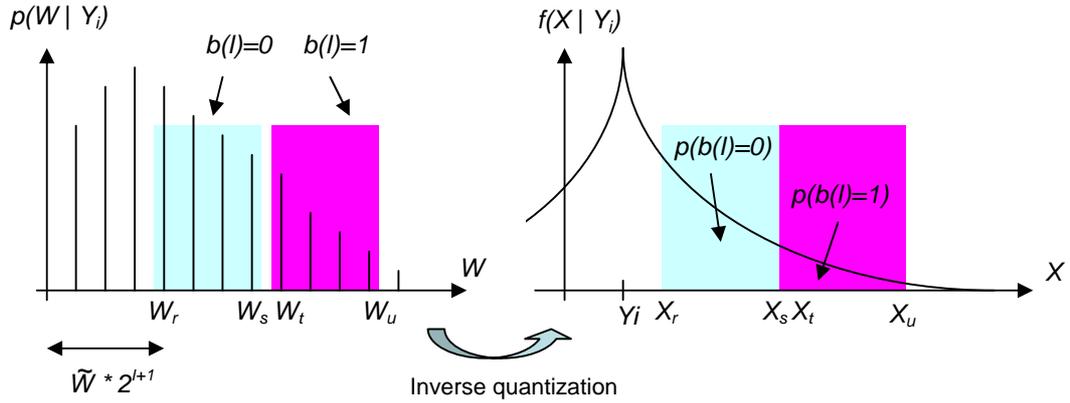


Figure 3.11: Estimate the conditional probability $p(b(l)|Y_i, b(l+1), b(l+2), \dots)$.

3.5 Application Scenarios

While the main objective of the present chapter is to propose coding techniques to facilitate flexible decoding, we will briefly discuss in this section some application scenarios that can benefit from the coding structures enabled by the proposed tools. One of these scenarios could be storage type applications, where the video data is pre-encoded, and the entire bitstream is made available to users through some storage media, e.g., DVD. In these applications, the proposed coding techniques can lead to new coding structures that would require less decoding complexity, i.e., a smaller number of computations and less amount of memory buffering. As an example, for multiview video applications, where conventionally an individual view could be encoded independently (i.e., simulcast) or multiple views could be compressed jointly, viewpoint switching would require reconstruction of several additional video frames (predictor frames) which the users did not request (Figure 3.12). In other words, the decoder would have to decode some extra frames that would not be displayed. The proposed techniques can, on the other hand, reduce this extra decoding overhead, since they could lead to coding structures where the

current frame can be recovered using any one of the several predictor candidate frames, and no additional processing would be needed as long as some of these candidates were previously requested by the users and hence are already decoded.

Another application scenario that can benefit from the proposed coding techniques could be client-server video streaming applications, where the server would send to the client (decoder) only part of the bitstream needed for decoding the video frames requested by users. In this case, the proposed techniques can reduce, in addition to decoding complexity, the amount of information sent to the client. This is possible because some of the predictor candidates may have been previously requested by the users and have already been communicated to the client. Therefore, the current frame can be decoded without requesting extra information. Note that in client-server applications a DSC based approach to flexible decoding could lead to a smaller total amount of pre-encoded data stored in the server, while a CLP based approach may result in less data being transmitted to the client, since in this case the client could inform the server about the predictor status and only the prediction residue matching the available predictor would need to be communicated to the client.

3.6 Experimental Results and Discussion

3.6.1 Viewpoint Switching in Multiview Video Coding

This section presents the experimental results. We first discuss our experiments on multiview video coding (MVC). Here we generate compressed multiview bit-streams that allow

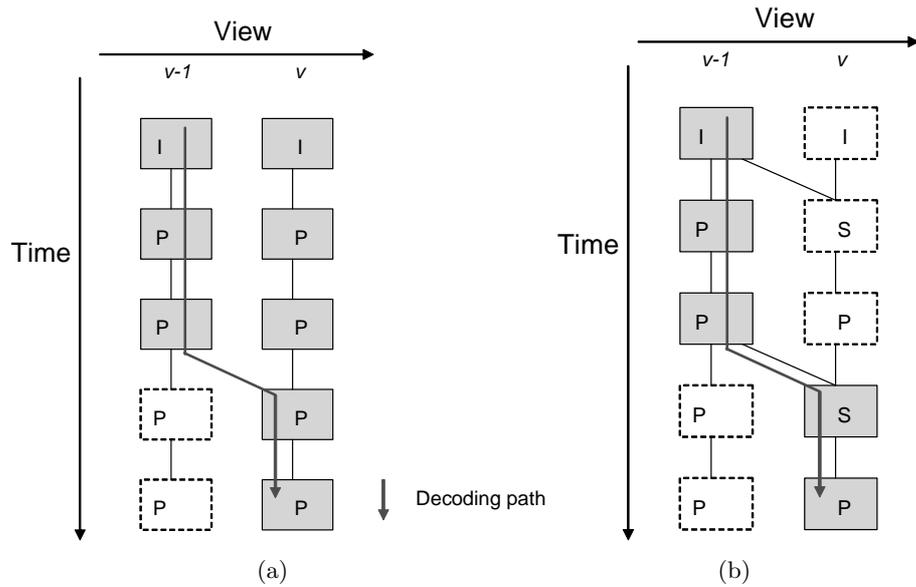


Figure 3.12: Different coding structures in multiview video application. Shaded video frames are those that needed to be decoded to enable the depicted decoding path. (a) Simulcast; (b) New coding structure enabled by the proposed tools, where “S” denotes those video frames that can be decoded using any one of the predictor candidates. Note that in simulcast, decoder needs to decode some extra video frames that users did not request, e.g., the first three frames in the v -th view in the depicted example. In client-server applications, bitstream corresponding to these extra frames is also needed to be sent to the client.

switching from the adjacent views as in Figure 3.1. Therefore, there are three predictor candidates. We compare the coding performance using the following algorithms to generate the bit-stream: (i) intra coding using H.263 I-frames; (ii) CLP approach with each of the three residues encoded using H.263 P-frames; (iii) proposed DSC-based algorithm, with H.263 half-pel motion estimation and quantization. Since we implement all the schemes using the same (H.263) coding tools (e.g., half-pixel accuracy motion estimation) the comparison is fair. We compare the schemes using MVC sequences Akko&Kayo and Ballroom, which are in 320×240 and encoded at 30fps and 25fps respectively. Figures 3.13 and 3.14 show the comparison results. As shown in the figures, the proposed algorithm outperforms CLP and intra coding, with about 1dB gain in the medium/high picture quality range (33-36dB). We also compare the approaches in terms of drifting by simulating a scenario where viewpoint switching from the $(v - 1)$ -th view to the v -th view occurs at frame number 2. Figure 3.15 compares the PSNR of the reconstructed frames within the GOP with that of the non-switching case, where the v -th view is being played back throughout the GOP. As shown in the figure, while CLP (using P-frame) may cause considerable amount of drifting, the proposed algorithm is almost drift-free, since the quantized coefficients in DSC coded macroblock are identically reconstructed.

We also evaluate how the coding performance of the proposed system scales with the number of predictor candidates. In this experiment, the temporally and spatially adjacent reconstructed frames are used as predictor candidates following the order depicted in Figure 3.16. As shown in Figure 3.17, the bit-rate of the DSC-based solution increases at a much slower rate compared with that of its CLP counterpart. This is because, with the DSC approach, an additional predictor candidate would cause a bit-rate increase (when

coding a bit-plane) only if it has the worst correlation among all the predictor candidates (w.r.t. that bit-plane).

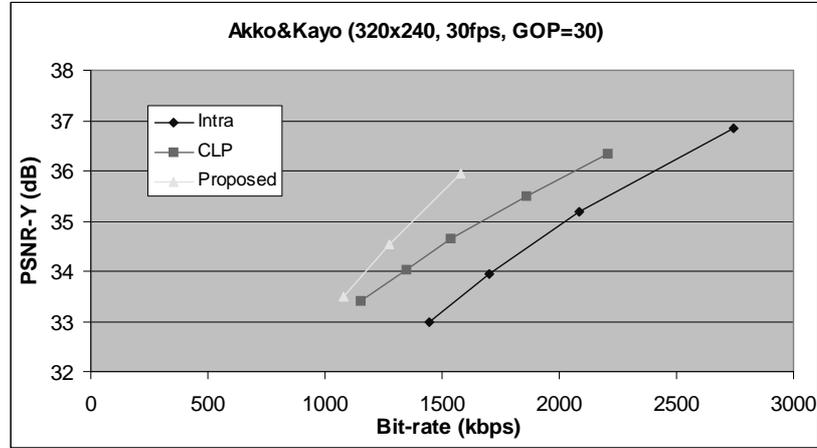


Figure 3.13: Simulation results of multiview video coding: Akko&Kayo. The results are the average of the first 30 frames (at the switching points) from 3 different views (views 27th-29th) arbitrarily chosen from all the available views.

3.6.2 Forward/Backward Video Playback

We then discuss our experiments on forward/backward playback application, where there are two predictor candidates (Figure 3.2). We compare our proposed algorithm with a CLP approach where both forward predicted H.263 P-frames and backward predicted H.263 P-frames are included. As discussed, such approach may incur drifting, since in general the reconstructed forward and backward predicted P-frames are not identical. We compare the schemes using sequences Coastguard and Stefan, which have considerable amounts of motion and picture details. As shown in Figures 3.18 and 3.19, the proposed algorithm outperforms CLP and intra coding. We also show the results of “normal” H.263 inter-frame coding (i.e., including only forward prediction residue) with the same

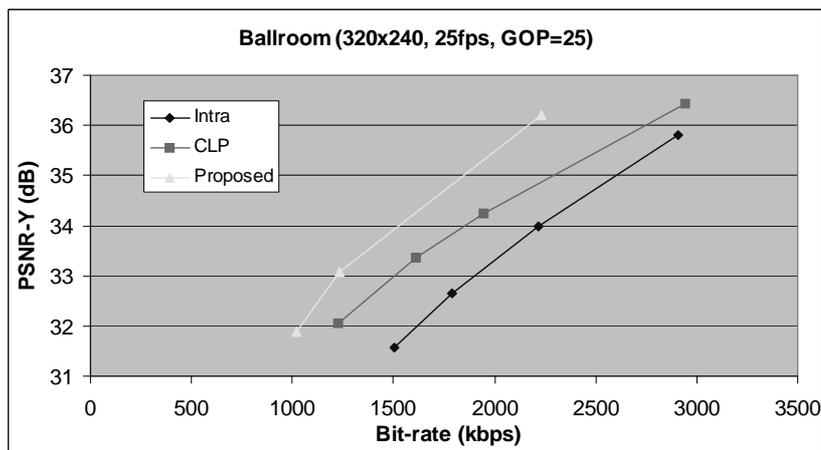


Figure 3.14: Simulation results of multiview video coding: Ballroom. The results are the average of the first 30 frames (at the switching points) from 3 different views (views 3rd-5th) arbitrarily chosen from all the available views.

GOP sizes. Note that inter-frame coding cannot support flexible decoding. The results are shown here for reference only.

We also compare the various approaches in terms of drifting with the following experiment: in forward decoding, a backward predicted frame is used for frame number 1 and as a reference for decoding the following frame. This is similar to what would happen when decoding direction switches from backward to forward. As shown in the results in Figure 3.20, the proposed algorithm incurs a negligible amount of drifting.

3.7 Conclusions and Future Work

We have proposed a video compression algorithm to support flexible decoding, based on DSC. The proposed algorithm integrates macroblock mode and significance coding to improve coding performance. Simulation results using MVC and forward/backward video playback demonstrate the proposed DSC-based algorithm can outperform the CLP

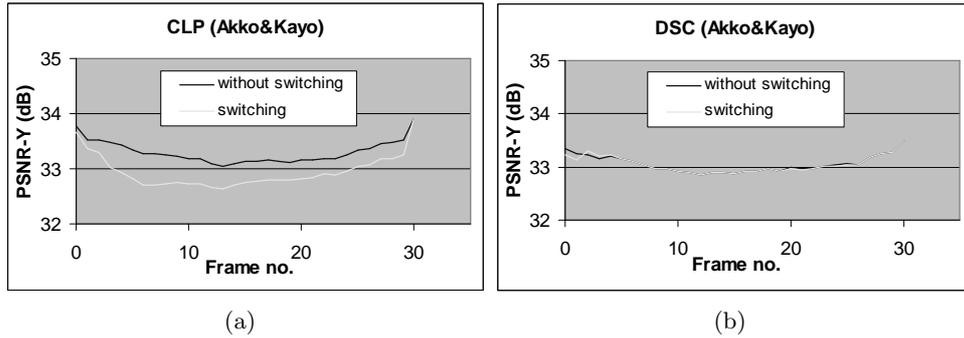


Figure 3.15: Drifting experiment using Akko&Kayo view 28th: (a) CLP; (b) DSC. GOP size is 30 frames. Note that with DSC, the PSNR are almost the same in the switching and non-switching cases.

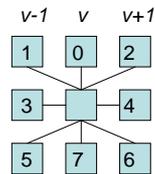


Figure 3.16: Scaling experiment: in this experiment, the temporally and spatially adjacent reconstructed frames are used as predictor candidates following the depicted order.

approach, while incurring only a negligible amount of drifting. Future work includes investigating improved model estimation methods.

Acknowledgment

The work was supported in part by NASA-JPL. The authors would like to thank Matthew Klimesh of NASA-JPL for pointing out the theoretical results by Csiszar and Korner.

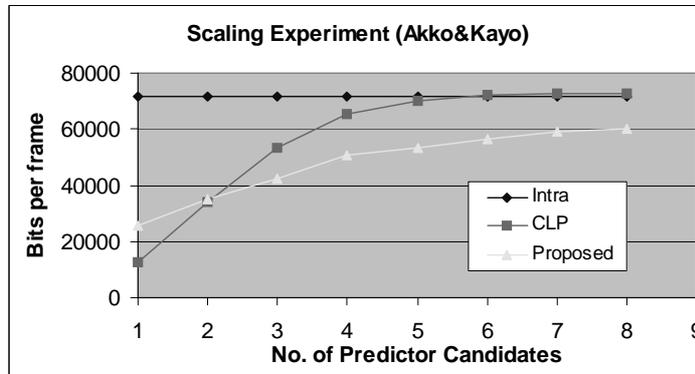


Figure 3.17: Scaling experiment using sequence Akko&Kayo view 29th. PSNR of different schemes are comparable - Intra: 35.07dB, CLP: 34.78dB, DSC:34.79dB.

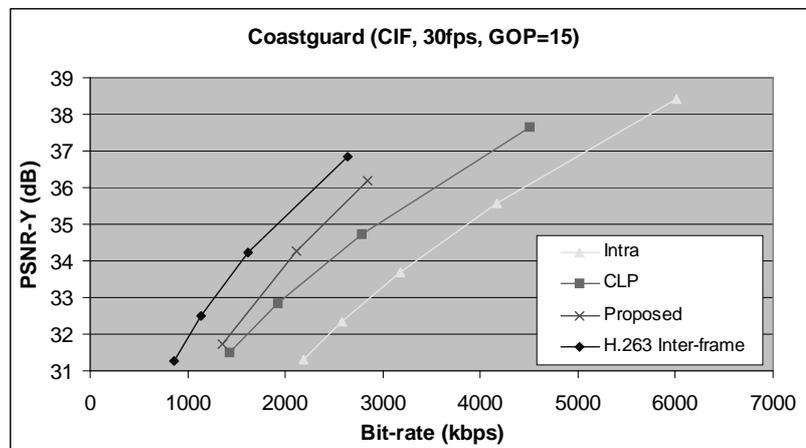


Figure 3.18: Simulation results of forward/backward video playback: Coastguard. Results are reported for the average of the first 30 frames. Note that H.263 inter-frame coding cannot support flexible decoding - the results are shown here for reference only.

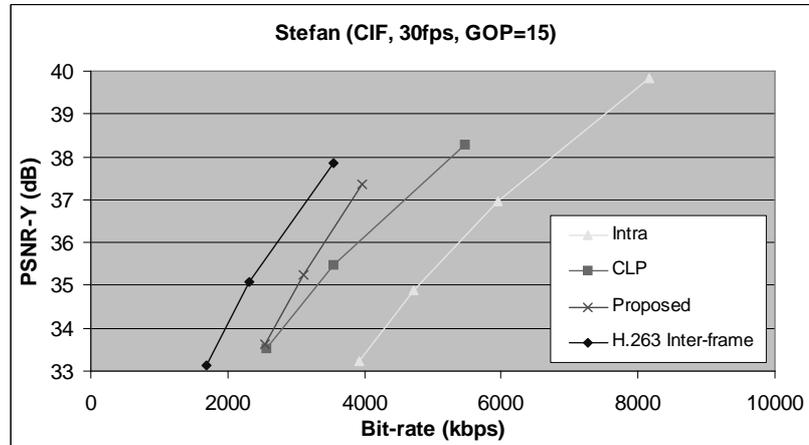


Figure 3.19: Simulation results of forward/backward video playback: Stefan. Results are reported for the average of the first 30 frames. Note that H.263 inter-frame coding cannot support flexible decoding - the results are shown here for reference only.

