



Vrije Universiteit Brussel

FACULTY OF ENGINEERING
Department of Electronics and Informatics

Scalable Single and Multiple Description Scalar Quantization

Thesis submitted in fulfillment of the requirements for the award of the
degree of Doctor in Engineering (Doctor in de Ingenieurswetenschappen)
by

ir. Shahid Mahmood Satti

July 2012

Advisors: Prof. dr. ir. Adrian Munteanu
 Prof. dr. ir. Peter Schelkens



Examining Committee

Prof. dr. ir. Adrian Munteanu – Vrije Universiteit Brussel – Promoter

Prof. dr. Ir. Peter Schelkens – Vrije Universiteit Brussel – Promoter

Prof. dr. ir. Leo Van Biesen – Vrije Universiteit Brussel – Committee Chair

Prof. dr. ir. Rik Pintelon – Vrije Universiteit Brussel – Committee Vice-chair

Prof. dr. ir. Jan Lemeire – Vrije Universiteit Brussel – Committee Secretary

Dr. ir. Joeri Barbarien – Technicolor – Member

Dr. ir. Fabio Verdicchio – University of Aberdeen – Member

Prof. dr. ir. Jan Cornelis – Vrije Universiteit Brussel – Member

Print: Silhouet, Maldegem

© 2012 Shahid Satti

2012 Uitgeverij VUBPRESS Brussels University Press
VUBPRESS is an imprint of ASP nv (Academic and Scientific Publishers nv)
Ravensteingalerij 28
B-1000 Brussels
Tel. +32 (0)2 289 26 50
Fax +32 (0)2 289 26 59
E-mail: info@vubpress.be
www.vubpress.be

ISBN 978 90 5718 143 6
NUR 950/980
Legal deposit D/2012/11.161/079

All rights reserved. No parts of this book may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author.

*Dedicated to the beautiful country,
Pakistan.*

TABLE OF CONTENTS

| | |
|---|------------|
| ACKNOWLEDGMENTS | V |
| ABSTRACT | VII |
| ACRONYMS | IX |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 QUANTIZATION | 1 |
| 1.2 SCALABLE CODING USING QUANTIZATION | 2 |
| 1.3 SOURCE SCALABILITY AND ERROR-RESILIENCE | 4 |
| 1.4 MAJOR CONTRIBUTIONS AND THESIS ORGANIZATION | 6 |
| CHAPTER 2 INFORMATION THEORY AND SOURCE CODING | 9 |
| 2.1 INTRODUCTION | 9 |
| 2.2 DEFINITIONS | 10 |
| 2.2.1 Entropy | 10 |
| 2.2.2 Conditional Entropy | 10 |
| 2.2.3 Joint Entropy..... | 11 |
| 2.2.4 Relative Entropy | 12 |
| 2.2.5 Mutual Information..... | 12 |
| 2.3 LOSSLESS SOURCE CODING | 13 |
| 2.3.1 Kraft's Inequality..... | 14 |
| 2.3.2 Lower Bound on Lossless Source Coding | 14 |
| 2.3.3 Shannon Coding | 15 |
| 2.3.4 Shannon-Fano Coding | 16 |
| 2.3.5 Huffman Coding | 17 |
| 2.3.6 Arithmetic Coding | 19 |
| 2.4 LOSSY SOURCE CODING | 23 |
| 2.4.1 Rate-Distortion Theory..... | 23 |
| 2.4.2 Practical Quantization Methods..... | 26 |
| 2.4.3 Embedded Quantization..... | 32 |
| 2.5 WAVELET-BASED SCALABLE IMAGE CODING | 36 |
| 2.5.1 Transforms for Image Compression | 36 |
| 2.5.2 Continuous-time Wavelet Transform | 37 |
| 2.5.3 Discrete Wavelet Transform (DWT) and Multiresolution Analysis.. | 37 |

| | | |
|------------------|--|------------|
| 2.5.4 | Transform Domain Coefficients' Dependencies..... | 40 |
| 2.5.5 | Embedded Bit-plane Coding using SAQ | 42 |
| 2.5.6 | Practical Wavelet-based Scalable Image Coding..... | 52 |
| 2.6 | SUMMARY | 53 |
| CHAPTER 3 | SCALABLE 3-D MESH COMPRESSION | 55 |
| 3.1 | INTRODUCTION | 55 |
| 3.2 | MULTIRESOLUTION ANALYSIS OF SEMI-REGULAR MESHES | 57 |
| 3.2.1 | Theory..... | 58 |
| 3.2.2 | Lifting-based Wavelet Transform..... | 62 |
| 3.2.3 | Spatially Adaptive Wavelet Transform (SAWT)..... | 63 |
| 3.3 | SCALABLE QUANTIZATION | 65 |
| 3.3.1 | Wavelet Coefficients' Histogram | 66 |
| 3.3.2 | Proposed Laplacian Mixture Model..... | 66 |
| 3.3.3 | Distortion-Rate Function | 68 |
| 3.3.4 | Model Validation | 68 |
| 3.3.5 | Optimal Embedded Quantization..... | 74 |
| 3.4 | ANALYSIS OF WAVELET COEFFICIENT DEPENDENCIES | 75 |
| 3.4.1 | Statistical Information Analysis..... | 76 |
| 3.5 | SCALABLE MESH COMPRESSION: OVERVIEW AND DESIGN | 81 |
| 3.5.1 | Progressive Geometry Compression (PGC) | 81 |
| 3.5.2 | Scalable Intra-band Mesh (SIM) Compression | 83 |
| 3.5.3 | Composite Context-conditioned (3xC) Compression | 87 |
| 3.5.4 | Visual Comparisons..... | 91 |
| 3.6 | CONCLUSIONS | 94 |
| 3.7 | APPENDIX | 95 |
| CHAPTER 4 | MULTIPLE DESCRIPTION CODING: AN OVERVIEW | 97 |
| 4.1 | INTRODUCTION | 97 |
| 4.2 | MULTIPLE DESCRIPTION PROBLEM | 98 |
| 4.2.1 | Distortion-rate Bound | 99 |
| 4.2.2 | Discussion..... | 100 |
| 4.2.3 | Balanced MDC | 101 |
| 4.2.4 | Redundancy | 102 |
| 4.3 | MULTIPLE DESCRIPTION SCALAR QUANTIZATION | 103 |

| | | |
|------------------|---|------------|
| 4.3.1 | Index Assignment of the Central Quantizer's Indices | 104 |
| 4.3.2 | Good Index Assignments..... | 105 |
| 4.3.3 | Source Specific MDSQ Optimization..... | 107 |
| 4.3.4 | High-Resolution D-R Function of MDSQ..... | 111 |
| 4.3.5 | Two-stage MDSQ..... | 112 |
| 4.4 | OTHER PRACTICAL METHODS FOR MDC | 114 |
| 4.4.1 | Multiple Description Transform Codes | 115 |
| 4.4.2 | MDC via the Polyphase Transform and Selective Quantization..... | 117 |
| 4.4.3 | Forward Error Correction based MDC | 119 |
| 4.4.4 | Discussion..... | 121 |
| 4.5 | SUMMARY | 122 |
| CHAPTER 5 | SCALABLE MULTIPLE DESCRIPTION SCALAR QUANTIZATION | 123 |
| 5.1 | INTRODUCTION | 123 |
| 5.2 | RELATED WORK | 124 |
| 5.3 | SCALABLE MULTIPLE DESCRIPTION QUANTIZER | 125 |
| 5.3.1 | Embedded Index Assignment | 126 |
| 5.3.2 | Description Packetization | 127 |
| 5.3.3 | Definitions | 128 |
| 5.4 | SYMMETRIC SCALABLE MDSQ | 129 |
| 5.4.1 | Connected-cell Scalable MDSQ | 133 |
| 5.4.2 | Disconnected-cell Scalable MDSQ | 134 |
| 5.5 | QUANTIZER OPTIMIZATIONS | 139 |
| 5.5.1 | Lagrangian Formulation | 139 |
| 5.5.2 | Optimization Algorithm..... | 142 |
| 5.5.3 | Discussion..... | 142 |
| 5.5.4 | Performance Comparison | 145 |
| 5.6 | TRANSMISSION VIA PACKET LOSS CHANNEL | 152 |
| 5.6.1 | Balanced Transmission of Descriptions | 153 |
| 5.6.2 | Unbalanced Transmission of Descriptions | 158 |
| 5.7 | CONCLUSIONS | 161 |
| 5.8 | APPENDIX | 163 |
| CHAPTER 6 | CONCLUSIONS AND FUTURE WORK | 165 |
| 6.1 | CONCLUSIONS | 165 |
| | Contribution 1 | 165 |

| | |
|---|------------|
| Contribution 2 | 166 |
| Contribution 3 | 167 |
| 6.2 OPEN ISSUES AND PROSPECTIVE WORK | 167 |
| LIST OF PUBLICATIONS | 171 |
| REFERENCES | 173 |

ACKNOWLEDGMENTS

During the last four years of my Ph.D., I shared great experiences with my colleagues, from whom I learnt not only the art and science of doing research but also many things about the life itself. I wish to pay my gratitude to all fellow workers who were involved directly or indirectly with my work. Without such wonderful peoples my Ph.D. adventure would not be as pleasant as it was.

At the outset, I would like to express my gratitude to my Ph.D. promoters Prof. Adrian Munteanu and Prof. Peter Schelkens, who provided me with the great opportunity and the resources to write my Ph.D. dissertation. As my daily supervisor Prof. Adrian Munteanu always spared his precious time to guide me through the hard times of my research work. At times, when things were not going as expected, his personality served as a constant source of inspiration and encouragement for me. As my advisor, he always kept a nice balance between the team-work and the individual effort; indeed such a balance is very important to make oneself a true independent researcher. I also thank Prof. Adrian Munteanu for his commendable efforts in improving my scientific writing style; as a young and inexperienced researcher I definitely needed this improvement. Prof. Peter Schelkens and Prof. Adrian Munteanu also gave me the chance to join the OptiMMA project, which was a great opportunity for me to learn the trends followed in team based product-oriented research. I am also thankful to Prof. Peter Schelkens and to Prof. Jan Cornelis for reviewing the drafts of my publications and for their expert opinion on my research work.

I would also like to acknowledge the members of my research teams for their valuable discussions and for the joint publications we produced over the years. In this regard, I would like to thank Frederik Verbist, Nikos Deligiannis, Leon Denis and Basel Nabulsi.

I am especially thankful to Karin Debruyne for her support in the administrative aspects of my Ph.D. Also, special thanks to An Vanbellinghen and Evelien Rooms for helping me in day to day office arrangements.

I am very grateful to Prof. Leo Van Biesen, Prof. Rik Pintelon, Dr. Joeri Barbarien, Dr. Fabio Verdicchio, Prof. Jan Cornelis, Prof. Jan Lemeire, Prof. Adrian Munteanu and Prof. Peter Schelkens for accepting to serve as the role of my Ph.D.

jury, for the time they spent in reading my Ph.D. thesis and for the valuable remarks and suggestions they made on my research work.

Certainly, the list of acknowledgements will not be complete without the names of my colleagues and friends at the ETRO department, namely, Nikos Deligiannis, Frederik Verbist, Ruxandra Florea, Leon Denis, Aneta Markova, Bruno Cornelis, Tim Bruylants, Tan Maxine, Petar Marendic, Andrei Sechelea, Geert Braeckman, Jan Hanca, Jan Lievens and Bob Andries. I will never forget the enjoyable experience of working with you guys; especially I will always cherish our exotic get-togethers at university BBQs and annual ETRO meetings.

How can I forget to mention the names of my friends I met in Leuven, namely, Asim, Sughis, Aftab, Amir, Obaid, Ali, Babar, and especially my cricket playing friends, namely, Shahab, Puneet, Rauf, Ajay Sampath, Ajay Godara, Khurram, Zaib, Rehan and Shahid Ahmad. Thank you all for making my stay in Leuven a memorable experience.

Last but not least, I would like to thank my family for their continuous support and encouragements throughout my Ph.D. term. I thank my wife Tabassim for her love during these years, without her patience the completion of my Ph.D. dissertation would have certainly delayed. Especially, I would like to thank my parents Shahida Perveen and Mahmood Khan for giving me the freedom of choice, for their prayers and for their endless encouragements throughout my academic career.

Shahid Mahmood Satti
Brussels, July 2012

ABSTRACT

Scalable representation of a source (e.g., image/video/3-D mesh) enables decoding of the encoded bit-stream on a variety of end-user terminals with varying display, storage and processing capabilities. Furthermore, it allows for source communication via channels with different transmission bandwidths, as the source rate can be easily adapted to match the available channel bandwidth. From a different perspective, error-resilience against channel losses is also very important when transmitting scalable source streams over lossy transmission channels. Driven by aforementioned requirements of scalable representation and error-resilience, this thesis focuses on the analysis and design of scalable single and multiple description scalar quantizers.

The first part of this dissertation deals with the design of scalable wavelet-based semi-regular 3-D mesh compression systems, which provide superior compression performance when compared to the existing mesh coding methods. We point out that, in general, existing methods employ coding techniques which were previously used for wavelet-based scalable coding of images. In principle, image and mesh data exhibit different statistical properties. In this sense, our design methodology thoroughly analyzes different modules of the coding system in order to develop appropriate design choices for efficient compression of semi-regular meshes. In particular, a Laplacian mixture (LM) model is proposed to closely approximate the distribution of the mesh wavelet coefficients. The distortion-rate (D-R) function of the LM model is analytically computed in order to identify the optimal embedded dead-zone quantizer to be used in wavelet-based coding of semi-regular meshes. Following an information-theoretic analysis of the statistical dependencies between wavelet coefficients we conclude that, for meshes, intraband and composite dependencies are far stronger than the commonly employed interband dependencies. Based on our analysis, we propose intraband and composite mesh codecs which give state-of-the-art compression performance. The proposed codecs provide both resolution and quality scalability. This lies in contrast to the existing zero-tree based interband mesh coding techniques, which only support quality scalable decoding of the compressed mesh.

The second part of the dissertation relates to the design of scalable multiple description scalar quantizers, in order to provide source scalability and error-resilience in a single coding framework. In the literature, such a joint framework is

referred to as the scalable multiple description coding (SMDC). In this context, a generic symmetric scalable multiple description quantizer (SSMDSQ) is proposed which generates perfectly balanced source descriptions. Compared to existing designs, it is shown that the proposed quantizer constructions exhibit superior D-R performance in both high and medium-to-low redundancy regimes. Moreover, an innovative extension of the Lloyd-Max algorithm is introduced in order to optimize scalable multiple description scalar quantizers. Anchored in the designed SSMDSQs, an SMDC framework is established to realize packet-based transmission over erasure channels. In this framework, transmission strategies are determined for scenarios wherein the average packet loss rate over the transmission link is (a) unknown and (b) can be estimated at the encoder. Experimental results for generalized Gaussian (GG) and image sources confirm that, compared to contemporary schemes, the designed quantizer constructions (with or without optimization) account for a significant average gain in the signal-to-noise-ratio (SNR) for a wide range of packet loss rates.

ACRONYMS

| | |
|------------------|--|
| 1-D | One-Dimensional |
| 2-D | Two-Dimensional |
| 3-D | Three-Dimensional |
| 3xC | Composite Context-Conditioned |
| A/D | Analog-to-Digital |
| AFX | Animation Framework eXtension |
| ARQ | Automatic Repeat reQuest |
| AVC | Advanced Video Coding |
| bpp | bits per pixel |
| bps | bits per source sample |
| CDF | Cumulative Density Function |
| DCT | Discrete Cosine Transform |
| DDZ | Double Dead-Zone |
| D-R | Distortion-Rate |
| DVB | Digital Video Broadcast |
| DWT | Discrete Wavelet Transform |
| EBCOT | Embedded Block Coding with Optimal Truncation |
| EDSQ | Embedded Dead-zone Scalar Quantizer |
| EIA | Embedded Index Assignment |
| EMDSQ | Embedded Multiple Description Scalar Quantizer |
| FEC | Forward Error Correction |
| FLC | Fixed-Length Coding |
| GG | Generalized Gaussian |
| HH | High-High |
| HL | High-Low |
| IA | Index Assignment |
| IID | Independent and Identically Distributed |
| IZ | Isolated Zero |
| JPEG | Joint Photographic Experts Group |
| JPEG-2000 | JPEG standard 2000 |
| JSCC | Joint Source-Channel Coding |
| KL | Kullback-Leibler |
| LH | Low-High |
| LL | Low-Low |
| LM | Laplacian Mixture |

| | |
|-----------------|--|
| LNC | List of Non-significant Coefficients |
| LOD | Levels Of Detail |
| LSB | Least Significant Bit |
| MDC | Multiple Description Coding |
| MDSQ | Multiple Description Scalar Quantizer |
| MDTC | Multiple Description Transform Coding |
| MDUSQ | Multiple Description Uniform Scalar Quantizer |
| MESH | Measuring Errors between Surfaces using the Hausdorff distance |
| ML | Modified Linear |
| MPEG | Moving Picture Experts Group |
| MSB | Most Significant Bit |
| MSE | Mean Squared Error |
| NBC | Natural Binary Code |
| NEG | NEGative |
| NOR | NORmal |
| NSG | Non-SiGnificant |
| PCM | Pulse Code Modulation |
| PCRD-OPT | Post Compression Rate Distortion OPTimization |
| PDF | Probability Density Function |
| PGC | Progressive Geometry Compression |
| PMF | Probability Mass Function |
| POS | POSitive |
| PSNR | Peak Signal-to-Noise-Ratio |
| QoE | Quality of user Experience |
| QT-L | QuadTree-Limited |
| RL | Refinement List |
| SAQ | Successive Approximation Quantization |
| SAWT | Spatially Adaptive Wavelet Transform |
| SGN | SiGNificant |
| SIM | Scalable IntraBand Mesh |
| SL | Significant List |
| SMDC | Scalable Multiple Description Coding |
| SNR | Signal-to-Noise-Ratio |
| SPIHT | Set Partitioning In Hierarchical Trees |
| SQP | SQUare Partitioning |
| SSMDSQ | Symmetric Scalable Multiple Description Scalar Quantizer |
| TAN | TANgential |
| TG | Touma and Gotsman |
| TP | Transmission Path |
| UEP | Unequal Error Protection |

| | |
|------------|----------------------------|
| UEQ | Uniform Embedded Quantizer |
| UQ | Uniform Quantizer |
| VLC | Variable Length Coding |
| WQT | Wavelet QuadTree |
| Z | Zero |
| ZTR | Zero-Tree Root |

Chapter 1

INTRODUCTION

1.1 QUANTIZATION

The aim of *quantization* is to map a quantity (a continuous function or a high cardinality discrete set) to a countable discrete number of elements. The error introduced during the quantization process is referred to as the *quantization error* or *distortion*. The theory of quantization is more than hundred years old and started in 1899 with Sheppard's study of round-off errors [101]. However, its main importance was only recognized during the early developments of the pulse code modulation (PCM) [79] telephonic systems. The idea of rate-distortion (or conversely distortion-rate (D-R)) was first introduced by Claude Elwood Shannon in his famous paper "A Mathematical Theory of Communication" [98]. Later, Shannon exhaustively dealt with rate-distortion as a complete theory and sketched the fundamental limits of the quantization process¹ in [99]. He also proved that these fundamental limits can be achieved with codes (quantizers) of infinite complexity. Since then, a significant amount of effort has been spent by the scientific community to reach these limits using quantizers of certain bounded complexity.

In most cases, it is desirable to reconstruct the source perfectly at the decoder. This requires infinite or at least an extraordinary precision of the quantizer de-quantizer pair. To avoid this bottleneck, a general goal in the signal encoding science is to minimize the introduced distortion in the reconstructed signal while fulfilling the implementation constraints at hand. In general, signal encoding can be understood as *signal compression* in the sense that a function of a continuous or large but finite alphabet is approximated using a function of a relatively small alphabet. In general, there can be multiple levels of quantization in a single data transmission system. For example, in PCM systems, a standard analog-to-digital (A/D) converter samples the continuous speech signal at 8000 samples/sec and

¹ Claude Shannon termed quantization as *source coding with fidelity criterion*. In the literature, the earlier emphasizes the practical design of quantizers while the latter is generally used to refer to the non-constructive theoretical study of quantizing sources.

quantizes each sample with 12 bits, making a binary stream of bitrate 96000 bits/sec. For transmission of this bit-stream over a standard 9600 bits/sec modem an additional compression of 10:1 is required. This can be achieved with an additional quantization step which maps the pre-quantized alphabet to a 10 times smaller alphabet.

One may suspect that with the ever increasing development of wideband technologies and fast processing equipment, the requirements for data compression would perhaps extinguish. In fact, the need for efficient compression is multifold. In addition to the ease of data storage and transmission, compression enables multiplexing of a large number of channels in wideband systems. Of particular concern are the low bandwidth radio channels used for commercial mobile and data telephony. In such applications, output (speech or data) from a large number of users needs to be multiplexed over low bandwidth wireless channels. The real-time operation of such applications cannot be achieved without efficient data compression. An alternative of data compression, in such cases, would be to install an entirely new transmission facility with an increased transmission bandwidth. Certainly, this alternative is less cost-efficient compared to making more sophisticated terminals that can perform efficient data compression.

1.2 SCALABLE CODING USING QUANTIZATION

In addition to the reduction in precision, quantization enables an appropriate representation of the data for a particular type of application. For example, in broadcast systems such as digital video broadcast (DVB), a single video bit-stream is transmitted to a number of viewers with varying requirements of bandwidth, display resolution and computational complexity. This requires *scalable representation* or *scalable compression* of the video content to enable the extraction of the parts of the bit-stream that are suitable for decoding the compressed video on a terminal with certain specific requirements. This way a single bit-stream can serve a number of different decoders, alleviating the need to independently encode the video for each user. Another requirement of scalable compression arises when the encoder needs to adapt the source rate to the available channel bandwidth. Using scalable compression, a single bit-stream corresponding to the highest rate of the source is first generated. From this highest-rate bit-stream, any lower-rate bit-stream can be easily constructed by extracting the relevant data chunks. This way, using scalable compression, the source rate can be tuned to the available resources without the need to repeat the complete encoding process again.

In practice, scalable representation of a source is created using a multi-level

quantization system, also referred to as *embedded quantization* [108]. In embedded quantization, quantization levels are created such that the quantization regions (cells) of the finer levels are embedded in those of the coarser levels. In particular, the quantization cells of a certain level are created by splitting each quantization cell of the previous level into two or more sub-cells. Thus, the quantization accuracy strictly increases or else, the quantization distortion strictly decreases with every new quantization level. In such a setting, the quantization index of a finest-level cell can be used to compute the corresponding index of the quantization cell of any coarser level. Due to this property, the produced highest-rate bit-stream can be easily parsed to produce a bit-stream which is suitable for decoding the source on a certain type of decoder, or which fulfills the available channel bandwidth constraint.

Over the last decade, many different scalable image, video, and 3-D mesh coding techniques have been proposed in the literature. The research efforts in the area of multimedia scalability have led to the development of scalable coding standards such as the JPEG-2000 [108] standard for scalable image coding and the MPEG AFX [1] standard for scalable 3-D mesh coding, respectively. These standards heavily rely on embedded quantization for providing scalability of the source representation. In scalable video compression, e.g., the scalable extension of H.264/AVC [2] also referred to as the scalable video coding (SVC) standard, the encoded bit-stream contains a single base and one or more refinement layers, which are realized using *predictive coding* methods [44] rather than embedded quantization. In SVC, the use of the refinement layers improves video quality (quality scalability), spatial resolution (resolution scalability), and/or the temporal resolution (temporal scalability) of the base layer. This is analogous to embedded quantization, wherein every new refinement level successively improves the source reconstruction quality.

In this dissertation, we studied embedded quantization from a theoretical as well as from a practical point of view. The considered sources are the coefficient subbands of wavelet decomposed semi-regular 3-D meshes. Using a novel Laplacian mixture (LM) model, a detailed analysis of embedded quantization, in the context of scalable 3-D mesh compression, is carried out. For any given wavelet subband, our analysis derives the optimal embedded quantization strategy that can be theoretically used to achieve the lowest distortion. However, application of subband-specific embedded quantization is generally not practical, as every wavelet subband needs to be analyzed first in order to determine the optimal embedded quantizer. Fortunately, the distortion deficit of commonly used successive approximation quantization (SAQ) [108] for any subband is only marginal with respect to the optimal quantizer, leading us to a subband independent, near-optimal practical quantization strategy. In

addition to the quantization aspects, we also analyzed other modules of a wavelet-based 3-D mesh codec in order to establish best design choices which will lead us to optimum designs of scalable semi-regular 3-D mesh compression systems.

1.3 SOURCE SCALABILITY AND ERROR-RESILIENCE

In addition to the scalability of source representation, network transmission also requires resilience against channel losses. This is because transmission losses can lead to an unpredictably high decoding distortion even though quantization (or coding) is done efficiently. In the literature, *channel coding* techniques, more formally known as forward error correction (FEC) schemes, provide an efficient solution for data recovery in the presence of channel losses. In general, FEC based data recovery involve complex decoding techniques, e.g., the famous *Viterbi* algorithm [119], which adds a significant structural delay in low-delay applications such as real-time delivery of audio/video content. Traditionally, FEC schemes are coupled with re-transmission based techniques, such as the automatic repeat request (ARQ) method [102], to ensure data recovery if the lost data cannot be recovered using FEC alone. Again, for low-delay applications, such as peer-to-peer video streaming [81] or real-time delivery of multimedia content [13, 117], the use of re-transmission methods under severe network conditions can lead to a significantly lower quality of user experience (QoE). For such applications, one needs to go beyond the traditional coding methods [48].