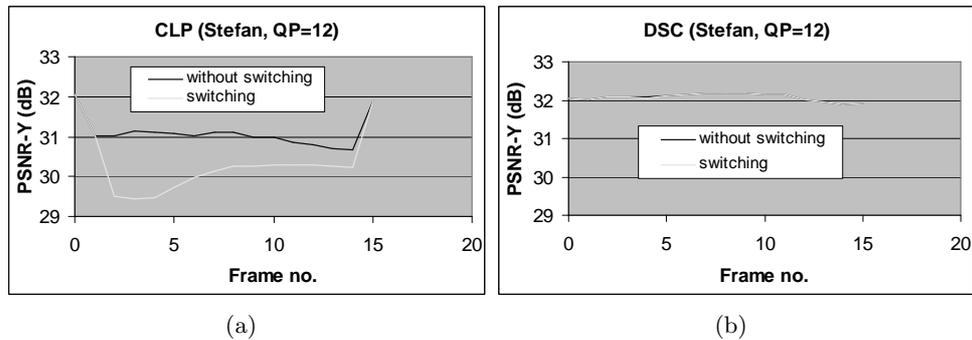


Figure 3.20: Drifting experiment using Stefan sequence: (a) CLP; (b) DSC. The figure shows the PSNR of the reconstructed frames in the first GOP. Note that with DSC, the PSNR are almost the same in the switching and non-switching cases.

Chapter 4

Correlation Estimation for Distributed Source Coding Under Rate and Complexity Constraints Using Sampling-based Techniques

4.1 Introduction

4.1.1 Motivation

Distributed source coding (DSC) [19, 63, 84] studies independent encoding and joint decoding of correlated sources, for which a correlation model is known at the encoder. Central to DSC is the information about existing correlation between the source and the side-information (SI) available at the decoder. Specifically, correlation information refers to the joint p.d.f. between the source and the SI. This correlation information plays several important roles in practical distributed source coding applications. First, many applications require correlation information at the encoder to determine the encoding rate. Essentially, the encoders use the correlation information to determine the number of *cosets* for partitioning the input space, so that error-free decoding can be

achieved [56, 59]. Second, for many practical Slepian-Wolf coding schemes that employ channel coding and iterative decoding, correlation information is required to initialize the decoding algorithms by providing likelihood estimates for the source bits [43]. Third, correlation information could be used at the decoder to determine the optimal reconstruction given the output of the Slepian-Wolf decoder and side information [56]. In this chapter we focus on estimating the correlation for the purpose of determining the encoding rate. The results may, however, be useful for the other two cases as well.

4.1.2 Correlation Information Models in Practical DSC Applications

At the heart of practical DSC applications is a lossless Slepian-Wolf (SW) codec, which plays a role similar to that of entropy codecs in conventional image/video compression. More precisely, the Slepian-Wolf encoder compresses a discrete i.i.d. source, which could be losslessly recovered at the decoder with the aid of the correlated side-information (SI) provided that enough compressed bits have been sent. Details on SW coding can be found in [28, 58, 85]. The problem to be investigated in this chapter is that of determining the amount of information communicated to the decoder, i.e., the encoding rate. From [63], an ideal rate to achieve a vanishing probability of decoding error is the conditional entropy of the input source given the SI ¹. The conditional entropy, in turn, depends on the correlation information between the source and SI. Thus rate allocation in DSC can be performed by solving the associated correlation estimation problem. Various types of correlation models (e.g., binary valued p.m.f. or continuous valued p.d.f.) may be

¹Practical SW coders would add a small margin to this ideal rate to account for using finite-length input blocks in SW coding.

exploited to compress the data depending on the specific SW coding algorithms and applications. We illustrate some of them below.

- **SW coding with binary correlation model.** These are cases where some form of *binary* valued joint p.m.f. is used to relate the source and SI (which are not necessarily binary) for SW coding. As an example, Figure 4.1(a) depicts SW coding with a binary correlation structure similar to that used in [14, 47, 72], etc. The continuous valued i.i.d. source X is mapped via scalar-quantization (or rounding) to a discrete source \tilde{X} . Then \tilde{X} is mapped to a bit-plane representation, and each extracted bit-plane $b_{\tilde{X}}(l)$, $l = 0 \dots L - 1$, is compressed independently by the SW encoder. Here $b_{\tilde{X}}(0)$ denotes the least significant bit-plane (LSB). The SW decoder recovers $b_{\tilde{X}}(l)$ with bit-plane $b_{\tilde{Y}}(l)$ extracted from the quantized version of the correlated source Y as side information. By exploiting the joint binary p.m.f. $p(b_{\tilde{X}}(l), b_{\tilde{Y}}(l))$, each extracted bit-plane can be compressed to a rate as low as $H(b_{\tilde{X}}(l)|b_{\tilde{Y}}(l))$. To determine this encoding rate, one can estimate the joint binary p.m.f. $p(b_{\tilde{X}}(l), b_{\tilde{Y}}(l))$, and derive the conditional entropy from the estimated p.m.f. Note that independent compression of each bit-plane facilitates efficient rate scalability, which is highly desirable in some imagery applications. For some applications it is also sufficient to achieve satisfactory coding performance by exploiting the correlation between corresponding bit-planes.

- **SW coding with continuous correlation model.** These are cases where a continuous valued joint p.d.f. is exploited for SW coding. Figure 4.1(b) illustrates an example similar to that proposed in [59]. The continuous i.i.d. source X is

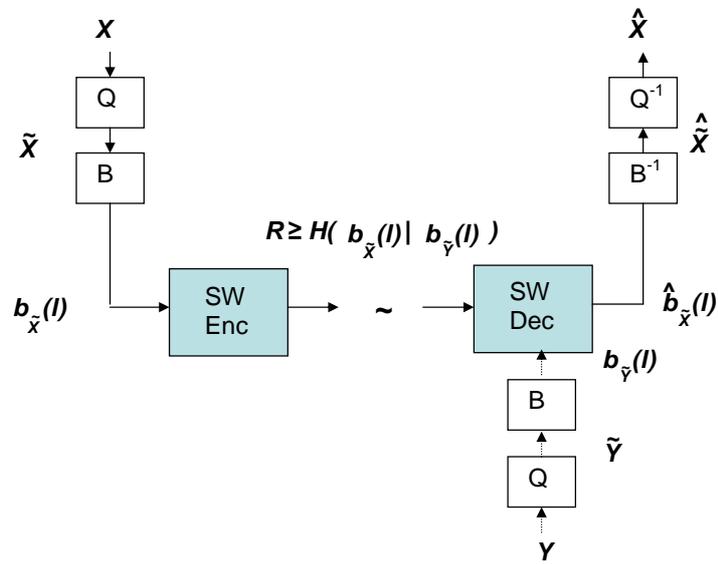
first scalar-quantized, and the quantized input \tilde{X} is directly compressed by the SW encoder. The decoder uses Y to recover \tilde{X} . By exploiting the joint p.d.f. $p(X, Y)$, the SW encoder can compress \tilde{X} to a rate as low as $H(\tilde{X}|Y)$. The encoder may determine this encoding rate from $p(X, Y)$.

4.1.3 Correlation Estimation in Distributed Source Coding

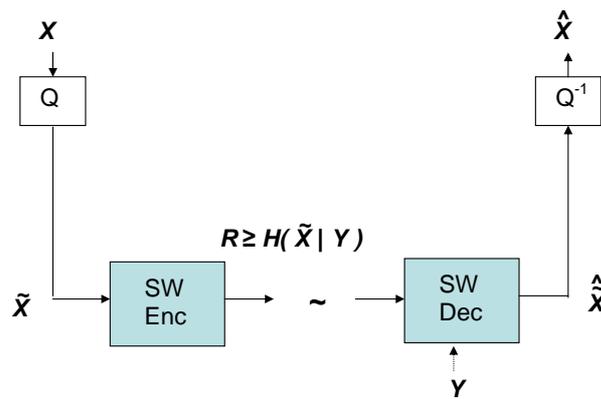
While both the underlying theory and the recently proposed code construction algorithms [1, 5, 27, 43, 44, 56] assume correlation information to be available *exactly* at the encoder, in many practical DSC applications, the correlation information may not be available beforehand, and one would need to estimate it during the encoding process [59]². The accuracy of this correlation estimation has a direct impact on the performance of DSC-based systems. While under-estimating the correlation may result in a penalty in coding efficiency, over-estimation can cause decoding error: in this case candidate decoded values within a given coset would be too close to each other, so that it is no longer possible to guarantee that they can be disambiguated without error by using the SI, leading to degradation in reconstruction quality.

Estimating the correlation information at the encoder is a non-trivial problem due to the computational and communication constraints imposed by the target applications, i.e., often, correlation estimation in DSC has to be performed under *rate* and *complexity constraints*. For example, when DSC is applied to compress wireless sensor measurements, it is important to limit the amount of data exchanged between nodes during correlation

²Note that in some cases, lack of an accurate correlation model is acceptable if there exists feedback from decoders to encoders [1], but this leads to an increase in overall delay.



(a)



(b)

Figure 4.1: (a) An example of Slepian-Wolf coding exploiting binary correlation. Boxes “ Q ” and “ Q^{-1} ” denote quantization and inverse quantization respectively. Boxes “ B ” and “ B^{-1} ” denote binarization and the inverse respectively. (b) An example of Slepian-Wolf coding exploiting continuous correlation.

estimation, in order to minimize the communication cost. Similarly, in applications such as video coding, source data needed to estimate the correlation is present at the encoder, but it is desirable to limit the computation resources devoted to this estimation [3, 59].

4.1.4 Our Contributions and Related Work

In this work we study correlation estimation strategies subject to rate and complexity constraints, and their impact on coding efficiency in a DSC framework. Our proposed algorithms are based on the observation that for many DSC applications side information is actually *available* at the encoder, but the encoder may not make use of this side information because of the associated communication or computational cost. As an example, in low complexity distributed video coding (DVC) [1, 49, 59], past frames that will be used as side information are available at the encoder, but the computation cost involved in performing motion estimation may be significant. Other examples can be found in distributed multiview image/video compression [31, 34, 88], wireless sensor data compression [55, 68, 85], etc. Focusing on these applications, we propose *sampling-based* algorithms to estimate the correlation information. Sampling is a well-established concept in statistics to infer the properties of a population from a small amount of individual observations [50]. To see how sampling applies to DSC consider these two examples:

- When compressing distributed sensors measurements, X , a node can request samples, Y , from the neighboring node in order to estimate the correlation $p(X, Y)$. The number of samples exchanged should be small, however, to keep the communications overhead low.

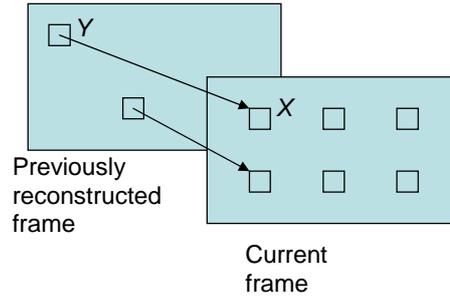


Figure 4.2: Apply sampling to distributed video coding: when encoding the current frame, we randomly sample n macroblocks of the current frame to undergo motion estimation, with n being much smaller than the total number of macroblocks. By doing so, we would obtain n pairs of samples (X, Y) , where X is the current macroblock and Y is the corresponding motion-compensated predictor. From these n sample pairs the encoding rate can be estimated. Note that here the sampling cost associated with each data sample is not primarily due to data exchange but the computation in motion search.

- In some DVC applications, the encoding rate depends on the joint p.d.f. between blocks in current frame (X) and the corresponding motion-compensated predictor blocks (Y) from reference frame [3, 11, 59]. Encoder can employ a sampling-based algorithm, where only a *small portion* of current frame's blocks would undergo motion estimation, so that the joint p.d.f. can be estimated from sample pairs (X, Y) , formed with a given current block and the corresponding predictor (Figure 4.2). Since motion search is required to acquire a sample pair (X, Y) each sampling operation would require some computational cost. Therefore, the proportion of blocks undergoing motion estimation should be small.

Sampling, however, leads to estimation errors and will have an impact on coding efficiency. Analyzing this impact is a key focus of our work. Since DVC applications may exploit various types of correlation models, it is difficult to address all of them. Therefore, we focus on one particular model in this chapter and briefly discuss how to analyze other models. Specifically, focusing on situations where a binary correlation is estimated for

SW coding (as discussed in Section 4.1.2) and correlation is estimated via sampling, this chapter makes the following contributions:

- **Rate penalty analysis in compression of a single binary source.** We analyze how estimation error in sampling impacts the coding performance of a DSC system when encoding a single binary source. We derive an expression to quantify how the number of samples relates to the increase in the encoding rate due to estimation error, taking into account that over-estimation can lead to significant increases in distortion in DSC applications (due to decoding error).
- **Rate penalty analysis and sample allocation in compression of multiple binary sources.** We then extend the rate penalty analysis to systems with multiple binary input sources, where each of them is compressed independently using SW coding with its corresponding side-information. Based on the analysis, we propose an algorithm to determine sampling rates to assign to each binary source so that the overall penalty in coding performance (due to estimation error) can be minimized, subject to a constraint on the total number of samples.
- **Model-based estimation in compression of a continuous source.** We then study encoding of a continuous input source. We consider scenarios where bit-planes are extracted from a continuous input source and each bit-plane is compressed via SW coding, e.g., as in [3, 14, 48, 86]. We propose a model-based method where the continuous-valued joint p.d.f. of the source and SI is first estimated via sampling of continuous valued inputs, and then the bit-plane level (binary) correlation is derived from the estimated model. This is in contrast to a direct approach where

the bit-plane correlation is estimated through exchanging binary samples from the extracted bit-planes. We demonstrate that the model-based method can achieve better estimation accuracy than the direct approach provided that the continuous-valued model is sufficiently accurate.

- **Model-based estimation for structured bit-planes.** We also describe how model-based estimation can be extended to the cases where bit-planes are extracted from continuous input data using more sophisticated methods. For example, in wavelet-based applications, bit-planes are separated into different “sub-bit-planes” depending on the magnitude (significance) of the transform coefficients. A concrete example of this, which we consider in this chapter, is that of bit-planes generated by *set-partitioning* as in SPIHT [61]. This type of bit-plane generation improves coding efficiency, but complicates the model-based correlation estimation process, as will be shown. Using a practical system as an example, we demonstrate that model-based estimation can lead to an additional advantage of efficient implementation in these types of DSC applications.

While this chapter focuses on cases where a binary correlation is used for SW coding, some of the proposed ideas may be extended to other types of correlation (details in Section 4.7).

Several methods have been proposed for correlation estimation problems in DSC. In DVC, low-complexity schemes to classify macroblocks into different correlation classes have been proposed [59], while other methods use a feedback channel to convey correlation information to the encoder [4]. For robust video transmission, recursive algorithms

have been proposed to estimate the correlation between the noise-free and noise-corrupted reconstructions [26, 78]. In our prior work, correlation estimation was performed by direct bit-plane comparisons between the source and an approximation of the decoder side-information [67]. This chapter proposes, however, several novel sampling-based correlation estimation algorithms applicable to a range of DSC applications, and presents the performance analysis.

A general approach for model-based estimation for DSC was first proposed in our work in [15]. The work focused on the simple cases where bit-planes are generated directly from the the binary representation of the sources. A recent work [33] has proposed a similar idea to show the advantage of Gray code representations, but does not discuss the exact algorithm to estimate the correlation.

This chapter is organized as follows. In Section 4.2 we present the rate penalty analysis. In Section 4.3 we propose the sample allocation algorithm to minimize the overall rate penalty. In Section 4.4 we propose the model-based estimation, and in Section 4.5 we extend the model-based estimation to cases where bit-planes are extracted based on the significance of the data. Section 4.6 presents experiments with real image compression applications. Section 4.7 discusses how to extend the rate penalty analysis to other correlation models. Finally, Section 4.8 concludes the work.

4.2 Single Binary Source: Rate Penalty Analysis

In this section we analyze how estimation error in sampling affects the compression performance of a DSC system in the case of a single binary source. Specifically, given that

n samples are used to estimate the correlation, we derive the corresponding increase in coding rate due to estimation error. As discussed, in most DSC systems there is a communication or computational cost associated with each acquired sample. Therefore, our results represent the trade-off between communication/computational cost and coding efficiency.