During the early developments of information theory, Claude Shannon [98] proved that, under infinite delay and infinite complexity conditions, optimal transmission of a stationary source through a lossy ergodic channel can be achieved by separate source and channel coding paradigms. In the literature, this result is known as Shannon's *source channel separation theorem*. For this reason, source coding and channel coding have grown into separate fields tackled by separate scientific communities. Specifically, the source channel separation theorem makes use of assumptions that often have limited practical validity. In particular, when the source is non-stationary, or the channel is non-ergodic, or if the design of the transmission system is constraint in terms of a finite delay or a finite complexity, a separate design of source and channel coders may lead to a significant performance penalty with respect to a joint design, also referred to, in the literature, as the joint source-channel coding (JSCC).

In the recent past, multiple description coding (MDC) has emerged as an attractive solution for JSCC in order to provide error-resilience in error-prone packet-switched networks [48]. In contrast to single description coding (SDC),

MDC benefits from the transmission of several mutually refinable source descriptions over multiple unreliable channels. Since the source descriptions are correlated, any subset of them received at the decoder can be used to reconstruct the source with certain fidelity. In MDC, the quality of source reconstruction increases with the reception of each additional source description. This is in contrast to scalable coding using embedded quantization, where all lower layers are required to decode the next refinement layer. Bearing this specific property, MDC avoids re-transmission of lost packets, and thus descriptions, if a decoder's quality constraint is met upon reception of a subset of descriptions. In real networks, under severe error-prone conditions, re-transmission of lost packets for a large number of receivers is not always feasible since it can lead to network congestion [6].

In the context of MDC, quantizer-based schemes are considered quite important due to their practical relevance in many applications [97, 115]. The first quantizer based two-description MDC schemes, also referred to as multiple description scalar quantizers (MDSQs), were proposed by Vaishampayan in [113, 114]. An MDSQ maps each source sample to two side quantizer indices, which individually yield a coarse reconstruction of the source sample while jointly corresponding to a fine reconstruction of the source sample. On the same principle, more efficient vectored extensions of two-description quantizers were later introduced in [47, 96, 116].

In applications that involve real-time delivery of multimedia content over best-effort networks, adaptation to varying channel conditions needs to be performed in real-time. In addition, real-time applications are deployed over networks with a high degree of heterogeneity in terms of available bandwidth, user requests and loss characteristics, thus simultaneously demanding error-resilience and a scalable representation of the source in a single coding framework. In the literature such a framework is referred to as embedded or scalable MDC (SMDC) [117]. In SMDC, each source description is made scalable by employing layered coding, hence facilitating its scalable encoding and decoding. Note that fixed-rate (i.e., non-scalable) solutions for MDC, such as fixed-rate MDSQs, are only optimized for a given source rate, and hence they cannot provide optimum performance over a range of source coding rates [51].

To realize SMDC using quantization based techniques, one needs to explicitly design a scalable extension of MDSQs. A scalable MDSQ framework was initially proposed in [51]. Unfortunately, the scalable MDSQ of [51] permits only a limited control over redundancy between descriptions. In order to produce an arbitrary number of scalable descriptions as well as to permit an enhanced control over the redundancy between the quantization levels, embedded MDSQs (EMDSQs) were introduced in [38, 39, 41-43]. Anchored in the designed EMDSQs, a scalable

multiple description image coding framework was presented in [40], which coupled the multiresolution nature of the wavelet transform with the robustness and scalability features provided by EMDSQs. An extension of [40] was the scalable multiple description video coding system of [117], which provides an improved coding efficiency versus error-resilience trade-off with respect to data-partitioning based video MDC systems, such as [9].

In this dissertation, we introduce a class of scalable MDSQs, referred to as symmetric scalable MDSQ (SSMDSQ), which entails a perfectly symmetric source D-R surface. In this context, quantization of generalized (e.g., Gaussian, Laplacian, etc.) sources as well as wavelet-decomposed photographic image sources is studied. Novel designs of un-optimized and optimized (optimization is done with respect to the source statistics) SSMDSQs entail exceptionally better D-R performance compared to existing scalable MDSQs, thus providing the basis for significantly improved packet-based SMDC frameworks for generalized as well as for photographic image sources.

1.4 MAJOR CONTRIBUTIONS AND THESIS ORGANIZATION

In this section, we give a general overview of the thesis structure and highlight the major contributions of our work.

In Chapter 2, fundamental concepts of source coding are presented. Specifically, this chapter briefly introduces notations and definitions to get the reader familiar with the basic concepts of source coding. Additionally, the material included in this chapter will serve as a general background for the remaining parts of the dissertation.

Using the fundamental concepts presented in Chapter 2, embedded quantization of wavelet coefficients of a wavelet-decomposed semi-regular 3-D mesh is studied in Chapter 3. In a broader sense, the main contribution of Chapter 3 lies in the design of scalable compression systems for semi-regular 3-D meshes. Existing mesh coding methods [57, 58], [83] employ coding techniques, e.g., zero-tree or EBCOT based bit-plane coding, which were specifically designed for wavelet-based coding of images [100], [108]. In existing systems, no theoretical or practical justification has been provided for utilizing image coding techniques for mesh coding. In general, image and mesh data may exhibit totally different statistical properties, as images are consisting of pixels (with intensities) while mesh data involve geometry, i.e., the positions of vertices in a 3-D space. In our design of scalable mesh codecs, we follow a constructive approach, whereby, different modules of the coding system are thoroughly analyzed in order to predict the optimal design choices. In particular,

we conduct a model-based study to evaluate the performance of embedded dead-zone uniform quantization in the context of scalable mesh compression. Furthermore, we employ an information-theoretic analysis of the statistical dependencies between wavelet coefficients in order to figure out whether the statistical dependencies exploited by the state-of-the-art scalable mesh coders [57, 58] are actually the best possible ones. We point out that the state-of-the-art scalable codecs [57, 58] exploit interband wavelet dependencies, i.e., the dependencies between the wavelet coefficients of different subbands [100]. Our analysis reveals that, for meshes, the intraband dependencies, i.e., the dependencies among the wavelet coefficients of a given subband [73, 75], and the composite dependencies, i.e., the combined interband and intraband dependencies [54], [124], are far more stronger than the interband ones. This theoretical finding suggests that, for achieving efficient compression of meshes, one should exploit either intraband or composite dependencies instead of the commonly exploited interband dependencies.

This analysis is followed by a careful design of intraband and composite mesh codecs. The proposed mesh coding systems surpass the existing techniques such as the progressive geometry compression (PGC) system of [57], in terms of the compression performance and the scalability of the compressed mesh. Additionally, it gives comparable compression results with respect to the non-scalable codec of [83]. The performance evaluation of the proposed codecs confirms their compression superiority over interband codecs [57, 58], thus confirming the conclusions drawn from our analysis. An additional feature of the proposed systems is the increased scalability of the compressed mesh. In particular, the proposed codecs provide both resolution and quality scalability of the compressed bit-stream. This lies in contrast to the existing state-of-the-art codecs [57, 58], which only support quality scalability.

In a packet loss transmission scenario, scalable coding alone cannot provide a graceful degradation of source reconstruction quality as the packet loss rate increases. In this context, MDC [48] has evolved as an attractive solution. In Chapter 4, an introduction to the fundamental concepts of MDC and an overview of the relevant information theoretic findings are given. Furthermore, this chapter will briefly present different techniques, proposed in the literature, for creating practical multiple description codes. In this context, a special emphasis is given to the quantization based methods since they are most relevant to the topic of this thesis. In general, the information presented in this chapter will serve as the background for Chapter 5, which will present novel techniques for introducing scalability in quantizer-based MDC systems.

The second main contribution of this dissertation can be found in Chapter 5,

which relates to the design of scalable MDSQs. In this context, a generic symmetric scalable multiple description quantizer (the so called SSMDSQ) is proposed, which generates perfectly balanced source descriptions for a symmetric source probability density function (PDF). This lies in contrast to the contemporary schemes [51], [38, 39, 41-43] which produce roughly balanced or sometimes highly unbalanced descriptions. Novel index assignments (IAs) are proposed to create high as well as medium-to-low redundancy SSMDSQs. Compared to existing designs [51], [38, 42], it is shown that the proposed quantizer constructions provide superior D-R performance when utilized for the quantization of generalized Gaussian (GG) and wavelet-decomposed image sources.

The third major contribution of this dissertation, detailed in Chapter 5, lies in proposing an innovative extension of the Lloyd-Max algorithm [65, 70] in order to enable a PDF specific optimization of scalable MDSQs. The proposed optimization methodology is generic, since it can optimize both symmetric as well as asymmetric scalable MDSQs. We propose both level-constrained and entropy-constrained optimizations of scalable MDSQs. Similar to the Lloyd-Max algorithm, the proposed optimization builds on the fact that an encoder-decoder pair can be iteratively optimized by keeping one component fixed while optimizing the other. By doing so, a non-increasing sequence of values of a cost function is obtained and the convergence of the algorithm is guaranteed. When transmission of generic sources is carried out using a packet-based SMDC system, it is experimentally demonstrated that, compared to contemporary schemes, the designed quantizer constructions (with or without optimization) account for a significant average gain in signal-to-noise-ratio (SNR) for a wide range of packet loss rates. Similar conclusions are also obtained for packet-based SMDC of wavelet transformed image sources.

Chapter 6 draws the conclusions of this work and highlights potential research directions related to this dissertation.

Chapter 2

INFORMATION THEORY AND SOURCE CODING

2.1 INTRODUCTION

In information theory literature, the problem of signal coding is known as *source coding*. The term “source coding” was originally introduced by Claude Elwood Shannon during his classical development of information theory in the late 40’s. The basic concept behind source coding is to remove the *redundancy* (the repeated information) from the message produced by an information source, so that, the message can be encoded using fewest encoding symbols or bits. This is in opposition to the concept of *channel coding* which aims at adding redundancy in the message to be transmitted over a channel, so that, the additionally added information (redundancy) may help in retrieving the original message despite the contamination caused by channel errors or losses. According to Shannon’s landmark paper [98], if the source rate is less than the *channel capacity*, error-free transmission of the source is attainable. This implies that the optimal rate-distortion performance can be achieved by first compressing the source to a rate that is less than the channel capacity and then transmitting this compressed source in an error-free manner over the channel (off course with the help of protection provided by the channel coding schemes and assuming that the sum of the source and channel rates does not exceed the channel capacity). Hence, the source and the channel coding parts of the transmission system can be designed *independently* while maintaining the overall rate-distortion optimality of the transmission system, an argument proven by Shannon’s *source-channel separation theorem*. In this chapter, we will confine our discussion to the source coding part only. Channel coding, or more specifically joint source-channel coding aspects are covered in Chapter 4 and Chapter 5.

In this chapter, we present fundamental information theoretic concepts of classical lossless and lossy source coding. These concepts will serve as a general background

for the remaining parts of the dissertation. For both lossless and lossy source coding, information theoretic limits are discussed without going into the details of how these limits were actually established. Since the main topic of the thesis is *scalable quantization*, we give a detailed overview of *embedded quantization*. Furthermore, a brief overview of the modern scalable image coding systems, which couple embedded quantization with multiresolution techniques to achieve the state-of-the-art scalable image compression efficiency, is also presented.

2.2 DEFINITIONS

In the following we give some important definitions, which will be used during the course of this dissertation.

2.2.1 Entropy

Let X denote a discrete random variable, defined over an alphabet \mathcal{X} . Also, let $p(x)$, be the probability mass function (PMF) of X , where $x \in \mathcal{X}$ represents a realization of X . The entropy $H_a(X)$ gives the exact measure of the *uncertainty* associated to describe X . In other words, $H_a(X)$ denotes the minimum number of a -ary coding symbols required to unambiguously (or losslessly) represent X . The entropy $H_a(X)$ of the random variable X is denoted as [21]:

$$H_a(X) = \sum_{x \in \mathcal{X}} p(x) \log_a \frac{1}{p(x)}. \quad (2.1)$$

In general, the logarithmic base used is 2, which means that on average the minimum number of binary symbols required to represent X is $H_2(X)$. The other commonly used base is the natural-log base, i.e., $a = e$. In this dissertation, the entropy function is written without the log-base subscript and it always refers to the binary entropy. Note that for a uniformly distributed source, i.e., when $p(x) = 1/|\mathcal{X}|$, for every $x \in \mathcal{X}$, $H(X) = \log_2 |\mathcal{X}|$, where $|\mathcal{X}|$ is the cardinality of the set \mathcal{X} .