4.2.1 Problem Definition

We focus on the cases where a binary correlation is exploited in SW coding. Consider compressing a binary source b_X with another binary SI b_Y available at the decoder. We assume $\{b_X, b_Y\}$ i.i.d. $\sim p(b_X, b_Y)$. To simplify the analysis, we assume (i) b_X is equiprobable, i.e., $Pr[b_X = 0] = 0.5$, and (ii) the correlation is symmetric, i.e., $Pr[b_Y = 1|b_X = 0] = Pr[b_Y = 0|b_X = 1] = p$, where $0 \leq p \leq 0.5$ is the *crossover* probability for the sources. With these assumptions, the lower bound in the lossless encoding rate of b_X with b_Y available at the decoder is [19, 63]

$$H(b_X|b_Y) = H(p). \quad (4.1)$$

Therefore, encoder can estimate the lossless compression rate of b_X by estimating p through n random samples pairs $\{b_X, b_Y\}$. Define the estimation error $(\Delta p)(n) = \hat{p}(n) - p$, where $\hat{p}(n)$ is some estimation of p ³. When Δp is negative, this could lead to a decoding error. This is because the crossover probability is under-estimated and so the number of cosets chosen for encoding may be too small. Instead, when $\Delta p \geq 0$, correct decoding

³Note the dependency of \hat{p} (and other quantities) on n . Precisely, the p.d.f. of \hat{p} is a function of n , as will be discussed.

and lossless recovery of b_X can be guaranteed theoretically, but there will be a penalty in compression efficiency. This difference in behavior (decoding error vs. coding penalty) will lead us to propose a biased estimator such that $\Delta p \geq 0$ with high probability (discussed in the next section). On average, the coding penalty, in bits/sample, is given by (assuming no decoding error):

$$(\Delta H)(n) = H(\hat{p}(n)) - H(p). \quad (4.2)$$

As will be discussed, ΔH is indeed a random variable, since we randomly choose samples of b_X and b_Y for estimating $H(\hat{p})$. Our focus is to derive the probability density of $(\Delta H)(n)$.

4.2.2 Correlation Estimation

For encoding b_X we need to estimate \hat{p} by acquiring n random samples of b_Y . In different DSC applications, encoder may obtain the samples in different ways. For example, in a sensor application, the encoder of b_X may request samples of b_Y from a spatially-separated sensor node. In distributed video coding, the encoder may perform motion estimation to generate samples of b_Y . Common to most of the applications is that communication/computational costs will be incurred in acquiring the samples. Therefore, it is desirable to keep n small.

By inspecting the n pairs $\{b_X, b_Y\}$ now available at the encoder, an estimate of p can be computed to determine the encoding rate for b_X . A naive estimate for p is $\frac{S(n)}{n}$, where $S(n)$ is the number of inspected samples such that $b_X \neq b_Y$. Since S is essentially

the summation of n i.i.d. Bernoulli random variables with success probability p , S is a binomially distributed r.v. with mean np and variance $np(1-p)$. Therefore, for sufficiently large n ,

$$\frac{S(n)}{n} \sim N(p, \sigma^2) \quad , \quad \sigma = \sqrt{p(1-p)/n}. \quad (4.3)$$

As a consequence, if we use $\frac{S}{n}$ as the estimator there is 50% probability of under-estimation of p . Therefore, we opt to use a *biased estimator* given by

$$\hat{p}(n) = \frac{S(n)}{n} + z_{\omega/2}\sigma. \quad (4.4)$$

That is, we bias $\frac{S}{n}$ by a factor proportional to σ . By choosing the constant $z_{\omega/2}$ we can control precisely the probability of under-estimation of p , e.g., if $z_{\omega/2} = 1.64$, $Pr[\hat{p} < p] = \omega/2 = 0.05$. We choose this biased estimator to minimize the risk of decoding failure, at the expense of some encoding rate penalty.

4.2.3 Rate Penalty Analysis

Using (4.4) as the estimator, we analyze how n relates to the p.d.f. of ΔH . From (4.4),

$$\begin{aligned} (\Delta p)(n) &= \hat{p}(n) - p \\ &= \frac{S(n)}{n} + z_{\omega/2}\sigma - p \sim N(z_{\omega/2}\sigma, \sigma^2). \end{aligned} \quad (4.5)$$

Expanding $H(\cdot)$ at p by a Taylor series and using the definition of ΔH in (4.2),

$$\begin{aligned} (\Delta H)(n) &= H'(p)\Delta p + \frac{H''(p)(\Delta p)^2}{2!} + \dots \\ &\approx H'(p)(\Delta p)(n). \end{aligned} \quad (4.6)$$

The approximation holds when Δp is sufficiently small. The p.d.f. of ΔH can then be derived as:

$$(\Delta H)(n) \sim N(H'(p)z_{\omega/2}\sigma, (H'(p))^2\sigma^2), \quad (4.7)$$

where $H'(p) = \ln(\frac{1}{p} - 1)$ and σ is given by (4.3). (4.7) relates n to the density of ΔH . Using (4.7), one can readily compute some statistics for ΔH , e.g., $E[(\Delta H)(n)]$. Note that these statistics are functions of n .

In practice, since p is unknown, σ is unknown when computing the estimator (4.4).

A good rule of thumb is to approximate the estimator using [50]

$$\hat{p}(n) \approx \frac{S}{n} + z_{\omega/2}\sqrt{\frac{S}{n}(1 - \frac{S}{n})/n}, \quad (4.8)$$

i.e., $\frac{S}{n}$ is used to approximate p in computing the estimator. The approximation would be valid when $n \cdot \frac{S}{n} \geq 4$ and $n(1 - \frac{S}{n}) \geq 4$, as a rule of thumb [50].

4.2.4 Experiment

In this section we assess the accuracy of the rate penalty model proposed in (4.7). Specifically, we perform sampling experiments and measure ΔH , and compare the empirical

distribution of ΔH with (4.7). Video data is used for the experiments. As the binary input source (b_X), we use the raw bit-planes extracted from all (quantized) DCT coefficients of a given frequency in the current frame, while as SI (b_Y) we use the raw bit-planes of same significance extracted from the corresponding (quantized) DCT coefficients of the motion-compensated predictors in the reference frame⁴. Therefore, the dimension of the source, M , is equal to the number of DCT blocks in a frame. We then sample n ($< M$) pairs of $\{b_X, b_Y\}$ randomly, and an estimation \hat{p} is then computed according to (4.8) from the chosen pairs. With \hat{p} , a single ΔH can then be obtained using (4.2). The sampling experiment is repeated N_E times, each time with different pairs of $\{b_X, b_Y\}$ sampled and a different ΔH obtained. We compare the empirical p.d.f. of ΔH (with N_E data) with the model in (4.7) using Kolmogorov-Smirnov (K-S) tests [50]. Table 4.1 shows the resulting K-S statistics at different sampling rates for some bit-planes extracted from the DC coefficients quantized at $QP = 24$ (H.263 quantization) in the 2nd frame of *Mobile* (720×576 , 30 fps), with $z_{\omega/2} = 1.64$ and $N_E = 100$. In particular, for the range of n and p where $n \cdot p \geq 4$, K-S tests approve the hypothesis that the empirical ΔH follows the model in (4.7). This results indicate that our proposed model can adequately characterize the distribution of the rate penalty for practical problems with sufficiently large $n \cdot p$, e.g., $n \cdot p \geq 4$. Note that a sampling size of 128 corresponds to about 2% of data for a 720×576 video. Additional results using different data lead to similar conclusions.

⁴Note that it is common for DVC systems to exploit correlation between the DCT coefficients in the current frame and the corresponding coefficients in the motion-compensated predictors in the reference frame, e.g., [3, 11].

Table 4.1: Kolmogorov-Smirnov (K-S) tests statistics for ΔH : Mobile DC, QP=24. Numbers in parenthesis indicate cases that K-S statistics are larger than the 1% significance cutoff value, 0.1608, and therefore do not pass K-S tests. Those are cases when $n \cdot p \geq 4$ does not hold. Note that bit position 6 corresponds to a more significant bit-plane.

Bit Position	6 $p = 0.0228$ $H(p) = 0.1571$	5 $p = 0.0529$ $H(p) = 0.2987$	4 $p = 0.0961$ $H(p) = 0.4566$	3 $p = 0.2002$ $H(p) = 0.7222$
$n = 96$	(0.1912)	0.1470	0.1310	0.1510
$n = 128$	0.1580	0.1401	0.1266	0.1358
$n = 256$	0.1458	0.1343	0.1073	0.1080
$n = 512$	0.1459	0.1219	0.0943	0.0957

4.3 Multiple Binary Sources: Rate Penalty Analysis and Sample Allocation

In this section we study compression of multiple binary input sources. The multiple sources scenario can arise in different applications. One example is the compression of multiple streams of sensor measurements captured in different nodes. Another important example is the compression of a continuous input source, where the source is first mapped to a bit-plane representation and then each bit-plane is compressed using DSC, so that the problem becomes one of compression of multiple binary sources. From (4.7), in the single source case the rate penalty due to estimation error depends on both sampling, i.e., n , and the characteristics of the source, i.e., p . In a system with multiple input sources each with a different p , we now investigate the optimal sample allocation to each source such that the overall rate penalty can be minimized, subject to the constraint on the total number of sample allocated to the system.

4.3.1 Problem Definition

Consider the compression of L binary sources $b_X(l)$, $l = 0$ to $L - 1$. Each binary source is independently encoded using SW coding with its respective SI $b_Y(l)$ available at the decoder. We shall follow the assumptions in Section 4.2, i.e., $\{b_X(l), b_Y(l)\}$ i.i.d. $\sim p(b_X(l), b_Y(l))$, with $Pr[b_X(l) = 0] = 0.5$ and crossover probability $Pr[b_X(l) \neq b_Y(l)] = p_l$. Let K_l be the number of binary values to be encoded for source $b_X(l)$. We follow the correlation estimation procedure in Section 4.2, where encoding $b_X(l)$ requires observing n_l ($\leq K_l$) random samples of $b_Y(l)$ in order to compute the biased estimator $\hat{p}_l(n_l)$ according to (4.4) (or (4.8) in practice), so that the encoding rate for $b_X(l)$ can be determined. The encoding of $b_X(l)$ would suffer a rate penalty $(\Delta H_l)(n_l) = H(\hat{p}_l(n_l)) - H(p_l)$. In particular, following the discussion in Section 4.2 and using (4.7), ΔH_l would be normally distributed:

$$(\Delta H_l)(n_l) \sim N(H'(p_l)z_{\omega/2}\sigma_l, (H'(p_l))^2\sigma_l^2), \quad (4.9)$$

where $H'(p_l) = \ln(\frac{1}{p_l} - 1)$ and $\sigma_l = \sqrt{p_l(1-p_l)/n_l}$. On average, the coding penalty of the whole system, in bits/sample, is given by:

$$\Delta H = \frac{1}{K_T} \sum_{l=0}^{L-1} K_l \Delta H_l. \quad (4.10)$$

where $K_T = \sum_{l=0}^{L-1} K_l$. Note that in this section ΔH refers to the average coding penalty of the entire system with L sources. Since the samplings are performed independently on

each source, ΔH_l are independent r.v., and therefore ΔH is normally distributed with expectation and variance given by:

$$E[\Delta H] = \frac{1}{K_T} \sum_{l=0}^{L-1} K_l H'(p_l) z_{\omega/2} \sigma_l, \quad (4.11)$$

$$VAR[\Delta H] = \sum_{l=0}^{L-1} \left(\frac{K_l}{K_T} \right)^2 (H'(p_l))^2 \sigma_l^2. \quad (4.12)$$

The total number of samples is limited to be n_T , i.e., $\sum_{l=0}^{L-1} n_l = n_T$, under the assumption that we would like to have $n_T \ll K_T$, because each sample would incur a communication/computational cost.

Our main goal is to minimize $E[\Delta H]$ subject to a given n_T . Note that $E[\Delta H]$ is a function of (i) $\mathbf{p} = \{p_l\}$, correlation of different sources, (ii) n_T , total number of samples used to estimate correlation, and (iii) $\mathbf{n} = \{n_l\}$, allocation of samples to different sources.

In the following sections:

1. we derive an optimal sample allocation strategy, i.e., given $\mathbf{p} = \{p_l\}$, n_T , we find the optimal $\mathbf{n} = \{n_l\}$ to minimize the rate penalty $E[\Delta H]$;
2. given the optimal sample allocation, we study how $E[\Delta H]$ changes with n_T .

As will be discussed in the next section, the optimal sample allocation requires knowledge of $\{p_l\}$, which obviously is not known a priori. Several strategies will therefore be described to apply our results in practice.

4.3.2 Optimal Sample Allocation Strategy

In this section we seek to find the optimal number of samples to allocate to different sources, $\{n_l^*\}$, so as to minimize $E[\Delta H]$. To find $\{n_l^*\}$, we solve the following constrained optimization problem:

$$\min_{\{n_l: \sum_{l=0}^{L-1} n_l = n_T; n_l \leq K_l\}} E[\Delta H], \quad (4.13)$$

where $E[\Delta H]$ is given by (4.11). Applying the Lagrangian optimization method and Kuhn-Tucker conditions to deal with the inequalities constraints, we obtain (details in Appendix B):

$$n_l^* = \begin{cases} \gamma(K_l \alpha_l)^{2/3}, & \text{if } \gamma < \frac{K_l^{1/3}}{\alpha_l^{2/3}}, \\ K_l, & \text{if } \gamma \geq \frac{K_l^{1/3}}{\alpha_l^{2/3}}, \end{cases} \quad (4.14)$$

where α_l is a constant that depends on $z_{\omega/2}$ and p_l :

$$\alpha_l = \ln\left(\frac{1}{p_l} - 1\right) z_{\omega/2} \sqrt{p_l(1-p_l)}, \quad (4.15)$$

and γ is chosen so that $\sum_{l=0}^{L-1} n_l^* = n_T$. This result gives rise to a sample allocation scheme analogous to the “water-filling” results in information theory [19]. We allocate equal *weighted* number of samples to each source, until for some sources the number of samples is equal to the number of source inputs. The weighting factor $(K_l \alpha_l)^{2/3}$ is a constant that depends only on the specific characteristics of the l th source (length and crossover probability).

In situations where $n_l < K_l$ can be guaranteed for all sources (e.g., when n_T is small enough such that $n_T < K_l$ is true for all l), (4.14) can be simplified to: $n_l^* =$

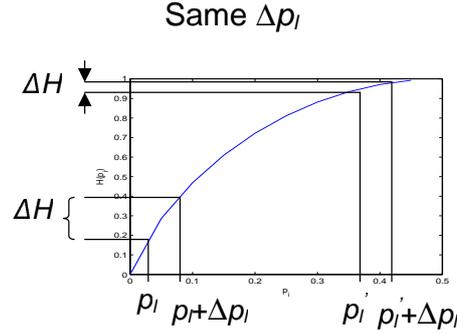


Figure 4.3: Encoding rate function $H(\cdot)$. The same Δp_l would have a larger impact to the estimated rate if the true p_l is small. The optimal sample allocation takes into account this characteristic.

$n_T \frac{(K_l \alpha_l)^{\frac{2}{3}}}{\sum_{l=0}^{L-1} (K_l \alpha_l)^{\frac{2}{3}}}$. In the important case when raw bit-planes are extracted from a continuous valued source and compressed independently by SW coding, similar to the example in Figure 4.1(a), K_l would be the same, and $\{n_l^*\}$ reduces to

$$n_l^* = n_T \frac{\alpha_l^{\frac{2}{3}}}{\sum_{l=0}^{L-1} \alpha_l^{\frac{2}{3}}}. \quad (4.16)$$

The intuition behind the optimal sample allocation can be understood by inspecting the rate function $H(\cdot)$. For the same estimation error $(\Delta p_l)(n_l) = \hat{p}_l(n_l) - p_l$, the impact to the encoding rate will tend to be larger when the true p_l is small (Figure 4.3). Since different sources have different p_l , we should allocate the samples accordingly and use more samples for those sources with small p_l , so as to minimize the overall rate penalty. This is reflected in (4.14) and (4.15).

Since we have chosen to use the same $z_{\omega/2}$ in all sources, we have the same probabilities of over-estimation for all sources. In some applications it may be more appropriate to choose different target failure probabilities for different sources, e.g., in the cases when the sources are bit-planes extracted from a continuous valued source, MSB bit-planes are

more important and should have smaller failure probabilities. This can be incorporated in the sample allocation by using different $z_{\omega/2}$ for different sources in (4.11).

Note that the optimal sample allocation depends on the crossover probabilities $\{p_l\}$. However, $\{p_l\}$ is obviously not known initially. In practical applications, the results of optimal sample allocation can be applied in the following ways:

- In many applications, we may have some a priori knowledge of $\{p_l\}$. For example, in hyperspectral image compression [14], where bit-planes of spectral bands are extracted and compressed, $\{p_l\}$ of neighboring spectral bands are usually similar. Therefore, we can use some approximations of $\{p_l\}$ in the optimal sample allocation equations to determine the sample assignment. We will present experimental results in Section 4.6 to demonstrate that this can be a viable method. Note that using the a priori knowledge directly to select the encoding rate may cause decoding error, if this a priori knowledge leads to over-estimating the correlation. Instead, by using the a priori knowledge to determine the sample allocation and (4.8) as the estimator to select the coding rate we are guaranteed that \hat{p}_l is larger than p_l with probability $(1 - \omega/2)$, and we can bound decoding error systematically. Also this is more robust in cases where the a priori knowledge may not be a good approximation to the true $\{p_l\}$.
- We can also use an iterative approach similar to [41]. Essentially, we would allocate samples in batches of same size. For the first batch, we allocate the same number of samples to all bit-planes and obtain some initial estimates for $\{p_l\}$. For the subsequent batches, we allocate the number of samples according to the current

estimates and the optimal sample allocation strategy. In this approach, we can also use any available a priori knowledge to initialize the scheme.

4.3.3 Rate Penalty Analysis - Multiple Sources

Having an expression for $E[\Delta H]$ as a function of n_T allows the encoder to select appropriate values for n_T , given that increasing n_T leads to additional overhead but also reduces the rate increase due to inaccurate estimation. We focus on the cases when $n_l < K_l$, where closed form equations can be derived. The equations relating $E[\Delta H]$ to n_T can be obtained from (4.11) and (4.14):

$$E[\Delta H] = \frac{\beta}{\sqrt{n_T}}, \quad (4.17)$$

where $\beta = \frac{1}{K_T} \left[\sum_{l=0}^{L-1} (K_l \alpha_l)^{2/3} \right]^{3/2}$. Note that β is a constant for a given system. Therefore, the average rate penalty is inversely proportional to the squared root of the amount of sampling overhead.

4.3.4 Experiments

In this section we assess the benefits of using the optimal sample allocation when compressing i.i.d. sources. We randomly generate L pairs of binary correlated sources $\{b_X(l), b_Y(l)\}$ each with crossover probability p_l and dimension K_l . Then n_T samples are used to estimate the correlation. The number of samples allocated to each source is determined according to the following schemes:

- *Optimal allocation.* We use (4.14) to determine n_l , the number of samples allocated to the l -th source.
- *Even allocation.* We allocate the same number of samples to each source, i.e., $n_l = n_T/L$.

Note that in this section we use the true crossover probabilities to determine the optimal allocation, while similar comparison using some a priori information to determine the optimal allocation in practical scenarios will be discussed in Section 4.6. The schemes are compared based on the average (overall) rate penalty after N_E sampling experiments. We compute the reduction in the average rate penalty (in bits) as: $E[\Delta H]_{even} - E[\Delta H]_{opt}$, where $E[\Delta H]_{opt}$ is the average rate penalty using the optimal sample allocation, and $E[\Delta H]_{even}$ is that of using the even sample allocation. Since there are many possible combinations of $\{p_l\}$, as an example we choose $\{p_l\}$ of the form $\{\bar{p} + k\delta_p\}$, $k = \pm 1, \pm 2, \dots, \bar{p} = 0.25$. Therefore, a large δ_p corresponds to more substantial variation (standard deviation) in $\{p_l\}$. Figure 4.4 depicts the comparison results and shows that over 0.07 bits reduction in rate penalty can be achieved in the case of considerable variation in $\{p_l\}$, with this extra number of bits representing a 11.7% rate increase. In practical applications diverse crossover probabilities are indeed common (e.g., see [14] or data in Section 4.2.4).

4.4 Continuous Input Source: Model-based Estimation

In this section we investigate correlation estimation methods in the particular but important cases when bit-planes are extracted from a continuous input source and each bit-plane

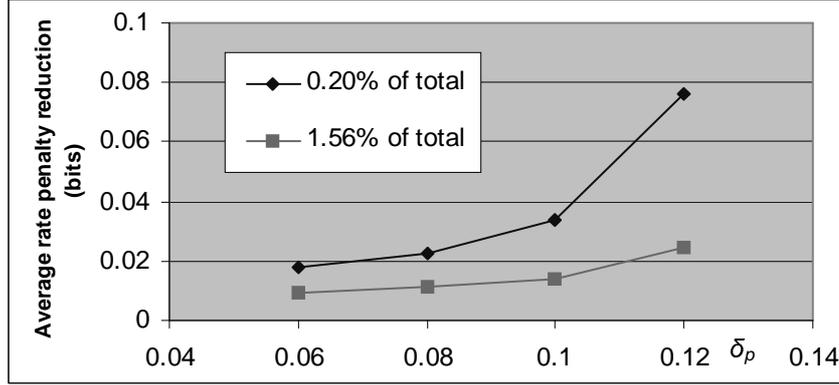


Figure 4.4: Reduction in rate penalty (bits) using the optimal sample allocation, with $L = 4$, $K_l = 16384$, $n_T = 128$ or 1024 (i.e., 0.20% or 1.56% of total respectively), $N_E = 100$, and $\bar{p} = 0.25$.

is compressed via SW coding. This situation arises in several proposed distributed image and video coding algorithms (e.g., [3, 14, 48, 86]). Often, in these applications, some a priori statistical model knowledge of the continuous-valued input source is available. For example, wavelet and DCT transform coefficients are typically considered to be well modeled by Laplacian distributions [65]. In the following we propose a model-based estimation method, where the continuous-valued joint p.d.f. of the source and SI is first estimated via sampling of continuous valued inputs⁵, and then the bit-plane level (binary) correlation is derived from the estimated model. This is in contrast to the direct approach studied in Sections 4.2 and 4.3, where the bit-plane correlation is estimated through exchanging binary samples from the extracted bit-planes. We shall demonstrate the model-based method can achieve better estimation accuracy than the direct approach provided that the continuous-valued model is sufficiently accurate.

⁵In practice, the continuous valued inputs are rounded so that the samples can be represented with a finite number of bits.

4.4.1 General Approach

We shall focus on the setting of Figure 4.1(a), where binary correlation is exploited for compression of a continuous input X using continuous side-information Y . Assume X and Y are drawn i.i.d. from $f_{XY}(x, y)$. We assume $Y = X + Z$, where Z is the correlation noise independent of X . Our proposed model-based approach starts by estimating the p.d.f.'s $f_X(x)$ and $f_Z(z)$. This can be done by choosing appropriate models for the data samples and then estimating the model parameters using one of the standard parameter estimation techniques, e.g., maximum likelihood estimation (MLE).

Once we have estimated $f_X(x)$ and $f_Z(z)$ we can derive the bit-plane statistics as follows. Suppose we extract *raw* bit-planes from the binary representations of X and Y , and are interested in estimating p_l , the crossover probability between the bit-planes of X and Y at significance level l . Figure 4.5(a) depicts the events (shaded regions A_i) that lead to the occurrence of crossover between X and Y at significance level l . For example, consider $l = 2$ (i.e., the 2nd bit-plane), when X takes the values from 8 (= 01000b) to 11 (= 01011b), crossover occurs when Y takes the values from 4 (= 00100b) to 7 (= 00111b) (region A_4 in Figure 4.5(a)), or 12 (= 01100b) to 15 (= 01111b) (region A_5 in Figure 4.5(a)), ..., etc. Specifically, A_i is a subset of \mathbb{R}^2 defined as

$$A_i = \{ (x, y) \mid 2c \cdot 2^l \leq |x| < (2c + 1) \cdot 2^l, (2d + 1) \cdot 2^l \leq |y| < (2d + 2) \cdot 2^l; \text{ or} \\ (2g + 1) \cdot 2^l \leq |x| < (2g + 2) \cdot 2^l, 2h \cdot 2^l \leq |y| < (2h + 1) \cdot 2^l \}, (4.18)$$

for some $c, d, g, h \in \mathbb{Z}^+$. Hence we can estimate the crossover probability at bit-plane l by

$$\begin{aligned}\hat{p}_l &= \sum_i \int \int_{A_i} f_{XY}(x, y) dx dy \\ &= \sum_i \int \int_{A_i} f_X(x) f_{Y|X}(y|x) dx dy\end{aligned}\tag{4.19}$$

Assuming that $Y = X + Z$ and that X, Z are independent, $f_{Y|X}(y|x)$ can be found to be equal to

$$f_{Y|X}(y|x) = f_Z(y - x)\tag{4.20}$$

and the integral in (4.19) can be readily evaluated for a variety of densities. In practice we only need to sum over a few regions, A_i , where the integrals are non-zero. Note that when l is small (i.e., least significant bit-planes) the crossover probability is close to 0.5, since in such cases A_i are small and evenly distributed throughout the sample space, and hence for most models (4.19) will give \hat{p}_l close to 0.5.

In Section 4.5, we will discuss how to extend model-based estimation to the cases when bit-planes are extracted using more sophisticated methods, in particular those being used in wavelet-based applications.

4.4.2 Experiments

We now compare the accuracy of direct estimation and model-based estimation. We generate i.i.d. Laplacian sources X and Z of dimension M with model parameters β and α respectively, i.e., $f_X(x) = \frac{1}{2}\beta e^{-\beta|x|}$, $f_Z(z) = \frac{1}{2}\alpha e^{-\alpha|z|}$. Then the crossover probability

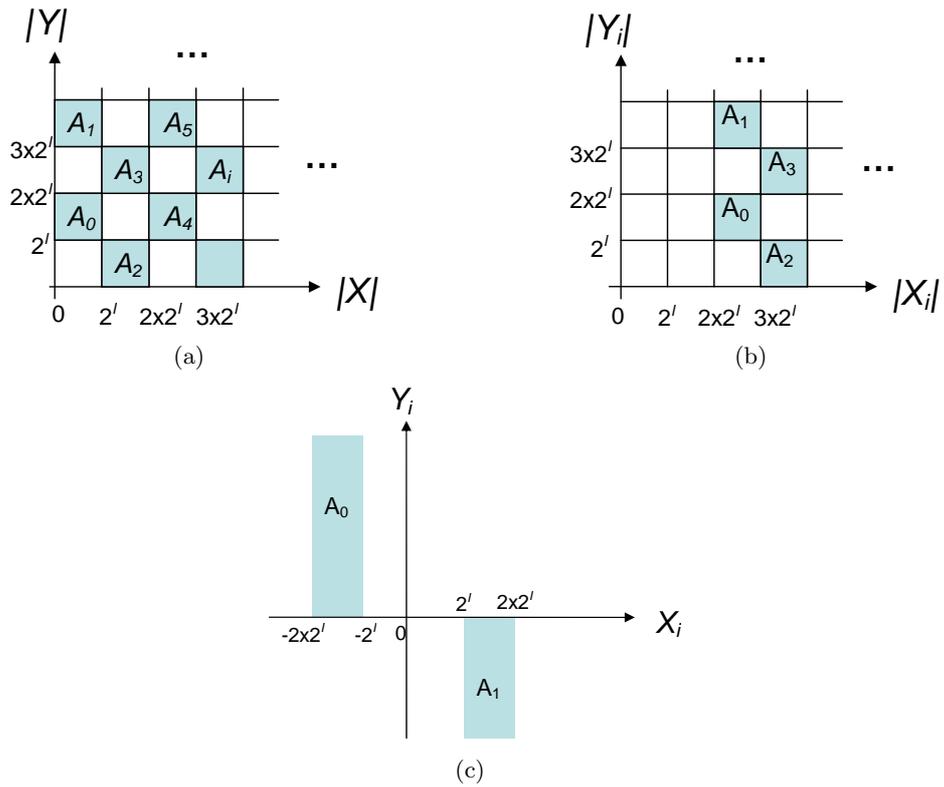


Figure 4.5: (a) Crossover probability estimation for raw bit-planes. A_i are the events that lead to occurrence of crossover between X and Y at significance level l . (b) Refinement bit-plane crossover probability estimation: probability of crossover and X_i is already significant. (c) Sign bit-plane crossover probability estimation: probability of sign crossover and X_i becomes significant.

p_l of X and $Y(= X + Z)$ at significance level $l, 0 \leq l \leq L - 1$, is estimated using the following approaches:

- *Direct estimation with even sample allocation.* This is similar to the estimation method in Section 4.2, where n binary samples of the l -th bit-planes are exchanged. Since there are L bit-planes in total, the total amount of exchanged data is $L \cdot n$ bits.
- *Direct estimation with optimal allocation.* This is similar to the aforementioned approach except that the optimal sample allocation (4.14) is used to distribute the $L \cdot n$ binary samples among bit-planes.
- *Model-based estimation.* Here n L -bits random samples of Y are sent to the encoder of X , where the model parameters β and α are estimated from the n samples of X and $Z(= Y - X)$ respectively, using MLE [50]. Then the estimate of $p_l, 0 \leq l \leq L - 1$, can be derived analytically from $\hat{\beta}$ and $\hat{\alpha}$ using (4.19). Therefore, the model-based approach also incurs $L \cdot n$ bits to estimate the crossover probabilities of all the bit-planes.

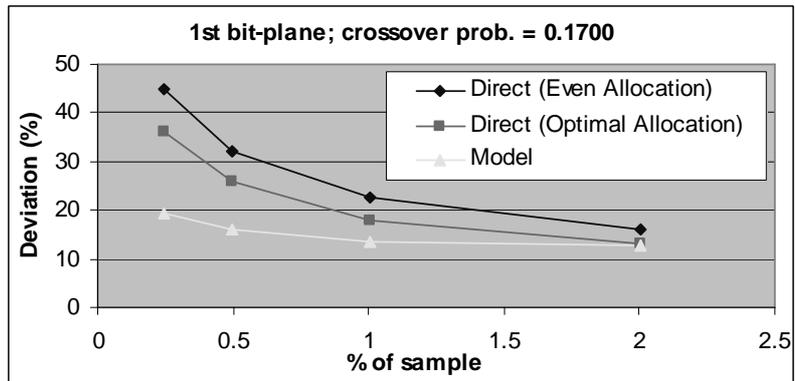
Note that in the direct approach we do not include offset in the estimator, i.e., the estimator is $\frac{S(n)}{n}$ following the notations in Section 4.2. Therefore, it is fair to compare with (4.19). Practical applications may choose to bias both the direct and model-based estimators as in (4.8) (in model-based estimation we would replace $\frac{S}{n}$ by that calculated in (4.19)). The approaches are compared based on the deviation of the estimates from the true (empirical) crossover probability: $|\hat{p}_l - p_l|/p_l$. The deviation is measured for different bit-planes using different percentage of exchanged samples, $\frac{n}{M}$, with $\beta = 0.3, \alpha = 2.5$,

$M = 6480$. These parameters are similar to those observed in the video data used in the experiments in Section 4.2.4, i.e., X and Y are the quantized DCT coefficients in a current frame and the corresponding quantized coefficients in the motion-compensated predictors in the reference frame, respectively. Figure 4.6 depicts the comparison results, where the deviations are obtained by averaging over $N_E = 1000$ experiments. As shown in the figure, the model-based estimation can achieve considerable improvements in the estimation accuracy, especially when only a small number of samples are used or crossover probability is small. Note that model-based estimation utilizes the information that the bits to be encoded have been extracted from continuous-valued data following certain distributions, and therefore would tend to perform better than direct estimation, which does not use such information. However, model-based estimation is applicable only to bit-planes extracted from continuous sources, and obviously its performance depends on how accurately the continuous-valued data can be modeled. Additional experiments assessing the performance in terms of coding rate and distortion using a real application will be presented in Section 4.6.

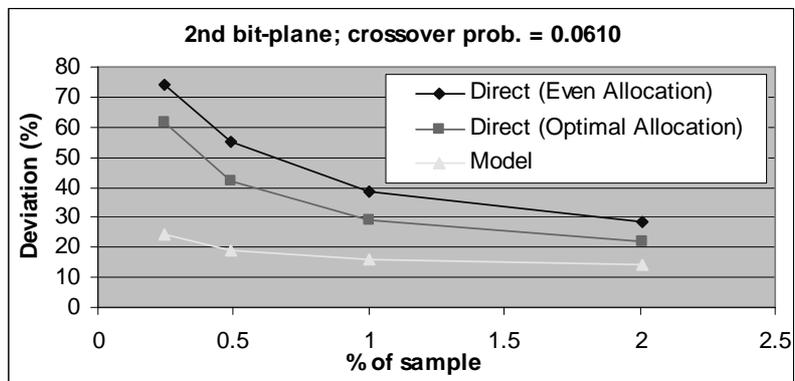
4.5 Model-based Estimation on Structured Bit-planes

In this section we discuss how to extend the model-based estimation to the cases when bit-planes are extracted using more sophisticated methods. For example, in the cases when X and Y are wavelet transform coefficients⁶, bit-planes are usually partitioned depending on the magnitude of the transform coefficients to improve coding efficiency, as

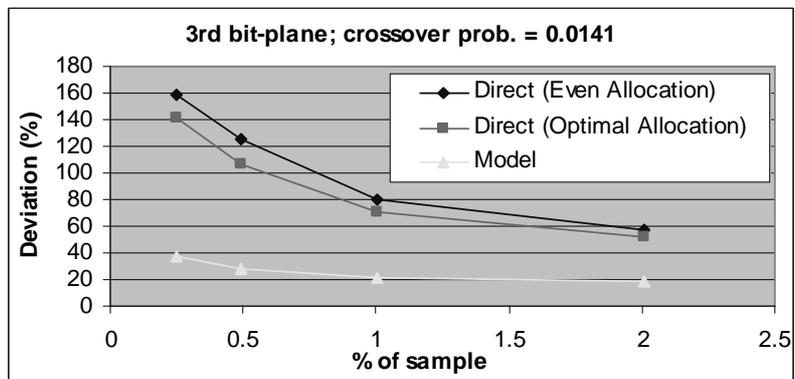
⁶A concrete application scenario can be X and Y are collocated wavelet transform coefficients of two correlated images.