For continuous sources, entropy is termed as the *differential entropy*. For a continuous random variable X , the differential entropy $h(X)$ is defined as [21]:

$$h(X) = \int_{x \in \mathcal{X}} f(x) \log_2 \frac{1}{f(x)} dx. \quad (2.2)$$

where \mathcal{X} in the above equation denotes the support interval of X and $f(x)$ denotes the probability density function (PDF) of X .

2.2.2 Conditional Entropy

The conditional entropy $H(X|Y)$ is the entropy of a random variable X given

the knowledge of a related random variable Y . In principle, the conditioning of a random variable on another *dependent* random variable always reduces the uncertainty of the first random variable [21], i.e.,

$$H(X) \geq H(X|Y).$$

The equality in the above expression is achieved if X and Y are two *independent* random variables. The conditional entropy is defined as:

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p(y) H(X|Y=y). \quad (2.3)$$

An alternate expression for $H(X|Y)$ can be derived as follows:

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p(y) \underbrace{\sum_{x \in \mathcal{X}} p(x|y) \log_2 \frac{1}{p(x|y)}}_{H(X|Y=y)}.$$

Using the definition of the joint probability, i.e., $p(x, y) = p(y)p(x|y)$ [21], the above expression simplifies to:

$$H(X|Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log_2 \frac{1}{p(x|y)}.$$

Substituting $p(x|y) = p(x, y)/p(y)$, the above expression becomes

$$H(X|Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log_2 \frac{p(y)}{p(x, y)}. \quad (2.4)$$

2.2.3 Joint Entropy

The joint entropy $H(X, Y)$ is the measure of the joint uncertainty associated to two random variables X and Y . The joint entropy $H(X, Y)$ is defined as:

$$H(X, Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log_2 \frac{1}{p(x, y)}. \quad (2.5)$$

Using $p(x, y) = p(x)p(y|x)$ and solving further, we can write equation (2.5) as

$$H(X, Y) = \underbrace{\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 \frac{1}{p(x)}}_{p(x)} + \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log_2 \frac{1}{p(y|x)}.$$

The first term in the above expression is the entropy of X while the second term is the conditional entropy $H(Y|X)$, thus,

$$H(X, Y) = H(X) + H(Y|X). \quad (2.6)$$

Similarly, an alternate simplification of equation (2.5) can result in the following expression

$$H(X, Y) = H(Y) + H(X|Y). \quad (2.7)$$

2.2.4 Relative Entropy

The *relative entropy* of a PMF $p(x)$ with respect to another PMF $q(x)$ is defined as [21]:

$$D(p\|q) = \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)}. \quad (2.8)$$

Note that $D(p\|q) \neq D(q\|p)$ and $D(p\|q) = D(q\|p) = 0$ only when $p(x) = q(x)$ for all x . It is intuitive to verify that:

$$D(p\|q) = E_p[-\log_2 q(x)] - H(X). \quad (2.9)$$

In the above equation $D(p\|q)$ is represented as the difference of two uncertainties. Thus, the relative entropy can also be understood as some kind of error function that computes the approximation error incurred when a PMF $p(x)$ is approximated using another PMF $q(x)$. In general, $D(p\|q)$ is considered as the positive distance between two PMFs, although it does not fulfill the requirements of a *distance metric*, namely the symmetry and the triangular property. In the literature, the relative entropy is also referred to as the Kullback-Leibler (KL) divergence [21], or simply the KL distance.

2.2.5 Mutual Information

The mutual information $I(X;Y)$ between two random variables X and Y is defined as the decrease in the entropy of X due to the knowledge that Y provides about X [21], i.e.,

$$I(X;Y) = H(X) - H(X|Y). \quad (2.10)$$

Using the definition of $H(X)$ and $H(X|Y)$, the following expression can be derived for the mutual information

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}. \quad (2.11)$$

If X and Y are two independent random variables, we have $p(x,y) = p(x)p(y)$, and the mutual information $I(X;Y)$ is zero. The expression in equation (2.11) is symmetric, thus we can write $I(X;Y) = I(Y;X)$, i.e., X and Y provide equal information about each other. Since $H(X) \geq H(X|Y) \geq 0$, we have $0 \leq I(X;Y) \leq H(X)$, where $I(X;Y) = H(X)$ only when Y gives a complete knowledge about X , i.e., $H(X|Y) = 0$. Substituting $H(X|Y) = H(X,Y) - H(Y)$ from equation (2.7) in equation (2.10) we get

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \quad (2.12)$$

i.e., the mutual information is the difference between the sum of marginal entropies and the joint entropy of two random variables.

useful to display the codewords of a prefix-free code in the form of a D -ary tree. The binary-tree for the prefix-free code in the example of Table 2.1 is shown in Figure 2.1. The codewords correspond to the nodes (labeled using the source symbols in Figure 2.1) of the binary-tree. A prefix-free code consists of codewords which correspond to the terminal nodes, while a non-prefix-free code also includes codewords corresponding to the internal nodes of the tree.

2.3.1 Kraft's Inequality

Theorem [21]: *There exists a D -ary prefix-free code for a source with alphabet \mathcal{X} if and only if*

$$\sum_{x \in \mathcal{X}} D^{-l(x)} \leq 1. \quad (2.14)$$

Proof:

If all nodes of the tree are considered at length l_{\max} , the total number of terminal nodes in a D -ary tree are $D^{l_{\max}}$. By choosing a codeword with length $l(x) \leq l_{\max}$ no node in the sub-tree is eligible for another codeword due to the prefix-free condition. Each sub-tree consists of $D^{l_{\max} - l(x)}$ nodes and the set of sub-trees is obviously disjoint, thus

$$\sum_{x \in \mathcal{X}} D^{l_{\max} - l(x)} \leq D^{l_{\max}}. \quad (2.15)$$

Dividing both sides of the above equation by $D^{l_{\max}}$ proves the necessary and sufficient condition of equation (2.14) for a prefix-free code. ■

2.3.2 Lower Bound on Lossless Source Coding

Theorem [21]: *Any D -ary prefix-free code for a discrete memoryless source X , $x \in \mathcal{X}$, satisfies*

$$\frac{H_D(X)}{\log_2 D} \leq E[l(X)], \quad (2.16)$$

with equality occurs if and only if $p(x) = D^{-l(x)}$, for $\forall x \in \mathcal{X}$.

Proof:

The problem of finding the optimal codeword lengths $l(x)$ can be formulated as a standard optimization problem using the Kraft's inequality $\sum_{x \in \mathcal{X}} D^{-l(x)} \leq 1$ as the constraint on the length of the prefix-free code. The Lagrangian functional for the optimization problem is then given by:

$$J = \sum_{x \in \mathcal{X}} p(x)l(x) + \lambda \left(\sum_{x \in \mathcal{X}} D^{-l(x)} - 1 \right). \quad (2.17)$$

Setting the derivative $\partial J / \partial l(x) = 0$, we obtain

$$D^{-l(x)} = \frac{p(x)}{\lambda \log_e D}. \quad (2.18)$$

Substituting the expression for $D^{-l(x)}$ in the expression of the constraint, i.e., $\sum_{x \in \mathcal{X}} D^{-l(x)} = 1$, the optimum value of λ is given by $\lambda = 1/\log_e D$. When substituted in equation (2.18), this expression for λ yields:

$$p(x) = D^{-l(x)}, \quad \forall x \in \mathcal{X}, \quad (2.19)$$

or

$$l(x) = -\log_D p(x). \quad (2.20)$$

For a prefix-free code whose codeword lengths are given by equation (2.20), the expected codeword length is

$$E[l(X)] = -\sum_{x \in \mathcal{X}} p(x) \log_D p(x) = H_D(X). \quad (2.21)$$

Thus, we can say that the D -base entropy is the lower bound on the average codeword length of a prefix-free code, created using a D -ary tree. For source codes using binary alphabets, the lower limit on the average codeword length is the binary entropy. ■

Note that for a uniformly distributed source, i.e., when $p(x) = 1/|\mathcal{X}|$, $\forall x \in \mathcal{X}$, a fixed-length code can be created using the terminal nodes of a binary-tree of depth $\log_2 |\mathcal{X}|$ if and only if $\log_2 |\mathcal{X}|$ is an integer number. Note that, $\log_2 |\mathcal{X}|$ is integer if $|\mathcal{X}|$ is some integer power of 2. In this case, all the codewords have the same length, i.e., $\log_2 |\mathcal{X}|$ bits, and the average codeword length is $E[l(X)] = \log_2 |\mathcal{X}|$. Since, for such a source $H(X) = \log_2 |\mathcal{X}|$, we can say that in this case the fixed-length code can achieve the entropy and hence is optimal. However, for the cases when $|\mathcal{X}|$ is not some power of 2, or when $p(x)$ is not uniform, fixed-length codes are generally sub-optimal and variable length codes are required to achieve the entropy in terms of the average codeword length.

2.3.3 Shannon Coding

We know that the lower bound on the data compression can be achieved by taking the codeword lengths of the source symbols as $l(x) = -\log_D p(x)$, $\forall x \in \mathcal{X}$. In general, $-\log_D p(x)$ will not always give an integer length for each codeword. A near-optimal codeword length can be derived by ceiling the suggested length $-\log_D p(x)$ to the closest integer value, i.e.,

$$l(x) = \lceil -\log_D p(x) \rceil, \quad \forall x \in \mathcal{X}. \quad (2.22)$$

A prefix-free code, complying with the derived lengths, can then be obtained by using the terminal nodes of the accordingly generated D -ary tree. This simple technique was used by Claude Shannon [98] to propose the first variable length

coding method, also known as Shannon coding. For a five alphabet random variable $\mathcal{X} = \{a_1, a_2, a_3, a_4, a_5\}$ with probabilities $\{0.3, 0.3, 0.2, 0.1, 0.1\}$, the Shannon coding gives the codeword lengths of $\{2, 2, 3, 4, 4\}$ bits. The average codeword length for the considered example is $2 \times 2 \times 0.3 + 3 \times 0.2 + 2 \times 4 \times 0.1 = 2.6$ bits per source sample (bps). Since the entropy of the source is 2.1710 bps, we can say that the produced Shannon code has a redundancy of 0.429 bps.

It can be easily verified that after ceiling $-\log_D p(x)$ the achieved codeword lengths will still satisfy the Kraft's inequality. Due to the ceiling operation,

$$-\log_D p(x) \leq l(x) < -\log_D p(x) + 1. \quad (2.23)$$

Multiplying both sides of the above equation with $p(x)$ and by taking the summation over all x , we can write:

$$H_D(X) \leq E[l(X)] < H_D(X) + 1. \quad (2.24)$$

In other words, we can say that the Shannon code has an average overhead of at most 1 bps due to the fact that $-\log_D p(x)$ doesn't always give an integer codeword length. However, this overhead can be made small by spreading it over many source symbols, i.e., by coding a block of n source symbols together. For an independent and identically distributed (IID) memoryless source, such a block coding of source symbol yields [21]:

$$H_D(X) \leq E[l(X)] < H_D(X) + \frac{1}{n}. \quad (2.25)$$

Thus, the average codeword length can reach the entropy $H_D(X)$ asymptotically as $n \rightarrow \infty$.

2.3.4 Shannon-Fano Coding

The Shannon coding, described in Section 2.3.3, is sub-optimal mainly due to the fact that almost no attention is paid towards efficient code-tree generation, i.e., the code-tree is simply generated using codeword lengths derived by the ceiling of $-\log_D p(x)$ values. Shannon-Fano coding on the other hand relies on efficient (although still sub-optimal) generation of the code-tree, wherein, each tree node is split into two sub-nodes with a smallest probability difference.

For the discrete random variable $\mathcal{X} = \{a_1, a_2, a_3, a_4, a_5\}$ with probabilities $\{0.3, 0.3, 0.2, 0.1, 0.1\}$, the code-tree generation of the Shannon-Fano coding is shown in Figure 2.2. In practice, the source symbols are first sorted by the order of their probability, starting from the most probable one to the least probable one (in our example the source symbols are already sorted). In the sorted symbol list, splitting the source symbols into two groups $\{a_1, a_2\}$ and $\{a_3, a_4, a_5\}$ will result in a minimum difference in the probability of two groups, i.e., $0.6 - 0.4 = 0.2$. Note that any other splitting results in a difference of probability that is higher than 0.2.

The first group $\{a_1, a_2\}$ is then split into individual elements. The second group $\{a_3, a_4, a_5\}$ is split into a_3 and $\{a_4, a_5\}$ as such a splitting again gives minimum probability difference. Proceeding this way from the root to the leaves a code-tree can be created. The assigned codewords for each source sample are reported in the caption of Figure 2.2. The average codeword length of the produced code is $2 \times 2 \times 0.3 + 2 \times 0.2 + 2 \times 3 \times 0.1 = 2.2$ bps, which has the redundancy of 0.029 bps, i.e., 0.4 bps lower than the Shannon code of Section 2.3.3. Moreover, block coding of source symbols can also be used in the context of Shannon-Fano coding to further reduce the average codeword length.

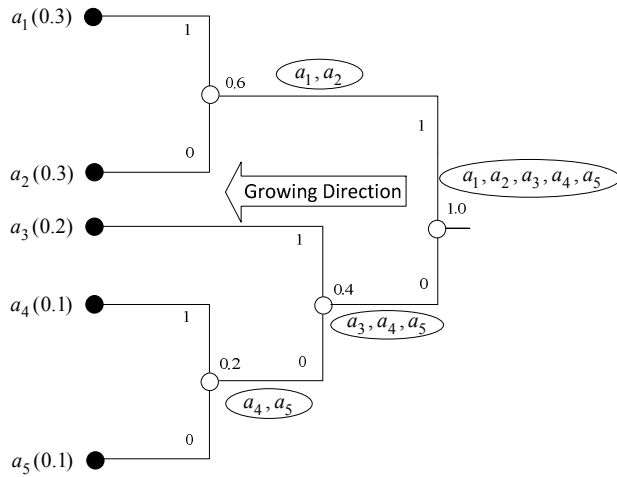


Figure 2.2: Code-tree generation in Shannon-Fano coding. The prefix-free binary Shannon-Fano code is: a_1 (11), a_2 (10), a_3 (01), a_4 (001), a_5 (000). The code-tree grows from the tree-root to leaves.

2.3.5 Huffman Coding

For long block lengths, Shannon-Fano coding provides average codeword length close to the optimum limit, however, for short block lengths a large performance gap exists with respect to the entropy. This performance gap is due to the sub-optimality of the code-tree generation. In contrast, Huffman coding avoids this problem by generating an optimal code-tree. Without going into theoretical details, we explain binary Huffman coding [55] of the discrete random variable defined in the examples of the previous sections.

The aim is to create an optimal binary code for the considered source. An optimal code will assign the largest number of bits to the least probable symbols, i.e., symbols a_4 and a_5 in the considered case. Let $l(a_4)$ and $l(a_5)$ denote the lengths of the codewords for a_4 and a_5 . $l(a_4)$ and $l(a_5)$ must be equal, since otherwise we can delete the last bit of the longer codeword and make them equal while still satisfying the prefix-free condition [55]. Note that this argument also holds for two

least probable symbols even though their probabilities are not equal. In Huffman's method, we first combine the minimum probability nodes a_4 and a_5 to create an internal code-tree node b with probability $p(a_4) + p(a_5) = 0.2$. Next, we combine two nodes which have the minimum probability values. For the considered example, these two nodes are b and a_3 , which are combined to get the internal node c having probability 0.4. Next, nodes a_1 and a_2 should be combined as they both have probabilities smaller than 0.4 (the probability of the internal node c). Proceeding this way we can reach the root of the code-tree which corresponds to probability 1.0. The generated code-tree is shown in Figure 2.3. Starting from the root, distinctive binary symbols are then assigned to each branch of the code-tree to generate the codewords, – see Figure 2.3. The assigned codewords for each source symbol are reported in the caption of Figure 2.3. The average codeword length for the produced code is 2.2 bps, which can be simply computed by summing the probabilities of the intermediate (white circle) nodes [21]. We note that, for the considered example, the final code-trees for Shannon-Fano and Huffman codes are exactly the same. For this reason, the average codeword length in both cases is also the same.

While the current example fails to highlight the advantage of Huffman code over Shannon-Fano code, Huffman proved [55] that the backward grouping of least probable symbols is an optimal tree generation mechanism. From this perspective Huffman codes are referred to as the *optimal prefix-free codes*, which in a general setting would result in an average codeword length smaller than the Shannon-Fano code. However, we point out that, despite being optimal, Huffman codes cannot approach the entropy bound closer than the limits specified in equation (2.24). This is due to the fact that the codewords need to be of integer length. The only cases in which binary Huffman coding can achieve the entropy limit are the ones wherein the source probabilities are some negative powers of 2. Moreover, for low entropy sources the performance gap from the entropy limit can be significantly large. Particularly, for source entropies below 1 bps, Huffman codes are quite inefficient. In these cases, the redundancy of the code can be reduced by carrying out Huffman coding on blocks of source symbols, also referred to as *vector* or *block* Huffman coding. However, for long block lengths, code-tree generation in block Huffman coding gets significantly complex and the method is highly impractical. For block Huffman coding the achievable performance bound is given by equation (2.25). Similar to the binary Huffman code one can create a D -ary Huffman code by combining D least probability symbols at each tree-level.

The source coding schemes described till now assume availability of the source statistics ahead of time. However, in many cases the source statistics may not be

pre-known. For such cases the encoder and the decoder in the Huffman coding can agree to periodically update the source statistics synchronously. In the literature, such an approach is known as the *dynamic* or *adaptive* Huffman coding [120]. Adaptive Huffman coding avoids a prior transmission of codewords to the decoder, which is an advantage. On the other hand, adaptation generally results in a poor code performance compared to the original Huffman code, since, the source estimate can only be done causally. The adaptive approach is generally not used in practice since it makes the decoder quite complex, as it needs to implement the optimal code construction algorithm as well.

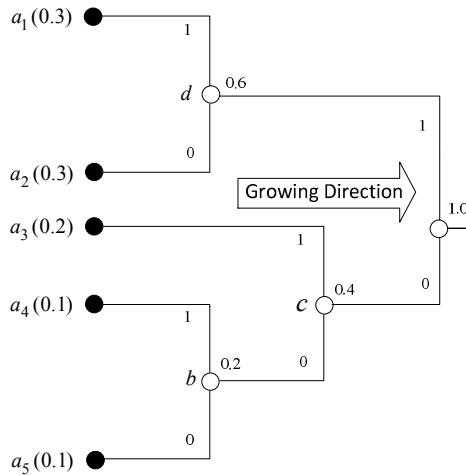


Figure 2.3: Binary tree for the example Huffman code. The prefix-free binary Huffman code is: a_1 (11), a_2 (10), a_3 (01), a_4 (001), a_5 (000). Note that the code-tree grows from the leaves to the root.

2.3.6 Arithmetic Coding

For long block lengths and large alphabet sizes Huffman codes are not practical as they require ordering and look-up tables for their implementation. On the other hand, for certain practical applications, it would be nice to have an algorithm which can “compute” a codeword from the source message instead of mapping the source symbols to a particular codeword using the look-up table. In these settings, arithmetic coding [123] provides a favorable alternative since, unlike Huffman coding, it does not need to store all codewords in the form of a look-up table. Like Huffman coding in Section 2.3.5, we explain the concept of arithmetic coding using the discrete source $\Pr(x) \in \{0.3, 0.3, 0.2, 0.1, 0.1\}$ example. For more details on arithmetic coding, we refer to [21, 123].

Arithmetic coding is based on what was previously known as the Shannon-Fano-

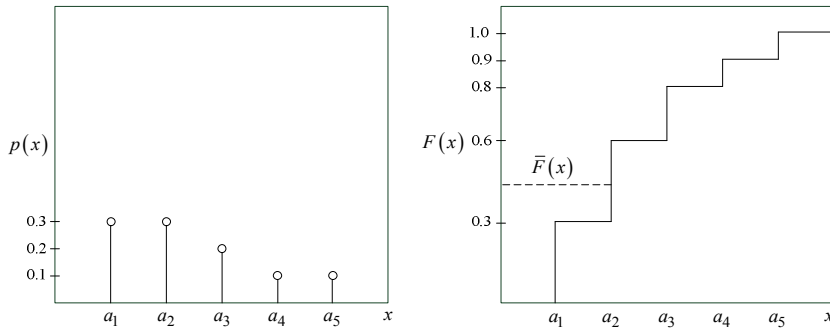


Figure 2.4: (Left) Probability mass function of the considered random variable.
(Right) Cumulative density function of the considered random variable.

Elias coding in the information theory community. Shannon-Fano-Elias coding relies on the cumulative density function (CDF) of the source random variable, defined as:

$$F(x) = \sum_{i \leq x} p(i). \quad (2.26)$$

The CDF of the considered random variable is shown in Figure 2.4. Consider the modified cumulative function

$$\bar{F}(x) = \sum_{i < x} p(i) + \frac{1}{2} p(x). \quad (2.27)$$

Note that, $\bar{F}(x)$ is the mid-point of the CDF interval corresponding to x . Since the intervals are unique to the source alphabets, the value of $\bar{F}(x)$ can be used to code any source alphabet x . In general, $\bar{F}(x)$ is a real fractional value which can only be expressed using an infinite number of bits. To avoid this problem, we can approximate $\bar{F}(x)$ by truncating its binary representation to some finite number of bits. However, for unique decoding, the approximated value should still belong to the interval of x as the original $\bar{F}(x)$. Assuming that we truncate the binary representation of $\bar{F}(x)$ to first $l(x)$ digits after the decimal point yielding an approximate value $\lfloor \bar{F}(x) \rfloor_{l(x)}$, by the definition of truncation

$$\bar{F}(x) - \lfloor \bar{F}(x) \rfloor_{l(x)} < 2^{-l(x)}. \quad (2.28)$$

If $l(x) = \lceil -\log_2 p(x) \rceil + 1$, then

$$2^{-l(x)} < \frac{1}{2} p(x) = \bar{F}(x) - F(x-1). \quad (2.29)$$

Thus, $\lfloor \bar{F}(x) \rfloor_{l(x)}$ always belongs to the interval of x and $l(x)$ suffices the number of bits to represent x . Furthermore, it can be easily shown that for disjoint CDF intervals the resulting code is prefix-free. For the considered random variable the Shannon-Fano-Elias code is given in Table 2.2. Note that Shannon-Fano-Elias code is always worse than the equivalent Shannon code as it always assigns one extra bit

Table 2.2: Shannon-Fano-Elias code for the considered random variable.

| x | $p(x)$ | $\bar{F}(x)$ | $\bar{F}(x)$ in binary | $l(x)$ | Code |
|-------|--------|--------------|------------------------|--------|-------|
| a_1 | 0.3 | 0.15 | 0.001... | 3 | 001 |
| a_2 | 0.3 | 0.45 | 0.011... | 3 | 011 |
| a_3 | 0.2 | 0.7 | 0.1011... | 4 | 1011 |
| a_4 | 0.1 | 0.85 | 0.11011... | 5 | 11011 |
| a_5 | 0.1 | 0.95 | 0.11110... | 5 | 11110 |

for each codeword. However, unlike Shannon code, the Shannon-Fano-Elias code does not need a look-up table for the encoding/decoding operations.

The poor performance of Shannon-Fano-Elias coding is simply because it encodes one source symbol at a time. In contrast, arithmetic coding represents a block of symbols using a finite precision fractional number [123]. In arithmetic coding, for coding the 2nd symbol, the encoder takes the CDF interval corresponding to the first symbol in the sequence and subdivides it into $|\mathcal{X}|$ sub-intervals. During this subdivision the relative width of the sub-intervals is kept exactly the same as the relative width of the parent intervals – see Figure 2.5. For the 3rd source symbol, the corresponding sub-interval in the second stage is taken and is again subdivided using the same principle. This procedure is also followed for the following source symbols in the sequence till the complete sequence can be localized to a single minute interval. A fractional value within this interval can be used to encode the source message. In general, the center of the interval is taken as this fractional value.

In Figure 2.5, an example of arithmetic coding of the source sequence $a_1a_3a_2a_2$ is shown. The source sequence is mapped to the fraction 0.2061. The equivalent binary representation of this fractional number is 0.001101001... When this binary number is truncated to the first 9 digits after the decimal points the corresponding fraction modifies to 0.205078125, which is still within the interval [0.2034, 0.2088) as the original fraction. Thus the 9 digit binary code 001101001 can be used to uniquely represent the source sequence. The decoding works in a similar way as the encoding, namely, given the binary representation of the sequence, the decoder can easily find the corresponding CDF interval of the first symbol in the sequence to unambiguously decode it. The CDF partition for the next symbol can be computed by subdividing the identified interval into relative size sub-intervals (as done at the encoder side). The fractional value is then used to identify a sub-interval within the parent interval, and hence the second decoded symbol. Proceeding this way the

decoder can decode all the symbols in the sequence². In the considered example, the achieved average codeword length is $9/4=2.4$ bps which is slightly poor compared to the equivalent Huffman code which gives an average codeword length of 2.2 bps. However, in practice the efficiency of the arithmetic code increases when coding longer source sequences. In fact, arithmetic coding asymptotically achieves the lower bound on lossless source coding for infinitely long sequences [123].