(a) The 1st bit-plane.



(b) The 2nd bit-plane.



(c) The 3rd bit-plane.

Figure 4.6: Comparing estimation accuracy between the direct approaches and the model-based approach.

in the *set-partitioning* algorithm used in SPIHT [61]. Specifically, in these “significance coding” techniques, the encoder first signals the *significance* of each of the components at a given bit-plane. After a component becomes significant, *sign* information is conveyed and then further *refinement* bits are transmitted. While different wavelet systems may use different techniques to encode the bits (e.g., context coding can be used as in JPEG2000 or alternatively zerotree coding can be used as in SPIHT to represent significance maps), the definitions of the (uncoded) sign/refinement bits and significance maps are largely the same, so the techniques we propose in the context of SPIHT in this section would also be applicable to other wavelet-based compression schemes that use bit-plane encoding. We shall consider systems where SW coding is applied to compress the sign/refinement bit-planes⁷, and propose extension of model-based approach to estimate the corresponding crossover probabilities, so that encoding rate can be determined. The extended approach can also be applied to facilitate adaptive combinations of SW/entropy coding to improve coding performance [8].

In what follows we will first discuss how to extend model-based approach to estimate crossover probabilities of sign/refinement bit-planes. Since significance coding is usually used in wavelet-based applications (e.g., [61,71]), we will also discuss how to address some of the issues when applying model-based estimation in wavelet-based DSC applications.

⁷Since significance map carries structural information, a single decoding error in the significance map would cause decoding failure of all the subsequent bits, and therefore SW coding may not be suitable for compressing significance map.

4.5.1 Model-based Estimation for Refinement/Sign Bit-planes

Given an input source X_i to be compressed using side information $Y_i (= X_i + Z_i)$, and following the same assumptions in Section 4.4.1, our goal is to estimate the crossover probability of the refinement and sign bit-planes of significance level l , denoted as $p_{ref}(l, i)$ and $p_{sgn}(l, i)$ respectively⁸. Following from the definition of refinement bits, the refinement bit-plane of significance level l includes only coefficients that are *already* significant [61], i.e., $|X_i| \geq 2^{l+1}$. Therefore, the crossover probability of the l -th refinement bit-plane for source X_i is

$$p_{ref}(l, i) = \frac{Pr(R \cap |X_i| \geq 2^{l+1})}{Pr(|X_i| \geq 2^{l+1})} \quad (4.21)$$

where R denotes the event of crossover in magnitude bits, i.e., $R = \bigcup A_i$, with A_i defined in (4.18). Following the discussion in Section 4.4.1, we can calculate $Pr(R \cap |X_i| \geq 2^{l+1})$ by integrating the joint p.d.f. of X_i and Y_i , $f_{X_i Y_i}$, over the shaded regions in Figure 4.5(b), similar to (4.19). Moreover, $f_{X_i Y_i}$ can be factorized as in (4.20). Hence $p_{ref}(l, i)$ can be readily calculated after estimating models for $f_{X_i}(x)$ and $f_{Z_i}(z)$. In practice, we only need to integrate a few regions where $f_{X_i Y_i}$ is non-zero.

The crossover probability of sign bit-planes can be derived in a similar fashion. The difference here is we need to integrate different regions in the sample space of X_i and Y_i . The l -th sign bit-plane includes only the sign bits of the coefficients that *become* significant

⁸We introduce the subscript i in this section to facilitate the discussion of wavelet-based applications in the next section. Specifically, in Section 4.5.2, X_i will be used to denote the wavelet transform coefficient in the i -th subband. We use separate models for different subbands in order to take into account different statistics in different subbands (e.g., variances tend to decrease when going from high level subbands to low level subbands).

at significance level l [61], i.e., $2^{l+1} > |X_i| \geq 2^l$. Hence the crossover probability of the l -th sign bit-plane in the source X_i is

$$p_{sgn}(l, i) = \frac{Pr(S \cap 2^{l+1} > |X_i| \geq 2^l)}{Pr(2^{l+1} > |X_i| \geq 2^l)} \quad (4.22)$$

where S denotes the event of crossover in sign bits, i.e.,

$$S = \{ (x_i, y_i) \mid x_i > 0, y_i < 0 \} \cup \{ (x_i, y_i) \mid x_i < 0, y_i > 0 \}. \quad (4.23)$$

$Pr(S \cap 2^{l+1} > |X_i| \geq 2^l)$ can be calculated by integrating the joint p.d.f. of X_i and Y_i over the shaded regions in Figure 4.5(c), similar to (4.19), and factoring the p.d.f. as in (4.20). However, estimation of $f_{X_i}(x)$ and $f_{Z_i}(z)$ is usually not necessary, since this has already been done in refinement crossover estimation.

4.5.2 Model-based Estimation for Wavelet-based Applications

Since significance coding is used mostly for wavelet-based compressions, in this section we will discuss the particular scenarios of applying model-based estimation for wavelet-based DSC applications. We denote X the wavelet transform coefficients of the input data, with the i -th subband denoted $X_i, 0 \leq i \leq N_B - 1$, where N_B is the total number of subbands. A main issue to extend model-based approach to wavelet-based applications is that in some subbands there may not be enough coefficients to obtain reliable estimates of the model parameters (e.g., high level subbands in the case of dyadic decomposition). We will discuss how to address the issue in the following.

4.5.2.1 Estimation with Adequate Samples

Following the discussion in Section 4.5.1, to estimate the crossover probabilities of sign or refinement bit-planes, we need to estimate the models $f_{X_i}(x)$ and $f_{Z_i}(z)$. Experimental results on real image data suggest a single model $f_Z(z)$ can be used for all Z_i without any noticeable impact on coding performance. Estimation of $f_Z(z)$ (at the encoder of X) involves samples of Y and therefore is subject to communication/computational constraints, and only a small amount of samples $Z (= Y - X)$ should be used to estimate $f_Z(z)$ (as illustrated in the experiments in Section 4.6). On the other hand, estimation of $f_{X_i}(x)$ involves samples of X_i and results are affected by the number of coefficients in the i -th subband, N_i . Since some subbands may not have enough coefficients to obtain reliable estimate of $f_{X_i}(x)$, we partition the set of subbands $\{i \mid 0 \leq i \leq N_B - 1\}$ into \mathbb{L} and \mathbb{H} , where \mathbb{L} denotes the subset of subbands (low level subbands) which have enough coefficients for reliable estimation of the models, and \mathbb{H} denotes the set of remaining subbands (high level subbands). The partition of all subbands into \mathbb{L} and \mathbb{H} is determined by N_i . In particular, if Laplacian model $f_{X_i}(x) = \frac{1}{2}\beta_i e^{-\beta_i|x|}$ is chosen of X_i , and MLE is used to estimate β_i , then the MLE estimator $\hat{\beta}_i$ has a *percentage deviation* $D = (\hat{\beta}_i - \beta_i)/\beta_i$. It can be shown that $D \sim N(0, 1/N_i)$, i.e., it depends on N_i only. Therefore, we can select a threshold to apply to N_i in order to classify a subband into \mathbb{L} or \mathbb{H} according to a desired distribution of D .

Estimation of $p_{ref}(l, i)$ and $p_{sgn}(l, i)$, where $i \in \mathbb{L}$, can be performed using the algorithms discussed in Section 4.5.1, with models $f_{X_i}(x)$ and $f_Z(z)$ estimated from transform coefficients samples using standard methods, e.g. MLE. Alternatively, the correlation

noise model $f_Z(z)$ can be estimated from statistics in the raw data domain (e.g. pixel data) calculated using raw domain samples, and this can lead to some implementation advantage as transformation of the side-information is no longer required. For example, if Laplacian model $f_Z(z) = \frac{1}{2}\alpha e^{-\alpha|z|}$ is chosen for Z , then we can estimate α by calculating the standard deviation of Z , σ_Z , using raw domain samples, and using the relationship between standard deviation and model parameter in Laplacian distribution, $\alpha = \sqrt{2}/\sigma_Z$. This is viable since the variance of the correlation noise would be the same in the raw and transform domains if orthogonal filters are used. For some bi-orthogonal filters, e.g. 9/7, the variance of the correlation noise in the raw data domain would also be very close to that in the transform domain [74], and we can estimate α using similar procedures. For other bi-orthogonal filters, e.g. 5/3, the raw domain variance would need to be properly normalized, following the discussions in [74], so that α can be accurately estimated.

4.5.2.2 Estimation without Adequate Samples

Subbands in \mathbb{H} do not have enough coefficients to estimate $f_{X_i}(x)$ reliably. Instead, we use the empirical p.m.f. $Pr(X_i = x)$ of subbands in \mathbb{H} along with the correlation noise $f_Z(z)$ to estimate sign/refinement crossover probabilities. Specifically, we derive the average crossover probability for the refinement bits consisting of i -th subband coefficients, $i \in \mathbb{H}$, by

$$p_{ref}(l, i) = \sum Pr(U(l, x))Pr(X_i = x) \quad (4.24)$$

where $U(l, x)$ denotes the events of l -th refinement bits crossover when $X_i = x$, and the summation is taken over all the possible values of X_i where $Pr(X_i = x)$ is non-zero. We

can determine $Pr(X_i = x)$ empirically during encoding by binning the coefficients in the i -th subband. To determine $Pr(U(l, x))$, we assume $Y_i = x + Z$ (note that here x is a constant instead of a random variable), and $U(l, x)$ is a subset of the sample space of Z and can be found to be equal to

$$U(l, x) = \begin{cases} \{z \mid -x + (2k)2^l \leq z \leq -x + (2k + 1)2^l, \text{ or} \\ \quad -x - (2k + 1)2^l \leq z \leq -x - (2k)2^l\}, & \text{if } \lfloor \frac{|x|}{2^l} \rfloor \text{ is odd,} \\ \{z \mid -x + (2k + 1)2^l \leq z \leq -x + (2k + 2)2^l, \text{ or} \\ \quad -x - (2k + 2)2^l \leq z \leq -x - (2k + 1)2^l\}, & \text{if } \lfloor \frac{|x|}{2^l} \rfloor \text{ is even,} \end{cases} \quad (4.25)$$

where $k \in \mathbb{Z}^+$. Therefore, $Pr(U(l, x))$ can be derived by summing the integrals of $f_Z(z)$ over the shaded regions as depicted in Figure 4.7. In practice we only need to sum over a few regions where the integrals are non-zero (e.g., around $Z = 0$, if Z is Laplacian distributed). Note that computing $Pr(X_i = x)$ by binning the coefficients may not incur much complexity as the subbands in \mathbb{H} have only a small number of coefficients.

Similarly, we can derive $p_{sgn}(l, i)$, $i \in \mathbb{H}$, by

$$p_{sgn}(l, i) = \sum Pr(V(l, x))Pr(X_i = x) \quad (4.26)$$

with $V(l, x)$ denotes the event of the l -th sign bits crossover when $X_i = x$. It can be shown that $Pr(V(l, x)) = \int_{-\infty}^{-|x|} f_Z(z)dz$ when $f_Z(z)$ is symmetric. Note that we use (4.21) and (4.22) to estimate the crossover probabilities when there are enough samples in a subband to allow reliable estimation of $f_{X_i}(x)$, and (4.24) and (4.26) when there are

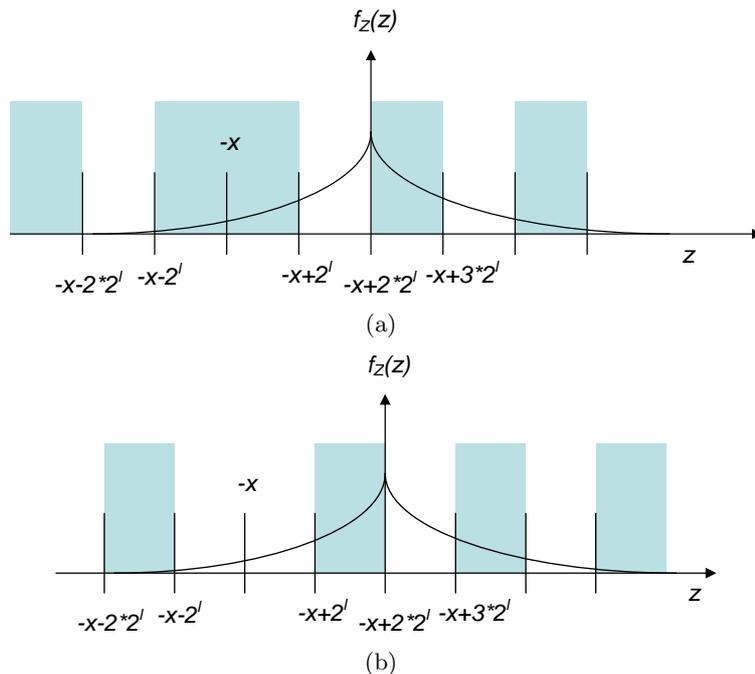


Figure 4.7: $Pr(U(l, x))$ when (a) $\lfloor \frac{|x|}{2^l} \rfloor$ is odd ; (b) $\lfloor \frac{|x|}{2^l} \rfloor$ is even.

insufficient samples in a subband and empirical p.m.f. $Pr(X_i = x)$ is used to characterize the data.

4.6 Hyperspectral Image Compression Experiments

In this section we describe several additional experiments on the proposed algorithms using real image compression applications. In particular, the DSC-based hyperspectral image compression proposed in Chapter 2 is used as the test-bed⁹ to assess the performance of the sample allocation strategy proposed in Section 4.3 and the model-based estimation proposed in Sections 4.4 and 4.5. We briefly review the system first, and present the experiment details and results in the following sections.

⁹We choose the hyperspectral image applications for experiments mainly because of the availability of the system.

4.6.1 DSC-based Hyperspectral Image Compression

Figure 4.8 depicts the encoding algorithm of the DSC-based hyperspectral image compression as proposed in Chapter 2 [8, 67]. To compress the current spectral band B_i , the sign and magnitude bits of the wavelet transform coefficients are extracted using an algorithm similar to the standard SPIHT [61], with modifications such that at some significance levels the magnitude bits are extracted as raw bit-planes (instead of separating them into significance and refinement bit-planes) in order to improve coding performance. Details of the bit-plane extraction strategy can be found in [8], while here our focus is to investigate efficient algorithms to estimate the correlation between extracted bit-planes and their corresponding SI.

Slepian-Wolf coding is employed to compress sign/refinement/raw bit-planes, using as side information the sign/refinement/raw bit-planes of same significance extracted from $a\hat{B}_{i-1} + b$, where \hat{B}_{i-1} is the previous adjacent reconstructed band available only at the decoder, and a and b are some linear prediction coefficients. Significance maps of B_i are intra-coded by zerotree coding. To determine the coding rate, the *original* previous band B_{i-1} is used to approximate \hat{B}_{i-1} at the encoder. This is viable since these applications focus on high fidelity. In particular, sign/refinement/raw bit-planes are explicitly extracted from the wavelet transform coefficients of $aB_{i-1} + b$, and the crossover probabilities are estimated by exchanging small subsets of bits and using the direct estimation approach discussed in Section 4.2. The amount of information exchanged needs to be kept small so that the algorithm can be used in parallel encoding scenarios, where each

band is assigned to a different processor, and the processors are connected by low bandwidth data buses. In order to ensure that the source and SI bit-planes are formed with the wavelet coefficients at the same locations, we need to apply the significance tree of B_i when extracting bit-planes from B_{i-1} [67]. Note that the extracted sign/refinement/raw bit-planes from B_{i-1} are solely used for correlation estimation.

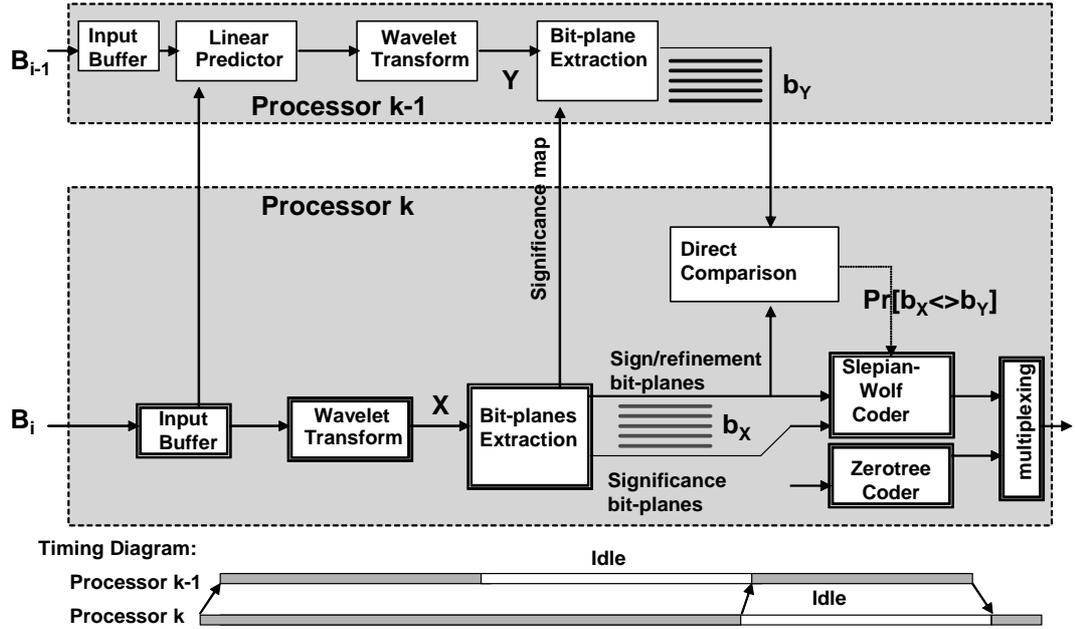


Figure 4.8: The DSC-based hyperspectral image compression with direct correlation estimation.