Arithmetic coding provides multifold advantages over the Huffman coding. Consider a source having alphabet $\{0,1\}$ where $p(0)=0.01$ and $p(1)=0.99$. For such a source, Huffman coding can at the best assign 1 bit as codeword to each symbol, leading to an average codeword length of exactly 1 bps. Compared to the entropy of the source, which is 0.0807 bps, such a code gives the redundancy of 0.919 bps (roughly 11 times the source entropy). Thus, for such an unbalanced probability model, Huffman code gives quite poor performance. On the other hand, arithmetic coding can perform very efficient for this case, provided that a long sequence of binary source symbols is coded.

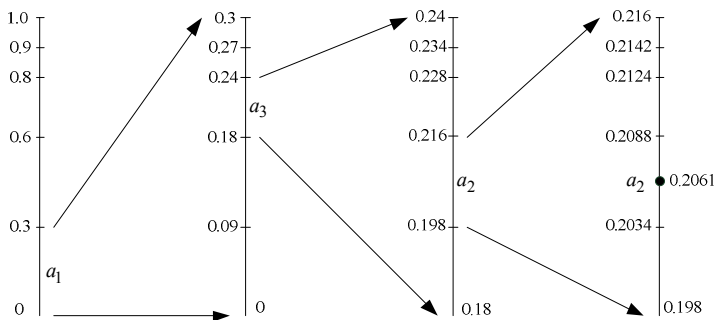


Figure 2.5: Arithmetic coding of the source sequence $a_1a_3a_2a_2$.

Another advantage of arithmetic coding is that it allows for seamless *adaptation* of probability models [123]. In many practical applications, the complete data to be compressed may not be available beforehand. In such cases, one cannot compute the relative frequency and hence the probability of each source symbol in advance. In contrast, the *adaptive arithmetic coding* allows for learning the source probability while encoding. The adaptive arithmetic coding starts by assuming equal probability of all source symbols (i.e., the uniform probability model). Then, every time a source symbol is encoded its relative frequency (and hence its probability) is modified at the encoder. The encoder and the decoder stay in perfect synchronization during the encoding and the decoding processes. In particular, the

² To properly stop the decoding process the decoder must know the total number of encoded source symbols.

decoder decodes the current symbol with a given model and updates the model (using the decoded information) in order to decode the next source symbol.

The literature on lossless source coding is very rich. In the previous sections, we have tried to include the most relevant source coding algorithms. In addition to the above described methods, other source coding techniques include Golomb codes [46], Lempel-Ziv codes (LZ77, LZ78) [126], [127], the Lempel-Ziv-Welch (LZW) code [122], etc. The details of these methods are not provided here.

2.4 LOSSY SOURCE CODING

In contrast to lossless coding, lossy coding algorithms enable compression in a setting where the source needs not to be perfectly decoded at the decoder side. In general, the aim is to efficiently compress the source, while fulfilling a certain constraint on the distortion between the original and the reconstructed information. In this context, a distortion measure ρ is defined as a *mapping* from the original source alphabet \mathcal{X} and the reconstructed alphabet $\hat{\mathcal{X}}$ to the set of non-negative real numbers \mathbb{R}^+ , i.e.,

$$\rho: \mathcal{X} \times \hat{\mathcal{X}} \rightarrow \mathbb{R}^+. \quad (2.30)$$

A distortion metric $\rho(x, \hat{x})$ is a *bounded* metric if $\max \rho(x, \hat{x}) < \infty$. When $\mathcal{X} = \mathbb{R}$, $\hat{\mathcal{X}} \in \mathbb{R}$, the most commonly used distortion metric is the *squared error*, which is defined as $\rho(x, \hat{x}) = (x - \hat{x})^2$. The mean of the squared error over all source outcomes, i.e., $d(x, \hat{x}) = E[(x - \hat{x})^2]$, is referred to as the *mean squared error* (MSE). In this dissertation, MSE is taken as the default distortion metric unless specified otherwise.

2.4.1 Rate-Distortion Theory

The theoretical limits for the lossy source coding paradigm were established by Claude Shannon in his second landmark paper [99]. In [99], Shannon formulated a set of rules for lossy source coding, typically referred to as the rate-distortion theory. The main contribution of his work was the derivation of the rate-distortion function $R(D)$ of a memoryless information source. The $R(D)$ function defines the lowest possible rate needed to encode the source such that the distortion is less than or equal to a fixed value D .

The $R(D)$ function of a continuous source X with the marginal PDF $f(x)$ is defined as:

$$R(D) = \sum_{x \in \mathcal{X}} f(x) \min_{\hat{x} \in \hat{\mathcal{X}}} \rho(x, \hat{x}) \quad I(X; \hat{X}), \quad (2.31)$$

where \hat{X} denotes the reconstruction random variable. The $R(D)$ function of

equation (2.31) can be simplified as follows:

$$\begin{aligned} I(X; \hat{X}) &= h(X) - h(X|\hat{X}), \\ &= h(X) - h(X - \hat{X}|\hat{X}). \end{aligned}$$

Since, $h(X - \hat{X}) \geq h(X - \hat{X}|\hat{X})$,

$$I(X; \hat{X}) \geq h(X) - h(X - \hat{X}). \quad (2.32)$$

Taking (2.32) into account, minimizing $I(X; \hat{X})$ in equation (2.31) is equivalent to maximizing $h(X - \hat{X})$, i.e.,

$$R(D) = \min_{f(\hat{x}|x): d(x, \hat{x}) \leq D} I(X; \hat{X}) \geq h(X) - \max_{f(\hat{x}|x): d(x, \hat{x}) \leq D} h(X - \hat{X}). \quad (2.33)$$

In information theory, the result of equation (2.33) is known as the *Shannon lower bound* on lossy source coding. Note that, the lower bound on lossy coding is the entropy of the source (which is the lower bound on lossless coding) minus some rate savings due to the introduction of distortion. Since, the aim is to minimize R for a given D , contemplating the right hand side of equation (2.33), one can say that a lossy compression algorithm should aim at maximizing the entropy of the source approximation error $(X - \hat{X})$. Since a memoryless Gaussian random variable \mathcal{N} has the highest differential entropy [21], the maximum $h(X - \hat{X})$ is attained if the error random variable $(X - \hat{X})$ is $\mathcal{N}(0, D)$, where $D = E[(X - \hat{X})^2]$. Since, $h(\mathcal{N}(0, D)) = (1/2) \log_2 2\pi e D$ [21], the lower bound of equation (2.32) is given by the following expression:

$$R(D) \geq h(X) - \frac{1}{2} \log_2 2\pi e D. \quad (2.34)$$

2.4.1.1 Memoryless Gaussian Source $\mathcal{N}(0, \sigma^2)$

For $X \sim \mathcal{N}(0, \sigma^2)$, $h(X) = (1/2) \log_2 2\pi e \sigma^2$ [21] and the lower bound on the achievable rate is given by:

$$R(D) \geq \frac{1}{2} \log_2 2\pi e \sigma^2 - \frac{1}{2} \log_2 2\pi e D = \frac{1}{2} \log_2 \left(\frac{\sigma^2}{D} \right). \quad (2.35)$$

The converse of the $R(D)$ function is the distortion-rate $D(R)$ function, for which the lower bound can be obtained easily from the expression of equation (2.35), i.e.,

$$D(R) \geq \sigma^2 2^{-2R}. \quad (2.36)$$

The $D(R)$ function of a Gaussian source is shown in Figure 2.6(a). Note that, due to the maximum entropy property of a Gaussian source, the $D(R)$ function of any non-Gaussian source will always be below the curve of Figure 2.6(a). With $D(R) = \sigma^2 2^{-2R}$ the source SNR computes to

$$SNR(R) = 10 \log_{10} \left(\frac{\sigma^2}{D(R)} \right) = 20R \log_{10} 2 = 6.02R \text{ (dB)}.$$

For a Gaussian source, no lossy coding scheme can achieve an SNR performance better than $\approx 6.02R$. Derived from this result, the achievable rate-SNR region for a Gaussian source is depicted in Figure 2.6(b).

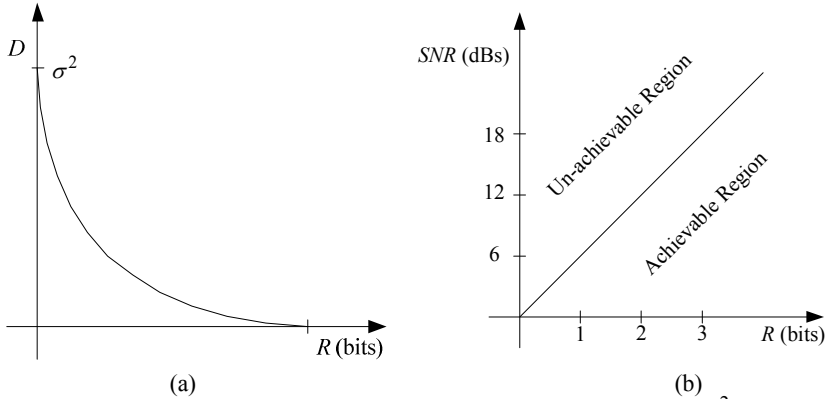


Figure 2.6: (a) The $D(R)$ function of a Gaussian source with variance σ^2 ; (b) The achievable rate-SNR region of a Gaussian source with variance σ^2 .

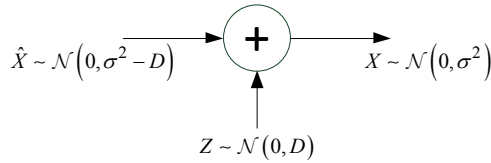


Figure 2.7: Test channel which can theoretically achieve the $R(D)$ for a Gaussian source.

Shannon [99] proved that equality in the rate-distortion bound of equation (2.35) is achieved if the approximation noise $X - \hat{X} = Z$ is $\mathcal{N}(0, \sigma^2 - D)$ and the source approximation can be represented as an *additive noise* channel shown in Figure 2.7. The proof is quite straightforward. Since Z and \hat{X} are independent, $h(X - \hat{X} | \hat{X}) = h(Z | \hat{X}) = h(Z)$ and thus:

$$\begin{aligned} I(X; \hat{X}) &= h(X) - h(Z), \\ &= \frac{1}{2} \log_2 2\pi e \sigma^2 - \frac{1}{2} \log_2 2\pi e D, \\ &= \frac{1}{2} \log_2 \left(\frac{\sigma^2}{D} \right). \end{aligned}$$

The above derived term is identical to the right hand side of equation (2.35), thus proving that the rate-distortion bound of a Gaussian source is theoretically achievable. In practice, however, achieving the rate-distortion bound requires a

careful development of efficient quantization methods. An overview of practical quantization is presented next.

2.4.2 Practical Quantization Methods

Quantization is a process in which the source alphabet is mapped to a relatively smaller reconstruction alphabet, also referred to as the quantizer's *codebook*. A simple example of quantization is the *rounding off* operation, which maps any fractional number to its nearest integer. Quantization is termed as either *scalar* or *vector* depending on the number of source samples quantized at the same time. In our discussion, we will confine ourselves to the scalar case, where $\mathcal{X} = \mathbb{R}$, i.e., $|\mathcal{X}| = \infty$ and $\hat{\mathcal{X}} \in \mathbb{R}$. Moreover, $|\hat{\mathcal{X}}| = N < \infty$.

2.4.2.1 Scalar Quantization

A scalar quantizer deals with one source sample at a time. Any scalar quantizer $Q(x)$ is represented using a set of reconstruction values $\hat{\mathcal{X}} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \dots, \hat{x}_N\}$ and a set of quantizer thresholds $\mathcal{T} = \{t_0, t_1, t_2, \dots, t_N\}$. The quantization operation can be summarized as:

$$\hat{x}_n = Q(x) : t_{n-1} < x \leq t_n, \quad \forall n \text{ where } 1 \leq n \leq N, \quad (2.37)$$

i.e., each sample is mapped to a reconstruction value \hat{x}_n provided that it lies within the n th *quantization cell*. The quantization process can be viewed as a combined effect of two mapping functions, namely, an encoder function \mathcal{E} and a decoder function \mathcal{D} . \mathcal{E} maps source alphabets to quantization indices, i.e., $\mathcal{X} \rightarrow \mathcal{I}$, where $\mathcal{I} = \{1, 2, 3, \dots, N\}$. In general, the quantization indices are further losslessly encoded and decoded using a fixed length code (FLC), e.g., the natural binary code (NBC), or a variable-length code (VLC), e.g., adaptive arithmetic coding. \mathcal{D} maps the quantization indices to the reconstruction values, i.e., $\mathcal{I} \rightarrow \hat{\mathcal{X}}$. With these definitions, $Q(x) = \mathcal{D}(\mathcal{E}(x))$. Occasionally, the decoder is referred to as *inverse quantizer* or *de-quantizer* in the literature.