4.6.2 Sample Allocation Experiments

Given that n_T binary samples can be used to estimate the crossover probabilities when compressing B_i , we compare two strategies to determine how to allocate the samples to different bit-planes:

- *Adaptive sample allocation.* We use (4.14), i.e., the optimal sample allocation, to decide the numbers of samples allocated to different bit-planes. However, since

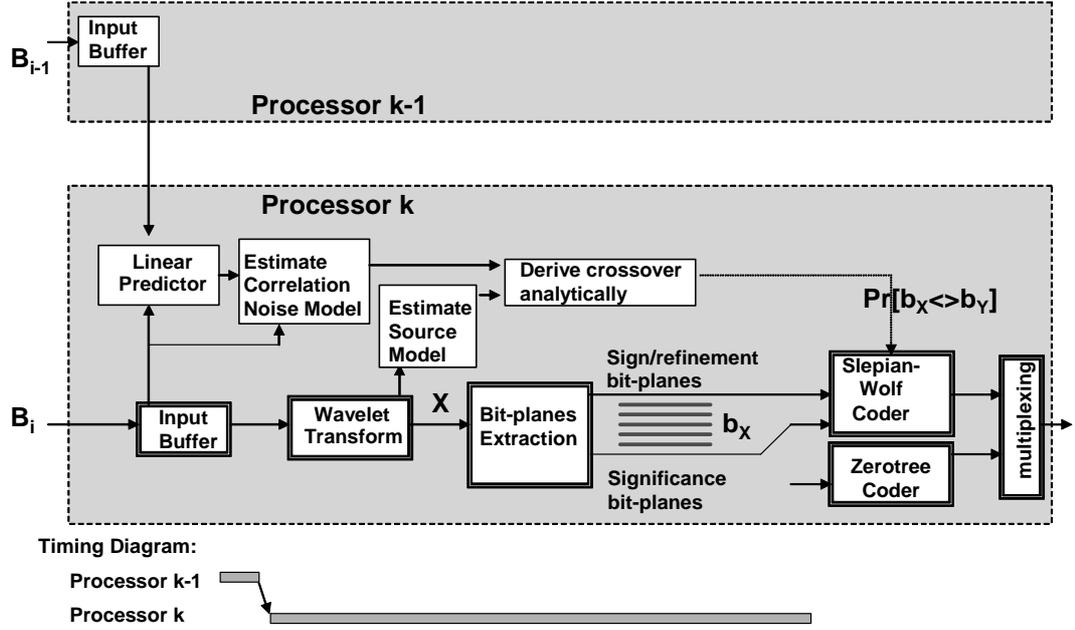


Figure 4.9: DSC-based hyperspectral image compression with model-based estimation.

the crossover probabilities of B_i are unknown, we use as a priori information the crossover probabilities of B_{i-1} in (4.14) (which have been estimated during the compression of B_{i-1}). When compressing the first DSC-coded band, since a priori information is not available we allocate the same number of samples to each bit-plane.

- *Even sample allocation.* We allocate the same number of samples to each bit-planes for all the bands.

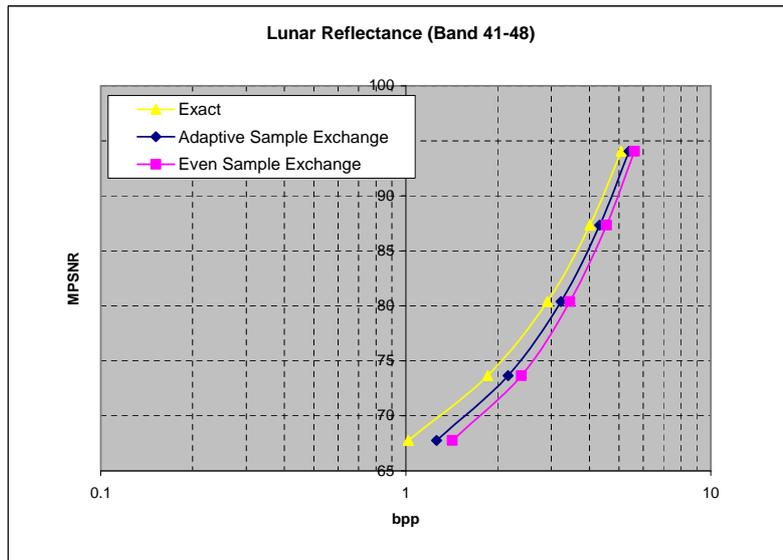
The NASA AVIRIS image data-sets [38] are used in the experiment. The original image consists of 224 spectral bands, and each spectral band consists of 614×512 16-bits pixels. In the experiment, we compress 512×512 pixels in each band. Figures 4.10(a) and 4.10(b) depict the RD performances of the system under different sample allocation strategies. Here $\text{MPSNR} = 10 \log_{10}(65535^2/\text{MSE})$, where MSE is the mean squared error

between all the original and reconstructed bands. Also shown in the figures are the RD performances when the exact crossover probabilities are used to determine the coding rate. As shown in the figures, in the situations with small number of samples exchanged (e.g., 0.25% of total), the adaptive sample allocation can reduce the rate penalty by about 1dB, as compared to the even sample allocation. Note that the adaptive sample allocation requires negligible overhead: it simply uses (4.14) to determine a more efficient sample allocation across bit-planes based on any available a priori information.

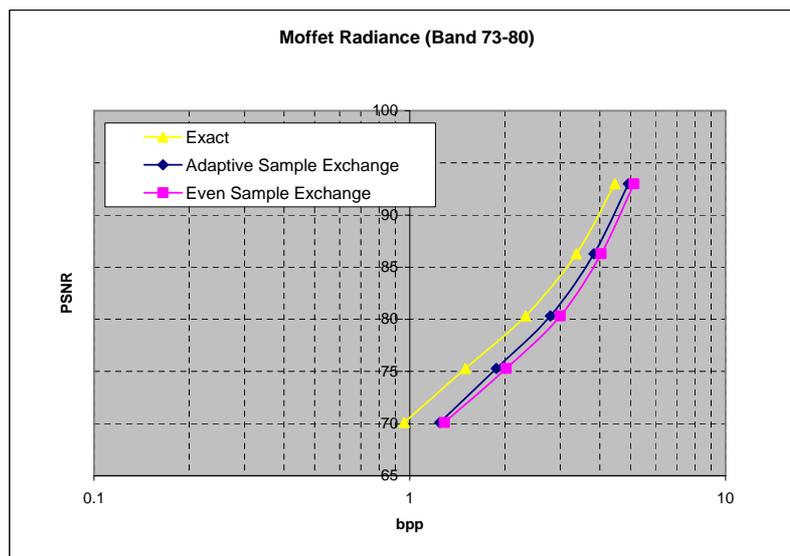
4.6.3 Model-based Estimation Experiments

Model-based estimation can be applied to the hyperspectral image system following the algorithms outlined in Sections 4.4 and 4.5, with X and Y being the transform coefficients of B_i and $aB_{i-1} + b$ respectively. As discussed, continuous-valued source samples are used to estimate the models in model-based estimation, so bit-plane extraction from SI is no longer necessary. In addition, the correlation noise model can be estimated in the pixel domain in this case following the discussion in Section 4.5.2. Therefore, the model-based estimation can result in a more efficient system as depicted in Figure 4.9. Here the core compression module is the same as that in the original system (with direct estimation) in Figure 4.8, while the correlation estimation algorithm is modified following the model-based approach, leading to the following implementation advantages in this application:

- First, the model-based system requires less computation. This is evident when comparing Figures 4.8 and 4.9: Wavelet transform on and bit-planes extraction



(a)



(b)

Figure 4.10: Sample allocation experiments - (a) *Lunar* (reflectance data), (b) *Moffet* (radiance data), using 0.25% total sample. An adaptive sample allocation scheme using the proposed optimal sample allocation strategy in Section 4.3 with a priori information from previous encoded band is compared with the even sample allocation. Here “Exact” denote the cases when the exact correlation information is used to determine the coding rate, i.e., no rate penalty.

from SI as in the original system¹⁰ are no longer required, while model estimation (using MLE) and calculating the crossover probability estimates (using analytical equations) require only small amounts of computation in the model-based system.

- Moreover, in parallel encoding scenarios, the model-based system requires less data traffic between processors. This is also evident when comparing Figures 4.8 and 4.9: In the model-based system, encoder of B_i only needs to request pixel domain samples from encoder of B_{i-1} at the beginning of processing to compute the linear prediction coefficients and correlation noise model, whereas in the original system additional traffic is incurred for exchanging significance tree and SI bit-planes.
- Furthermore, the model-based system can achieve better parallel encoding, as there are only small amounts of dependency between encoders of different bands at the beginning of processing, and processors can proceed without further synchronization (See timing diagrams in Figures 4.8 and 4.9).

The performance of model-based estimation is assessed by comparing with the original system, where SI bit-planes are explicitly extracted and *exact* crossover probabilities are used to estimate the encoding rate, i.e., there is no rate penalty. In the model-based approach, samples of B_{i-1} are obtained by downsampling the image by factors of four and eight horizontally and vertically respectively. Therefore, 3.125% of image data are used to estimate the correlation noise model. To prevent decoding error due to under-estimating the crossover probability, we allow a larger margin to determine the encoding

¹⁰It may be possible to avoid wavelet transform of SI by storing and re-using the coefficients during the compression of B_{i-1} . But this would significantly increase the memory requirements. And bit-plane extraction from SI is always required if generating SI bit-planes explicitly, since the SI sign/refinement bits need to be extracted based on the significance tree of B_i .

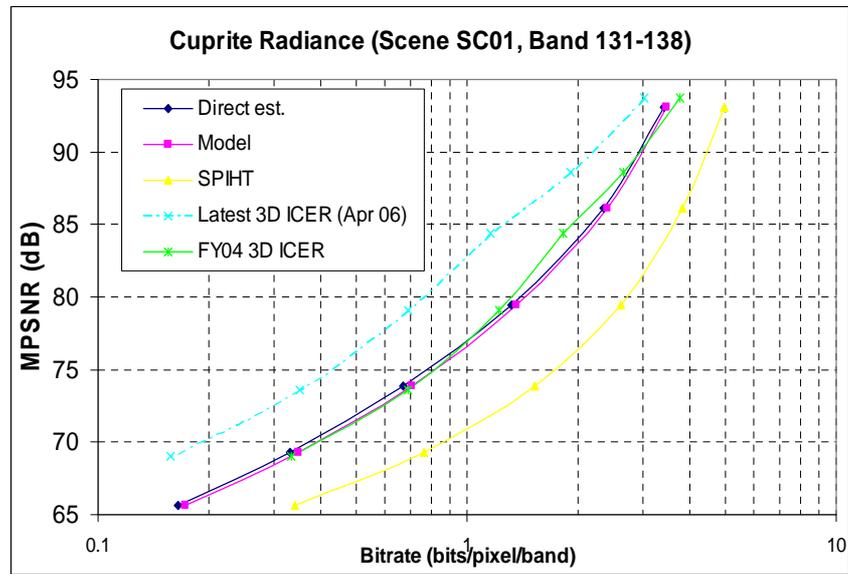
rate, at the expense of coding efficiency. Specifically, a 0.15-bit margin is added to the estimated Slepian-Wolf bound, so that there is no decoding error occurred in the testing data-sets. Figure 4.11 depicts the RD performance. As shown in the figures, model-based estimation incurs only small degradation in coding efficiency. In most cases, the difference is less than 0.5dB when compared to the direct estimation with exact crossover probabilities used to determine the coding rate.

4.7 Extensions to Other Correlation Models

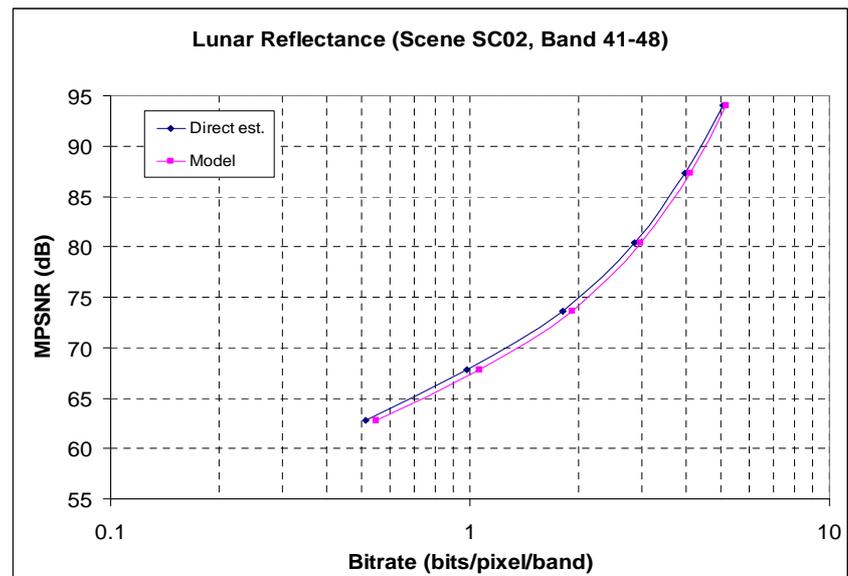
While in the previous sections we focused on binary correlation, in this section we give examples of how some of the proposed ideas can be extended for several other correlation models, where non-equiprobable inputs are considered or previously (higher significance) decoded bitplanes are used as side information for decoding.

4.7.1 Rate Penalty Model for Non-equiprobable Input

In Section 4.2 we presented a model for rate penalty $(\Delta H)(n)$ for equiprobable inputs, and experiment results in Section 4.2.4 suggest the model is sufficiently accurate for real-world data, which are in general non-equiprobable. Nevertheless, more accurate (but more complicated) penalty models can be obtained by relaxing the equiprobable assumption. We follow the notations in Section 4.2, and assume (i) $Pr[b_X = 0] = \theta$, and (ii) the correlation is symmetric, i.e., $Pr[b_Y = 1|b_X = 0] = Pr[b_Y = 0|b_X = 1] = p$. It can be



(a)



(b)

Figure 4.11: Coding efficiency comparison: (a) *Cuprite* (radiance data); (b) *Lunar* (reflectance data). The model-based system is compared with the original system, which uses all samples in direct estimation and exact crossover probabilities to determine the encoding rate, i.e. no rate penalty. Coding performances of several other wavelet systems are also presented for reference.

shown that $Pr[b_X \neq b_Y] = p$, and the lower bound in the lossless encoding rate of b_X is a function of p and θ

$$H(b_X|b_Y) = g(p, \theta) = \pi H\left(\frac{(1-p)\theta}{\pi}\right) + (1-\pi)H\left(\frac{p\theta}{1-\pi}\right), \quad (4.27)$$

where

$$\pi = Pr[b_Y = 0] = \theta + p - 2\theta p. \quad (4.28)$$

We follow the approach of Section 4.2, where the p.d.f. of the estimation error is first derived, and then the rate penalty is determined from the relationship between estimation error and rate penalty. In particular, following the estimation procedures in Section 4.2.2, the estimation error $(\Delta p)(n)$ is given by (4.5). Note that only a subset of b_Y 's are available when encoding b_X and thus $Pr[b_X \neq b_Y]$ cannot be found exactly. On the other hand, since all b_X 's are available at the encoder there is no estimation error for θ . Therefore, the rate penalty can be approximated by:

$$\begin{aligned} (\Delta H)(n) &\approx \frac{\partial g(p, \theta)}{\partial p} (\Delta p)(n) \\ &\sim N\left(\frac{\partial g(p, \theta)}{\partial p} z_{\omega/2}\sigma, \left(\frac{\partial g(p, \theta)}{\partial p}\right)^2 \sigma^2\right), \end{aligned} \quad (4.29)$$

where

$$\begin{aligned} \frac{\partial g(p, \theta)}{\partial p} &= (1-2\theta)\left(H\left(\frac{(1-p)\theta}{\pi}\right) - H\left(\frac{p\theta}{1-\pi}\right)\right) + \\ &\quad \frac{\theta(\theta-1)}{\pi} \ln\left(\frac{p(1-\theta)}{(1-p)\theta}\right) + \\ &\quad \frac{\theta(1-\theta)}{1-\pi} \ln\left(\frac{(1-p)(1-\theta)}{p\theta}\right), \end{aligned} \quad (4.30)$$

and n is the number of samples of b_Y used in the estimation. One can verify (4.29) simplifies to (4.7) when $\theta = 0.5$. Note that one can use (4.29) to derive the penalty model for multiple non-equiprobable binary sources and determine the corresponding optimum bit allocation following the discussion in Section 4.3.