Figure 2.8(a) depicts the thresholds and reconstruction values of a scalar quantizer. Additionally, the transfer function of a scalar quantizer is plotted in Figure 2.8(b). The transfer function is a staircase function, which consists of horizontal (*tread*) and vertical (*rise*) patches. The quantizer of Figure 2.8(b) is a *mid-tread* quantizer, since the origin $x = 0$ lies within a tread. The opposite type is the *mid-rise* quantizer for which the origin lies on a rise.

The MSE distortion of the reconstructed signal and the entropy of the produced quantization indices are given by the following expressions,

$$D = E \left[(X - \hat{X})^2 \right] = \sum_{n=1}^N \int_{t_{n-1}}^{t_n} (x - \hat{x}_n)^2 f(x) dx, \quad (2.38)$$

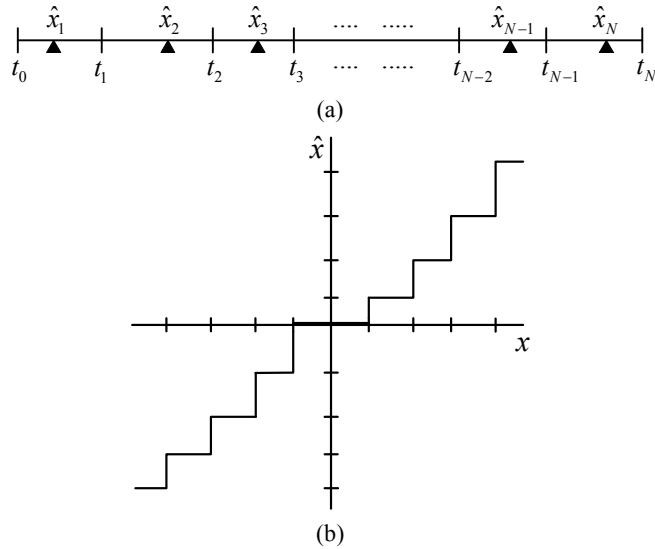


Figure 2.8: A scalar quantizer. (a) The quantizer thresholds and reconstruction points, (b) The quantizer's transfer function.

$$H = -\sum_{n=1}^N p_n \log_2 p_n, \text{ for } p_n = \int_{t_{n-1}}^{t_n} f(x) dx. \quad (2.39)$$

The aim is to create an optimal scalar quantizer so as to achieve the quantizer's experimental rate-distortion performance as close as possible to the $R(D)$ function of equation (2.35).

2.4.2.2 Lloyd-Max Algorithm

Lloyd and Max [65, 70] first established the necessary conditions for the optimality of a scalar quantizer and provided an algorithm, known as the Lloyd-Max algorithm, to minimize the MSE distortion of a scalar quantizer subject to a constraint on the size of the reconstruction codebook, N . Such a constraint makes the Lloyd-Max [65, 70] quantizer an optimal choice when the quantization indices are coded using a FLC. The FLC rate at which each quantization index is encoded is $R = \log_2 N$ bps.

The two necessary conditions for the optimality of a scalar quantizer are the *nearest-neighbor condition* and the *centroid condition*. The nearest-neighbor condition requires that the n th quantization cell consists of all source samples which are closer to \hat{x}_n than any other reconstruction value, i.e., $\forall x: |x - \hat{x}_n| < |x - \hat{x}_{n'}|$, for any $n \neq n'$. Indeed, such a condition gives a minimum MSE distortion and thus the nearest-neighbor condition can be derived by solving $\partial D / \partial t_n = 0$ for any t_n , which turns out to be

$$t_n = \frac{\hat{x}_n + \hat{x}_{n+1}}{2}, \text{ for } n=1,2,3,\dots,N-1. \quad (2.40)$$

Equation (2.40) suggests that any threshold point is always midway between its right and left reconstruction points.

The centroid condition deals with the optimal reconstruction value within a certain quantization cell. The optimal reconstruction value can be determined by solving for zero rate-of-change of the distortion D (equation (2.38)) with respect to any reconstruction variable \hat{x}_n , i.e., by solving $\partial D / \partial \hat{x}_n = 0$. This derivation gives

$$\hat{x}_n = \frac{\int_{t_{n-1}}^{t_n} xf(x) dx}{\int_{t_{n-1}}^{t_n} f(x) dx} = E[X | t_{n-1} < X \leq t_n], \text{ for } n=1,2,3,\dots,N. \quad (2.41)$$

The above equation suggests that the optimal reconstruction value within a quantization cell is the *center of mass* (or centroid) of the part of the PDF which belongs to the quantization cell.

The above described conditions of equations (2.40) and (2.41) are necessary but not sufficient and hence they do not ensure optimality of the scalar quantizer for all general source PDFs. However, for log-concave PDFs, i.e., source PDFs for which $\log(f(x))$ is a concave function, fulfillment of these conditions assures optimality [108]. Uniform, Laplacian and Gaussian distributions satisfy the log-concave property, hence, we can say that the Lloyd-Max scalar quantizers for these distributions are always optimal.

In practice, we often come across situations where the PDF of the source is unknown while its representative data is available. One can use a part of the representative source data as a training set to approximate the solution of equations (2.40) and (2.41). In the limit sense, as the size of the training set increases, the approximated solutions will approach the true ensemble values (weak law of large numbers [21]). Under this analogy, the approximated conditions of (2.40) and (2.41) are given by the following set of equations

$$t_n \approx \frac{\hat{x}_n + \hat{x}_{n+1}}{2}, \quad n=1,2,3,\dots,N-1, \quad (2.42)$$

$$\hat{x}_n \approx \frac{1}{|C_n|} \sum_{x \in C_n} x, \quad C_n = \{x : t_{n-1} < X \leq t_n\}, \quad n=1,2,3,\dots,N. \quad (2.43)$$

$|C_n|$ in the second condition denotes the cardinality of the set C_n , i.e., the number of training sample belonging to the n th quantization cell.

The nearest-neighbor condition creates the optimal encoder partition for a given decoder, while the centroid condition forms the optimum decoder for a fixed encoder partition. Using the training set, Lloyd and Max executed these conditions

one after the other in a recursive fashion up to a point where the distortion value is almost constant between two consecutive iterations. The result is an optimized scalar quantizer for which the achieved distortion point may just be a local minima or even a saddle point. We point out that, for such an iterative optimization the final solution depends on the initialization. Usually, for generalized source PDFs, e.g., Gaussian, Laplacian, etc. a uniform reconstruction point set, i.e., reconstruction points are displaced uniformly in the source granular region, serves as a good initialization setup. Note that, for a zero-mean generalized source, x can be any real number, i.e., $-\infty \leq x \leq \infty$, so $t_0 = -\infty$ (a large negative value) and $t_N = \infty$ (a large positive value) is taken to cover the complete source range.

For a large R , it can be shown that the distortion of an optimal scalar quantizer with a constrained codebook size N can be modeled as [44]:

$$D(R) \cong C \sigma^2 2^{-2R}, \quad (2.44)$$

where, C is a constant factor which depends on the source PDF. For smooth zero-mean symmetric source PDFs, C can be expressed as [44]

$$C = \frac{2}{3} \left(\int_0^{\infty} \sqrt[3]{f(x)} dx \right)^3. \quad (2.45)$$

For a Gaussian source, $C = \sqrt{3}/2 \cong 2.721$ [44], and the Lloyd-Max scalar quantizer operates on a $10 \log_{10}(2.721) \cong 4.35$ dB gap from the D-R bound $\sigma^2 2^{-2R}$ (derived in equation (2.36)).

2.4.2.3 Entropy-constrained Scalar Quantizer

If the quantization indices are coded using a VLC of average rate say R bps, it is more logical to constraint the entropy of the produced quantization indices instead of the codebook size (as was done in the previous section). This is because, the efficient VLC schemes such as arithmetic entropy coding [123] can achieve the entropy of a general information source. The Lagrangian cost function for optimizing the distortion subject to the entropy constraint is given by:

$$\begin{aligned} \mathcal{J} &= D + \lambda H \\ &= \left[\sum_{n=1}^N \int_{t_{n-1}}^{t_n} (x - \hat{x}_n)^2 f(x) dx \right. \\ &\quad \left. - \lambda \sum_{n=1}^N \int_{t_{n-1}}^{t_n} f(x) dx \log_2 \int_{t_{n-1}}^{t_n} f(x) dx \right]. \end{aligned} \quad (2.46)$$

The nearest-neighbor condition, when solved for the entropy-constrained case, is given by [108],

$$t_n = \frac{\hat{x}_n + \hat{x}_{n+1}}{2} + \lambda \frac{\log_2 p_n - \log_2 p_{n+1}}{2(\hat{x}_n - \hat{x}_{n+1})}, \quad n = 1, 2, 3, \dots, N-1. \quad (2.47)$$

Since the second term on the right side of equation (2.46) is independent of \hat{x}_n , the centroid condition for the entropy-constrained case is exactly the same as for the Lloyd-Max case and is given by equation (2.41). Similar to the Lloyd-Max quantizer case, centroid and modified nearest-neighbor (i.e., equation (2.47)) conditions can be recursively executed to realize a locally optimal entropy-constrained scalar quantizer. Moreover, similar to the Lloyd-Max case, a uniform reconstruction point set is also a good initialization for the entropy-constrained optimization. Theoretically, for unbounded random variables, e.g., Gaussian, Laplacian etc., $N = \infty$ is the optimal choice. If a certain choice of N is too large for a given PDF, the algorithm will result in zero probability of some indices, thus effectively reducing to an optimum N . For situations where the source PDF is unknown but the training samples are available, approximation of (2.47) can be easily derived by approximating p_n as $p_n \approx |C_n|/T$, where T denote the size of the training set.

It can be easily shown that H is a non-increasing function of λ [108]. This makes the search procedure for an appropriate λ quite straightforward. The correct value of λ should satisfy the target rate constraint with equality, $H = R_T$. This can be done by simply choosing a large value of λ (for which $H > R_T$) and successively reducing its value till the rate condition is satisfied [108].

It was first established in [45] that, asymptotically (i.e., when $R \rightarrow \infty$), a uniform threshold quantizer with *center* of cell reconstruction is the optimal entropy-constrained scalar quantizer for smooth source PDFs. For a uniform mid-tread quantizer, the thresholds and the reconstruction values can be written as $\pm(n + \frac{1}{2})\Delta$ and $\pm n\Delta$, $n=0,1,2,\dots$, respectively, where Δ is the size of the quantization cell. From equation (2.38), the distortion D for such a scalar quantizer with $N \rightarrow \infty$ can be formulated as:

$$D = \sum_{n=-\infty}^{\infty} \int_{(n-\frac{1}{2})\Delta}^{(n+\frac{1}{2})\Delta} (x-n\Delta)^2 f(x) dx. \quad (2.48)$$

For a very small Δ , the PDF over each quantization cell can be assumed constant, i.e., $f(x) \cong f(n\Delta)$ for $(n-\frac{1}{2})\Delta \leq x < (n+\frac{1}{2})\Delta$, leading to:

$$D \cong \sum_{n=-\infty}^{\infty} f(n\Delta) \int_{(n-\frac{1}{2})\Delta}^{(n+\frac{1}{2})\Delta} (x-n\Delta)^2 dx = \sum_{n=-\infty}^{\infty} f(n\Delta) \frac{\Delta^3}{12}. \quad (2.49)$$

In quantization theory the assumption $f(x) \cong f(n\Delta)$ is often referred to as the *high-rate approximation* [45], [44]. In theory, the high-rate approximation is used to derive the high-rate $R(D)$ function of a quantizer. For an infinitesimally small Δ , the term on the right hand side of equation (2.49) can be approximated using a continuous integral by taking $dx = \Delta$, as

$$D = \frac{\Delta^2}{12} \sum_{n=-\infty}^{\infty} f(n\Delta)\Delta \cong \frac{\Delta^2}{12} \int_{-\infty}^{\infty} f(x)dx. \quad (2.50)$$

Since $\int_{-\infty}^{\infty} f(x)dx = 1$, the above expression leads to:

$$D \cong \frac{\Delta^2}{12}. \quad (2.51)$$

Similar to the distortion, a closed form expression can also be derived for the entropy of the uniform quantizer output, i.e.,

$$H = - \sum_{n=-\infty}^{\infty} p_n \log_2 p_n, \quad p_n = \int_{(n-\frac{1}{2})\Delta}^{(n+\frac{1}{2})\Delta} f(x)dx. \quad (2.52)$$

Using the high-rate approximation $p_n \cong f(n\Delta)\Delta$, $dx = \Delta$, the output entropy can be simplified as

$$\begin{aligned} H &\cong - \sum_{n=-\infty}^{\infty} f(n\Delta)\Delta \log_2 (f(n\Delta)\Delta), \\ &= - \sum_{n=-\infty}^{\infty} f(n\Delta)\Delta \log_2 (f(n\Delta)) - \sum_{n=-\infty}^{\infty} f(n\Delta)\Delta \log_2 (\Delta), \\ &\cong - \int_{-\infty}^{\infty} f(x) \log_2 (f(x)) dx - \log_2 (\Delta) \int_{-\infty}^{\infty} f(x) dx, \end{aligned}$$

leading to,

$$H = h(X) - \log_2 (\Delta), \quad (2.53)$$

where, $h(X)$ denotes the differential entropy of the source – see equation (2.2). Since an efficient VLC can approach the entropy limit, i.e., $R \cong H$, then from the above equation we have

$$\Delta = 2^{H(X)-R}.$$

Substituting the above expression for Δ in equation (2.51) yields:

$$D(R) \cong \frac{1}{12} 2^{2H(X)} 2^{-2R}. \quad (2.54)$$

For a continuous Gaussian source $\frac{1}{12} 2^{2H(X)} = \pi e \sigma^2 / 6$ and the high-rate performance gap of an optimal entropy-constrained scalar quantizer with respect to the Shannon D-R bound of equation (2.36) is $10 \log_{10} (\pi e \sigma^2 / 6) \cong 1.53$ dB.