4.7.2 Model-based Estimation with Previously Decoded Bit-planes as Side-information

We now discuss how to extend model-based estimation to the cases when the previously decoded bit-planes are used as SI. Specifically, we consider the cases when bit-planes are extracted from a continuous input source, and each bit-plane $b_X(l)$ is compressed with both the previously decoded bit-planes of the same source $b_X(l+1), \dots, b_X(l+m)$ and that of the correlated source $b_Y(l), b_Y(l+1), \dots, b_Y(l+m)$ as SI (Figure 4.12(a)). We assume bit-planes are communicated to the decoder starting from the MSB, while the reverse order can be addressed similarly. For encoding $b_X(l)$ we need to estimate the coding rate

$$H(b_X(l)|b_X(l+1), \dots, b_X(l+m), b_Y(l), b_Y(l+1), \dots, b_Y(l+m)). \quad (4.31)$$

We follow the same general approach as discussed in Section 4.4. To determine (4.31), we need the joint p.d.f. between the input and all the SI:

$$p(b_X(l), b_X(l+1), \dots, b_X(l+m), b_Y(l), b_Y(l+1), \dots, b_Y(l+m)), \quad (4.32)$$

which has $2^{(2m+2)} - 1$ free parameters. It may seem complicated to estimate the joint p.d.f. However, it turns out the model-based estimation for (4.32) exhibits regular structure,

which greatly simplifies the estimation process. In addition, some video data suggests further improvement could be negligible with $m > 1$ in practical applications [10, 11] (Figure 4.13).

We denote $\gamma_{i,j}$ the joint probability when the binary representation of i is $b_X(l+m)\dots b_X(l+1)b_X(l)$ and that of j is $b_Y(l+m)\dots b_Y(l+1)b_Y(l)$ (Figure 4.12(b)), i.e.,

$$\gamma_{i,j} = p(\langle b_X(l+m)\dots b_X(l+1)b_X(l) \rangle = i, \langle b_Y(l+m)\dots b_Y(l+1)b_Y(l) \rangle = j),$$

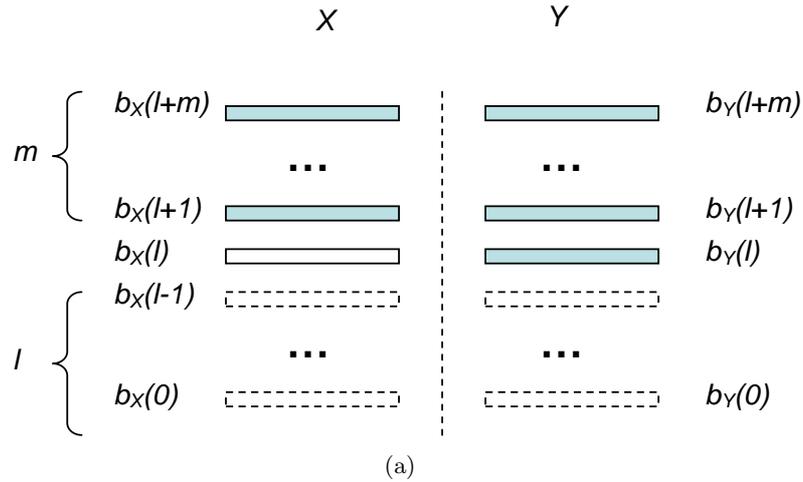
and $\langle b(l+m)\dots b(l+1)b(l) \rangle$ denotes the numerical value of the concatenation of the sequence of the bits $b(l+m), \dots, b(l+1), b(l)$, i.e., $\sum_{i=0}^m b(l+i) \times 2^i$. It can be shown that by tracing the binary representations of X and Y the events leading to the occurrence of $\langle b_X(l+m)\dots b_X(l+1)b_X(l) \rangle = i$ and $\langle b_Y(l+m)\dots b_Y(l+1)b_Y(l) \rangle = j$ correspond to the region $A_{i,j}$ in the sample space of X and Y in Figure 4.14. Therefore,

$$\gamma_{i,j} = \int \int_{A_{i,j}} f_{XY}(x,y) dx dy \quad (4.33)$$

where

$$A_{i,j} = \{ (x,y) \mid c \cdot 2^{l+m+1} + i \cdot 2^l \leq |x| \leq c \cdot 2^{l+m+1} + i \cdot 2^l + 2^l - 1, \\ d \cdot 2^{l+m+1} + j \cdot 2^l \leq |y| \leq d \cdot 2^{l+m+1} + j \cdot 2^l + 2^l - 1, c, d \in \mathbb{Z} \} \quad (4.34)$$

(4.33) can be readily computed by factorizing f_{XY} and estimating f_X and f_Y as discussed in Section 4.4. In practice, we only need to sum over a few regions where the integrals of f_{XY} are practically non-zero. Note that we can extend this to estimate the encoding



(a)

	$b_X(l+m) \dots b_X(l+1) b_X(l)$		
$b_Y(l+m) \dots b_Y(l+1) b_Y(l)$	00...00	00...01	... 11...11
00...00	$Y_{0,0}$	$Y_{1,0}$	$Y_{2^{m+1}-1,0}$
00...01	$Y_{0,1}$...	
00...10	$Y_{0,2}$		
...	...		
11...11	$Y_{0,2^{m+1}-1}$		$Y_{2^{m+1}-1,2^{m+1}-1}$

(b)

Figure 4.12: (a) Encoding of bit-plane $b_X(l)$ with both the previously decoded bit-planes $b_X(l+1), \dots, b_X(l+m)$ and that of the correlated source $b_Y(l), b_Y(l+1), \dots, b_Y(l+m)$ as SI. (b) Joint p.d.f. between the input and all the SI.

rate for structured bit-planes (i.e., sign/refinement bit-planes) following the discussion in Section 4.5.

4.7.3 Model-based Estimation with Continuous Side Information in Joint Decoding

In this section we consider situations where each bit-plane $b_X(l)$ is compressed with both the previously decoded bit-planes of the same source $b_X(l+1), \dots, b_X(l+m)$ and the

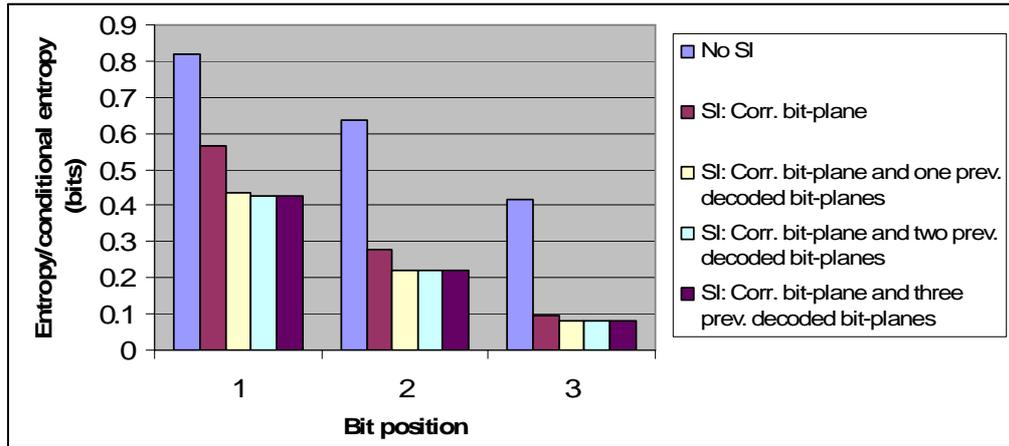


Figure 4.13: Entropy/conditional entropy of the video bit-planes data used in the experiment in Section 4.2.4, i.e., X and Y are the quantized DCT coefficients in a current frame and the corresponding quantized coefficients in the motion-compensated predictors in the reference frame, respectively, using the 2nd AC coefficients of *Mobile* (720×576 , 30 fps), at QP= 12: (a) Without SI, i.e., intra coding (“No SI”); (b) Using only the corresponding bit-plane as SI, i.e., $m = 0$ (“SI: Corr. bit-plane”); (c), (d), (e): Using corresponding and one, two or three previously decoded bit-planes as SI, i.e., $m = 1, 2$ or 3 respectively. The results suggest further improvements with $m > 1$ could be negligible.

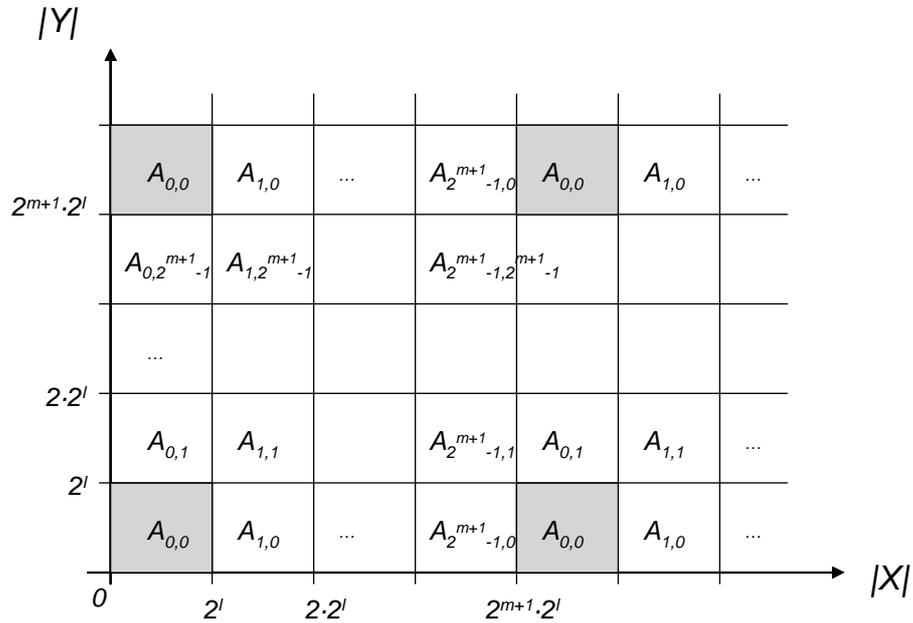


Figure 4.14: Extension of model based estimation: The events leading to the occurrence of $\langle b_X(l+m) \dots b_X(l+1) b_X(l) \rangle = i$ and $\langle b_Y(l+m) \dots b_Y(l+1) b_Y(l) \rangle = j$ correspond to the region $A_{i,j}$ in the sample space of X and Y .

continuous SI Y available to be used in joint decoding¹¹. For encoding $b_X(l)$ we need to estimate the coding rate

$$H(b_X(l)|b_X(l+1), \dots, b_X(l+m), Y). \quad (4.35)$$

To determine (4.35), we need the joint p.d.f.:

$$p(b_X(l), b_X(l+1), \dots, b_X(l+m), Y). \quad (4.36)$$

We denote $p(\langle b_X(l+m)\dots b_X(l+1)b_X(l) \rangle = i, Y = y)$ by $\gamma_i(y)$. Following the discussion in Section 4.7.2 it can be shown that

$$\gamma_i(y) = \int_{A_i} f_{X,Y}(x, y) dx \quad (4.37)$$

where A_i are subsets of X (Figure 4.15):

$$A_i = \{ x \mid c \cdot 2^{l+m+1} + i \cdot 2^l \leq |x| \leq c \cdot 2^{l+m+1} + i \cdot 2^l + 2^l - 1, c \in \mathbb{Z}^+ \}. \quad (4.38)$$

4.8 Conclusions

In this paper, we have investigated correlation estimation for distributed image and video applications under rate and complexity constraints. Focusing on the situations when

¹¹Note that while this model is commonly used in distributed video applications (e.g., [3]), many lossy image compression applications may not communicate the LSB bit-planes and therefore a full resolution version of Y would not be available to be used for joint decoding.

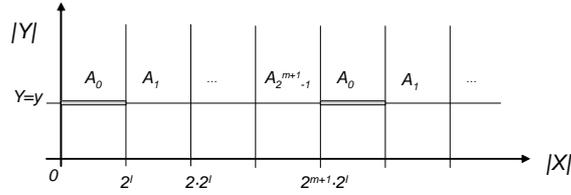


Figure 4.15: The events leading to the occurrence of $\langle b_X(l+m) \dots b_X(l+1)b_X(l) \rangle = i$ correspond to regions A_i , subset of X .

sampling techniques are employed to estimate the correlation, we first analyzed how the number of samples relates to the p.d.f. of the rate penalty when compressing a binary input source. The rate penalty was found to be normally distributed with parameters depending on the number of samples and the crossover probability between the source and SI. We then extended the analysis to the cases when multiple binary input sources are to be compressed and proposed a strategy to allocate samples to the sources such that the overall rate penalty can be minimized. Furthermore, We proposed a model-based estimation for the particular but important situations when bit-planes are extracted from a continuous-valued input source, and discussed extensions to the cases when bit-planes are extracted based on the significance of the data for wavelet-based DSC applications. Experiment results including real image compression demonstrated model-based approach can achieve accurate estimation. In addition, the model-based estimation might lead to some implementation advantages.

Chapter 5

Conclusions and Future Work

We have proposed in this thesis new video and image coding algorithms based on DSC to address some of the issues in conventional compression framework.

First, we discuss a wavelet-based hyperspectral image compression algorithm that combines set partitioning with our proposed DSC techniques to achieve competitive coding performance. DSC tools allow encoding to proceed in “open loop”, and this facilitates parallel compression of spectral bands in multi-processors architectures. We also discuss a coding strategy that adaptively applies DSC or intra coding to bit-planes according to the statistics of the data, to maximize the coding gain. Experimental results suggest our scheme is comparable to a simple 3-D wavelet codec developed by NASA-JPL in terms of compression performance.

Moreover, we propose to address flexible video decoding using a DSC approach. With DSC, the overhead to support flexible decoding depends on the worst case correlation noise rather than the number of possible decoding paths. As a result, DSC can lead to a solution that compares favorably to conventional approaches. Experimental results

using multiview video coding and forward/backward video playback demonstrate the improvement.

Furthermore, we study the correlation estimation problem in DSC. We propose a model to characterize the coding rate penalty due to estimation error, and based on the model we propose a sample allocation algorithm to minimize the overall rate penalty in the multiple sources scenarios. We also propose a model-based estimation for distributed coding of continuous-valued input sources. We demonstrate the effectiveness of the proposed algorithms by experimental results including some based on real image data.

Some related future research topics could be

- *Peer-to-peer multicast streaming based on DSC.* Conventional compression tools may lack the robustness to address peer-to-peer (P2P) multicast video streaming. In P2P networks, peer nodes serving the video data may disconnect at any time. Moreover, individual peer nodes may support different and time-varying upstream data rates. Conventional compression algorithms fail to cope with these operating conditions since they are vulnerable to data loss and delay variation. In Chapter 3, we demonstrate that with our proposed algorithms it is possible to generate a single compressed bitstream that can be decoded in several different ways under different operating conditions. This enhanced adaptability can, in principle, greatly facilitate P2P multicast streaming.
- *Application-specific hyperspectral image compression.* Existing hyperspectral image compression algorithms are mostly optimized for rate-distortion. However, one

of the main applications of hyperspectral image data is to identify the ground objects. Therefore, it is important to investigate new hyperspectral image compression techniques that optimally preserve the spectral signatures at a given rate, so as to achieve the best classification performance.

Appendix A

Derivation of Bounds on the Estimation Error

Here we justify the bounds on the estimation error in Section 2.4.1.2. We let the crossover probability estimator be the upper bound of the $(1 - \omega) \times 100\%$ confidence interval for a population proportion, i.e.,

$$\hat{p}_i = \frac{s_i}{n_i} + z_{\omega/2} \sqrt{p_i(1 - p_i)/n_i} \approx \frac{s_i}{n_i} + z_{\omega/2} \sqrt{\frac{s_i}{n_i} \left(1 - \frac{s_i}{n_i}\right) / n_i}$$

Let $m = z_{\omega/2} \sqrt{p_i(1 - p_i)/n_i}$. By the definition of confidence interval, we have

$$Pr \left(\frac{s_i}{n_i} - m \leq p_i \leq \frac{s_i}{n_i} + m \right) = 1 - \omega$$

Equivalently,

$$Pr \left(p_i - m \leq \frac{s_i}{n_i} \leq p_i + m \right) = 1 - \omega$$

By this and the fact that $\frac{s_i}{n_i}$ can be approximated by a Normal density with mean p_i and variance $p_i(1 - p_i)/n_i$, we have

$$\begin{aligned} Pr\left(\frac{s_i}{n_i} < p_i - m\right) &= \omega/2 \\ Pr\left(\frac{s_i}{n_i} + m - p_i < 0\right) &= \omega/2 \\ Pr(\hat{p}_i - p_i < 0) &= \omega/2 \end{aligned}$$

and

$$\begin{aligned} Pr\left(\frac{s_i}{n_i} > p_i + m\right) &= \omega/2 \\ Pr\left(\frac{s_i}{n_i} + m - p_i > 2m\right) &= \omega/2 \\ Pr(\hat{p}_i - p_i > 2m) &= \omega/2 \end{aligned}$$

From these equations, the probability of decoding error and probability of large encoding rate penalty can be estimated.

Appendix B

Derivation of Optimal Sample Allocation

Here we give the detail of the derivation of the optimal sample allocation given in (4.14).

We consider

$$\min_{\{n_l: \sum_{l=0}^{L-1} n_l = n_T; n_l \leq K_l\}} E[\Delta H], \quad (\text{B.1})$$

where $E[\Delta H] = \frac{1}{K_T} \sum_{l=0}^{L-1} K_l \alpha_l n_l^{-1/2}$ and $\alpha_l = \ln(\frac{1}{p_l} - 1) z_{\omega/2} \sqrt{p_l(1-p_l)}$. Using Lagrange multipliers, we construct

$$J(\mathbf{n}) = \frac{1}{K_T} \sum_{l=0}^{L-1} K_l \alpha_l n_l^{-1/2} + \lambda \sum_{l=0}^{L-1} n_l. \quad (\text{B.2})$$

Differentiate (B.2) and set to zero,

$$\frac{\partial J}{\partial n_l} = \frac{-1}{2K_T} K_l \alpha_l n_l^{-3/2} + \lambda = 0, \quad (\text{B.3})$$

or

$$n_l^* = (K_l \alpha_l)^{2/3} \left(\frac{1}{2\lambda K_T} \right)^{2/3} = (K_l \alpha_l)^{2/3} \gamma. \quad (\text{B.4})$$

This is true as long as $n_l^* < K_l$. Otherwise, we need to use the Kuhn-Tucker conditions to find n_l^* :

$$\frac{\partial J}{\partial n_l} \begin{cases} = 0, & \text{if } n_l^* < K_l, \\ \leq 0, & \text{if } n_l^* = K_l. \end{cases} \quad (\text{B.5})$$

Solving (B.5) we obtain the results in (4.14).

Bibliography

- [1] A. Aaron and B. Girod. Compression with side information using turbo codes. In *Proc. Data Compression Conference (DCC)*, 2002.
- [2] A. Aaron, P. Ramanathan, and B. Girod. Wyner-Ziv coding of light fields for random access. In *Proc. Workshop on Multimedia Signal Processing (MMSP)*, 2004.
- [3] A. Aaron, S. Rane, and B. Girod. Wyner-Ziv video coding with hash-based motion compensation at the receiver. In *Proc. Int'l Conf. Image Processing (ICIP)*, 2004.
- [4] A. Aaron, R. Zhang, and B. Girod. Wyner-Ziv coding of motion video. In *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 2002.
- [5] J. Bajcsy and P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. In *Proc. Global Telecommunications Conference (GLOBECOM)*, 2001.
- [6] M. Barni, D. Papini, A. Abrardo, and E. Magli. Distributed source coding of hyperspectral images. In *Proc. Int'l Symp. Geoscience and Remote Sensing (IGARSS)*, 2005.
- [7] A. R. Calderbank, I. Daubechies, W. Sweldens, and D. Yeo. Wavelet transforms that map integers to integers. *Applied Comp. Harmonic Analy.*, 5:332–369, 1998.
- [8] N.-M. Cheung and A. Ortega. An efficient and highly parallel hyperspectral imagery compression scheme based on distributed source coding. In *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2006.
- [9] N.-M. Cheung and A. Ortega. A model-based approach to correlation estimation in wavelet-based distributed source coding with application to hyperspectral imagery. In *Proc. Int'l Conf. Image Processing (ICIP)*, 2006.
- [10] N.-M. Cheung and A. Ortega. Distributed source coding application to low-delay free viewpoint switching in multiview video compression. In *Proc. Picture Coding Symposium (PCS)*, 2007.
- [11] N.-M. Cheung and A. Ortega. Flexible video decoding: A distributed source coding approach. In *Proc. Workshop on Multimedia Signal Processing (MMSP)*, 2007.
- [12] N.-M. Cheung and A. Ortega. Compression algorithms for flexible video decoding. In *Proc. Visual Communications and Image Processing (VCIP)*, 2008.

- [13] N.-M. Cheung and A. Ortega. Flexible video decoding: A distributed source coding approach. *IEEE Transactions on Image Processing*. In preparation, 2008.
- [14] N.-M. Cheung, C. Tang, A. Ortega, and C. S. Raghavendra. Efficient wavelet-based predictive Slepian-Wolf coding for hyperspectral imagery. *EURASIP Journal on Signal Processing - Special Issue on Distributed Source Coding*, Nov. 2006.
- [15] N.-M. Cheung, H. Wang, and A. Ortega. Correlation estimation for distributed source coding under information exchange constraints. In *Proc. Int'l Conf. Image Processing (ICIP)*, 2005.
- [16] N.-M. Cheung, H. Wang, and A. Ortega. Video compression with flexible playback order based on distributed source coding. In *Proc. Visual Communications and Image Processing (VCIP)*, 2006.
- [17] N.-M. Cheung, H. Wang, and A. Ortega. Sampling-based correlation estimation for distributed source coding under rate and complexity constraints. *IEEE Transactions on Image Processing*. Under review, 2007.
- [18] R. N. Clark, G. Swayze, J. Boardman, and F. Kruse. Comparison of three methods for materials identification and mapping with imaging spectroscopy. In *JPL AVIRIS Airborne Geoscience Workshop*, 1993.
- [19] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. New York: Wiley, 1991.
- [20] I. Csiszar. Linear codes for sources and source networks: Error exponents, universal coding. *IEEE Trans. Information Theory*, 28(4):585–592, July 1982.
- [21] I. Csiszar and J. Korner. Towards a general theory of source networks. *IEEE Trans. Information Theory*, 26(2):155–165, Mar. 1980.
- [22] S. Draper and E. Martinian. Compound conditional source coding, Slepian-Wolf list decoding, and applications to media coding. In *Proc. Int'l Symp. Information Theory (ISIT)*, 2007.
- [23] J. E. Fowler. An open source software library for quantization, compression and coding. In *Proc. SPIE Applications of Digital Image Processing XXIII*, pages 294–301, Aug 2000.
- [24] J. E. Fowler. QccPack: Quantization, Compression, Coding Library and Utilities. <http://qccpack.sourceforge.net>, 2006.
- [25] C.-H. Fu, Y.-L. Chan, and W.-C. Siu. Efficient reverse-play algorithms for MPEG video with VCR support. *IEEE Trans. Circuits and Systems for Video Technology*, 16(1), Jan. 2006.
- [26] M. Fumagalli, M. Tagliasacchi, and S. Tubaro. Improved bit allocation in an error-resilient scheme based on distributed source coding. In *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2006.

- [27] J. Garcia-Frias and Y. Zhao. Compression of correlated binary sources using turbo codes. *IEEE Communications Letters*, pages 417–419, Oct. 2001.
- [28] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery*, 93(1):71–83, Jan. 2005.
- [29] M. Grangetto, E. Magli, and G. Olmo. Distributed arithmetic coding. *IEEE Communications Letters*, 11(11), Nov. 2007.
- [30] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann. Distributed monoview and multiview video coding. *IEEE Signal Processing Magazine*, 24, Sept. 2007.
- [31] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li. Distributed multi-view video coding. In *Proc. Visual Communications and Image Processing (VCIP)*, 2006.
- [32] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li. Free viewpoint switching in multi-view video streaming using Wyner-Ziv video coding. In *Proc. Visual Communications and Image Processing (VCIP)*, 2006.
- [33] Z. He, L. Cao, and H. Cheng. Correlation estimation and performance optimization for distributed image compression. In *Proc. Visual Communications and Image Processing (VCIP)*, 2006.
- [34] Z. He and X. Chen. Correlation structure analysis for distributed video compression over wireless video sensor networks. In *Proc. Visual Communications and Image Processing (VCIP)*, 2006.
- [35] ISO/IEC JTC1/SC29/WG11. Introduction to multiview video coding. Technical Report N7328, MPEG document, 2005.
- [36] A. Jagmohan, A. Sehgal, and N. Ahuja. Compressed of lightfield rendered images using coset codes. In *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2003.
- [37] M. Karczewicz and R. Kurceren. The SP- and SI-frames design for H.264/AVC. *IEEE Trans. Circuits and Systems for Video Technology*, 13, July 2003.
- [38] A. Kiely, M. Klimesh, H. Xie, and N. Aranki. ICER-3D: A progressive wavelet-based compressor for hyperspectral images. Technical report, NASA-JPL IPN Progress Report, 2006. http://ipnpr.jpl.nasa.gov/progress_report/42-164/164A.pdf.
- [39] M. Klimesh, A. Kiely, H. Xie, and N. Aranki. Spectral ringing artifacts in hyperspectral image data compression. Technical report, NASA-JPL IPN Progress Report, 2005. http://ipnpr.jpl.nasa.gov/progress_report/42-160/160C.pdf.
- [40] D. Landgrebe. Hyperspectral image data analysis. *IEEE Signal Processing Magazine*, 19, Jan 2002.

- [41] P. Lassila, J. Karvo, and J. Virtamo. Efficient importance sampling for monte carlo simulation of multicast networks. In *Proc. Conf. Computer Communications (INFOCOM)*, 2001.
- [42] C. W. Lin, J. Zhou, J. Youn, and M. T. Sun. MPEG video streaming with VCR functionality. *IEEE Trans. Circuits and Systems for Video Technology*, 11(3), Mar. 2001.
- [43] A. Liveris, Z. Xiong, and C. Georghiades. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Communications Letters*, 6, Oct. 2002.
- [44] A. Liveris, Z. Xiong, and C. Georghiades. A distributed source coding technique for correlated images using turbo-codes. *IEEE Communications Letters*, 2002.
- [45] A. D. Liveris, C.-F. Lan, K. Narayanan, Z. Xiong, and C. N. Georghiades. Slepian-Wolf coding of three binary sources using LDPC codes. In *Proc. Intl. Symp. Turbo Codes and Related Topics*, Sept. 2003.
- [46] D. J. C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Information Theory*, pages 399–431, Mar. 1999.
- [47] E. Magli, M. Barni, A. Abrardo, and M. Grangetto. Distributed source coding techniques for lossless compression of hyperspectral images. *EURASIP Journal on Applied Signal Processing*, 2007, 2007.
- [48] A. Majumdar, J. Chou, and K. Ramchandran. Robust distributed video compression based on multilevel coset codes. In *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2003.
- [49] A. Majumdar, R. Puri, P. Ishwar, and K. Ramchandran. Complexity/performance trade-offs for robust distributed video coding. In *Proc. Int'l Conf. Image Processing (ICIP)*, 2005.
- [50] W. Mendenhall and T. Sincich. *Statistics for Engineering and the Sciences*. Prentice-Hall, 1995.
- [51] A. C. Miguel, A. R. Askew, A. Chang, S. Hauck, and R. E. Ladner. Reduced complexity wavelet-based predictive coding of hyperspectral images for FPGA implementation. In *Proc. Data Compression Conference (DCC)*, 2004.
- [52] A. Naman and D. Taubman. A novel paradigm for optimized scalable video transmission based on JPEG2000 with motion. In *Proc. Int'l Conf. Image Processing (ICIP)*, 2007.
- [53] A. Ortega. Video coding: Predictions are hard to make, especially about the future. In *Proc. Image Media Processing Symposium*, 2007.
- [54] M. Pedram and E. J. M. Rabaey. *Power Aware Design Methodologies*. Kluwer Academic Publishers, 2002.

- [55] S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Magazine*, pages 51–60, Mar. 2002.
- [56] S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. In *Proc. Data Compression Conference (DCC)*, 1999.
- [57] S. Pradhan and K. Ramchandran. Distributed source coding: symmetric rates and applications to sensor networks. In *Proc. Data Compression Conference (DCC)*, 2000.
- [58] S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS). *IEEE Trans. Information Theory*, 49(3):626–643, Mar. 2003.
- [59] R. Puri and K. Ramchandran. PRISM: a new robust video coding architecture based on distributed compression principles. In *Proc. Allerton Conf. Communications, Control, and Computing*, Oct. 2002.
- [60] S. Rane and B. Girod. Systematic lossy error protection based on H.264/AVC redundant slices. In *Proc. Visual Communications and Image Processing (VCIP)*, Jan. 2006.
- [61] A. Said and W. A. Pearlman. A new, fast, and efficient image codec using set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, pages 243–250, June 1996.
- [62] A. Sehgal, A. Jagmohan, and N. Ahuja. Wyner-Ziv coding of video: an error-resilient compression framework. *IEEE Trans. Multimedia*, 6(2):249–258, Apr. 2004.
- [63] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Information Theory*, 19:471–480, July 1973.
- [64] N. T. Slingerland and A. J. Smith. Cache performance for multimedia applications. In *Proc. Int'l Conf. Supercomputing*, 2001.
- [65] S. R. Smoot and L. A. Rowe. Study of DCT coefficient distributions. In *Proc. SPIE*, Jan. 1996.
- [66] E. Steinbach, N. Farber, and B. Girod. Standard compatible extension of H.263 for robust video transmission in mobile environments. *IEEE Trans. Circuits and Systems for Video Technology*, 7, Dec. 1997.
- [67] C. Tang, N.-M. Cheung, A. Ortega, and C. S. Raghavendra. Efficient inter-band prediction and wavelet based compression for hyperspectral imagery: A distributed source coding approach. In *Proc. Data Compression Conference (DCC)*, 2005.
- [68] C. Tang and C. S. Raghavendra. Bitplane coding for correlations exploitation in wireless sensor networks. In *Proc. Int'l Conf. Communications (ICC)*, Seoul, May 2005.

- [69] X. Tang, W. A. Pearlman, and J. W. Modestino. Hyperspectral image compression using three-dimensional wavelet coding: A lossy-to-lossless solution. submitted to *IEEE Transactions on Geoscience and Remote Sensing*.
- [70] M. Tanimoto. FTV (free viewpoint television) creating ray-based image engineering. In *Proc. Int'l Conf. Image Processing (ICIP)*, 2005.
- [71] D. Taubman and M. Marcellin. *JPEG2000: Image compression fundamentals, standards and practice*. Kluwer, 2002.
- [72] V. Thirumalai, I. Tomic, and P. Frossard. Distributed coding of multiresolution omnidirectional images. In *Proc. Int'l Conf. Image Processing (ICIP)*, 2007.
- [73] J. Tian. Software for MATLAB SPIHT codes. Available at MATLAB Central <http://www.mathworks.com/matlabcentral/>.
- [74] B. Usevitch. Optimal bit allocation for biorthogonal wavelet coding. In *Proc. Data Compression Conference (DCC)*, 1996.
- [75] M. van der Schaar and P. Chou. *Multimedia over IP and Wireless Networks: Compression, Networking, and Systems*. Academic Press, 2007.
- [76] H. Wang, N.-M. Cheung, and A. Ortega. WZS: Wyner-Ziv scalable predictive video coding. In *Proc. Picture Coding Symposium (PCS)*, Dec. 2004.
- [77] H. Wang, N.-M. Cheung, and A. Ortega. A framework for adaptive scalable video coding using Wyner-Ziv techniques. *EURASIP Journal on Applied Signal Processing*, 2006.
- [78] J. Wang, A. Majumdar, K. Ramchandran, and H. Garudadri. Robust video transmission over a lossy network using a distributed source coded auxiliary channel. In *Proc. Picture Coding Symposium (PCS)*, 2004.
- [79] J. Wang, V. Prabhakaran, and K. Ramchandran. Syndrome-based robust video transmission over networks with bursty losses. In *Proc. Int'l Conf. Image Processing (ICIP)*, 2006.
- [80] Y. Wang, J. Ostermann, and Y. Zhang. *Video Processing and Communication*. Prentice Hall, New Jersey, 2002.
- [81] S. J. Wee and B. Vasudev. Compressed-domain reverse play of MPEG video streams. In *Proc. Multimedia Systems and Applications*, 1999.
- [82] T. Wiegand. Joint final committee draft for joint video specification H.264. Technical Report JVT-D157, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, 2002.
- [83] A. Wyner. Recent results in the Shannon theory. *IEEE Trans. Information Theory*, 20(1):2–10, 1974.

- [84] A. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Information Theory*, 22:1–10, Jan. 1976.
- [85] Z. Xiong, A. Liveris, and S. Cheng. Distributed source coding for sensor networks. *IEEE Signal Processing Magazine*, 21:80–94, Sept. 2004.
- [86] Q. Xu and Z. Xiong. Layered Wyner-Ziv video coding. In *Proc. Visual Communications and Image Processing (VCIP)*, Jan. 2004.
- [87] X. Zhou, W.-Y. Kung, and C.-C. J. Kuo. A robust H.264 video streaming scheme for portable devices. In *Proc. Int'l Symp. Circuits and Systems*, 2005.
- [88] X. Zhu, A. Aaron, and B. Girod. Distributed compression for large camera arrays. In *Proc. Workshop on Statistical Signal Processing*, 2003.