At low rates the uniform threshold quantizer with center reconstruction is not optimal, however, a uniform threshold quantizer with centroid reconstruction is near optimal [108]. In general, a small improvement in the D-R performance of the uniform quantizer can be obtained by increasing the size of the quantization cell which corresponds to $x = 0$, also referred to as the *zero-cell* or the *dead-zone cell*. Widening the size of the dead-zone cell somewhat increases its distortion contribution, however, it decreases the entropy contribution enough to yield a net D-R gain [108].

The quantizer partition for a generic family of variable size dead-zone uniform

quantizer is given by the thresholds [108]:

$$T = \begin{cases} [-(1-\xi)\Delta, (1-\xi)\Delta) & n = 0 \\ [(n-\xi)\Delta, (n+1-\xi)\Delta) & n > 0 \\ [(n-1+\xi)\Delta, (n-\xi)\Delta) & n < 0 \end{cases} \quad (2.55)$$

In the above equation $n = 0$ corresponds to the dead-zone cell, $n > 0$ corresponds to the quantization bins on the right side of the dead-zone cell, while $n < 0$ corresponds to the bins on the left side. Note that the dead-zone size is $2(1-\xi)\Delta$, for $\xi \leq 1$. With the above quantizer partition, the encoder function can be implemented as [108]:

$$n = \mathcal{E}(x) = \begin{cases} \text{sign}(x) \left\lfloor \frac{|x|}{\Delta} + \xi \right\rfloor & \frac{|x|}{\Delta} + \xi > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (2.56)$$

while the decoder function $\mathcal{D}(x)$ is given by [108]:

$$\hat{x}_n = \begin{cases} 0 & n = 0 \\ \text{sign}(n)(|n| - \xi + \delta)\Delta & n \neq 0 \end{cases} \quad (2.57)$$

where the value of δ , $0 \leq \delta < 1$, defines the exact position of \hat{x}_n within the n th quantization cell. For reconstruction at the middle of the cell, $\delta = 1/2$. Note that, $\xi = 0$ and $\xi = 1$ correspond to the *double* dead-zone and mid-rise uniform quantizers, shown in Figure 2.9(a) and Figure 2.9(b), respectively. Larger dead-zone sizes can be selected by choosing $\xi < 0$.

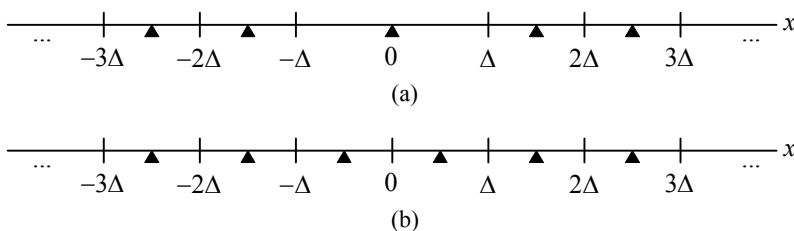


Figure 2.9: Uniform scalar quantizers with center reconstructions. (a) double dead-zone uniform quantizer, $\xi = 0$ (b) mid-rise uniform quantizer, $\xi = 1$.

2.4.3 Embedded Quantization

Fixed-rate scalar quantizers, such as the Lloyd-Max quantizer, provide a single $R(D)$ point on the rate-distortion axis. Thus, every time a different rate-distortion trade-off is required, one needs to quantize the source using an accordingly designed quantizer. In many applications, re-encoding of the source is not an attractive option. Hence, it is beneficial to have a quantization system that can attain different rate-distortion trade-offs using a single encoding of the source.

Along with efficient rate-distortion performance, a very desirable trait of a quantization system is the ability to successively reduce the reconstruction distortion as the compressed bit-stream is decoded. Such a characteristic in turn demands that the decoder should be able to decode the bit-stream at any intermediate rate R_t , $0 < R_t \leq R$, where R denotes the rate at which the source was originally encoded. In theory, a source is referred to as *successively refinable* [35] if $D_t(R_t)$ for an intermediate rate R_t achieves the D-R bound³, i.e., $D(R_t) = D_t(R_t)$. In practice, however, *successive refinability* is the property of a source code that enables quality scalable decoding, i.e., the quality of source reconstruction (or distortion) scales as a function of the employed decoding rate R_t . Indeed, the full reconstruction quality is attained upon decoding of the entire bit-stream at rate R bps.

A quantization method which can fulfill the abovementioned requirements is commonly known as *embedded quantization* [108]. Embedded quantization is a multi-level framework, wherein the quantization cells of the higher level (higher rate) quantizer are embedded within the quantization cells of the lower level (lower rate) quantizer. Equivalently, the quantization cells of the lower rate quantizers can be sub-divided to yield the quantization cells of the higher rate quantizers. $\{Q_p\}_{p=0}^{p=P}$ denotes the set of $P+1$ quantizers, where $p=P$ and $p=0$ correspond to the coarsest and the finest quantization levels, respectively. Partition cells of a three level ($P=2$) embedded scalar quantizer are depicted in Figure 2.10. Notice that the partition cells of Q_p at any level p are embedded in the ones of level $p+1$. Each source sample is mapped to a unique quantization cell $n^{(0)}$ in the finest quantizer Q_0 and the corresponding binary representation n_p, n_{p-1}, \dots, n_0 of the produced quantization index is generated. The most significant bit n_p is the sign (say binary 0 for + and binary 1 for -) of the quantized source sample. A set containing p th binary bit of all source samples forms the p th quantization bit-plane. The bit-stream is formed by combining the bit-planes of the produced quantization indices in the order of their significance. In particular, first the most significant bits (MSBs), i.e., n_p 's, from all source samples are stored in the bit-stream followed by the 2nd MSBs, i.e., n_{p-1} 's, and so on and so forth, till the least significant bits (LSBs), i.e., n_0 's⁴. The decoder can decode each source sample at any of the $P+1$ quality levels. In particular, for a specific level p , the decoder can gather $P-p+1$ bits, i.e., n_p, n_{p-1}, \dots, n_p of a source sample and decode it using the center (or centroid) of the corresponding quantization cell $C_p^{n^{(p)}}$, $n^{(p)} = n_p, n_{p-1}, \dots, n_p$.

³ The D-R bound for any general source can be computed by substituting the expression for its differential entropy $h(X)$ in equation (2.34).

⁴ The produced binary bit-stream can be fed to a binary arithmetic coder for further compression [108].

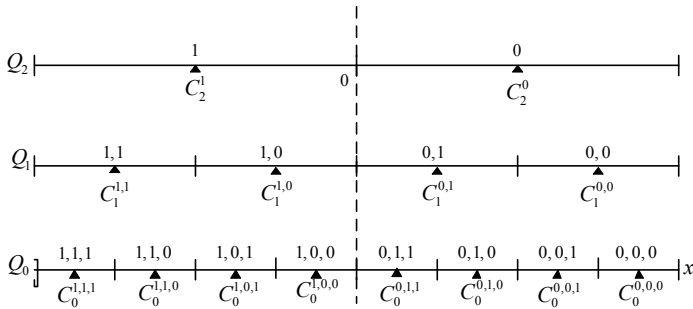


Figure 2.10: An example of uniform mid-rise embedded scalar quantizer for $P = 2$. The triangles indicate the *center reconstruction points*.

One may notice that, for generalized source distributions, it is not possible for more than one embedded level to satisfy the optimality conditions described in Section 2.4.2.2. For example, if the highest level ($p = 0$) quantizer is chosen to satisfy the optimality conditions for a Gaussian distribution, the partition cells for lower levels are fixed and will be sub-optimal. A unique case in which all levels of the embedded quantizer are optimal is the uniform distribution case. This is because, for a uniform distribution, the Lloyd-Max algorithm results in uniform quantizers at each level which can be made embedded. We also notice that, the quantizer of Figure 2.10 does not contain a dead-zone cell which is needed to improve the D-R performance of a quantizer with respect to a conventional uniform quantizer [108].

In the context of embedded quantization, particularly important is the family of embedded dead-zone uniform quantizers [108]. In this family, the quantizer partition at each level contains a dead-zone cell, which gets smaller and smaller for finer levels. A special instance of this family, where the dead-zone cell at each level is twice the size of a non-dead-zone quantization cell, is depicted in Figure 2.11. For embedded dead-zone quantizers, the quantization index $n^{(0)}$ for the finest level quantizer can be determined using the equation (2.56). Let the binary representation of $n^{(0)}$ be $s, n_p, n_{p-1}, \dots, n_0$, where s denotes the sign bit. Then by dropping the p , $0 \leq p \leq P$, LSBs of $n^{(0)}$ we can obtain the following value of $n^{(p)}$:

$$n^{(p)} = (-1)^s \left\lfloor \frac{|n^{(0)}|}{2^p} \right\rfloor = (-1)^s \sum_{l=p}^P n_l 2^{P-l}. \quad (2.58)$$

Substituting the value for $n^{(0)}$ from equation (2.56) to equation (2.58), we have:

$$n^{(p)} = (-1)^s \left\lfloor \frac{\lfloor |x|/\Delta + \xi \rfloor}{2^p} \right\rfloor. \quad (2.59)$$

Since $\lfloor |x|/\Delta + \xi \rfloor = 2^p a + b$ for $a \in \mathbb{Z}$, $0 \leq b \leq 2^p - 1$ and $|x|/\Delta + \xi = \lfloor |x|/\Delta + \xi \rfloor + \varepsilon$, for $0 \leq \varepsilon < 1$, the following is true [76]:

$$\left\lfloor \frac{\lfloor |x|/\Delta + \xi \rfloor}{2^p} \right\rfloor = \left\lfloor \frac{|x|}{\Delta 2^p} + \frac{\xi}{2^p} \right\rfloor.$$

Using the above derived identity, $n^{(p)}$ in equation (2.59) becomes:

$$n^{(p)} = (-1)^s \left\lfloor \frac{|x|}{\Delta 2^p} + \frac{\xi}{2^p} \right\rfloor = \text{sign}(n^{(0)}) \left\lfloor \frac{|x|}{\Delta 2^p} + \frac{\xi}{2^p} \right\rfloor. \quad (2.60)$$

The above equation resembles equation (2.56) and we can say that the quantization index corresponding to any level p can be determined using the equation (2.56) with scaled parameters $2^p \Delta$ and $\xi/2^p$.

Using the above notation the inverse quantization can directly result from a slight modification of equation (2.57), as:

$$\hat{x}_n^{(p)} = \begin{cases} 0 & n^{(p)} = 0 \\ \text{sign}(n^{(p)}) \left(|n^{(p)}| - \xi/2^p + \delta \right) 2^p \Delta & n^{(p)} \neq 0 \end{cases} \quad (2.61)$$

In general, for the family of embedded dead-zone uniform quantizer, $\xi < 1$. The double dead-zone case of Figure 2.11 occurs when $\xi = 0$. In the literature, the double dead-zone embedded quantizer of Figure 2.11 is also known as successive approximation quantization (SAQ) [108].

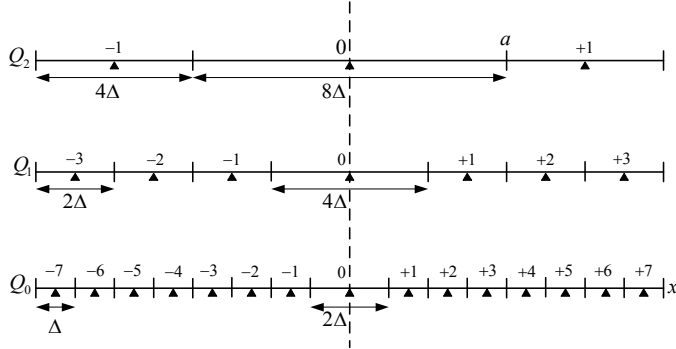


Figure 2.11: Quantizer partitions for embedded double dead-zone uniform quantizer for $P = 2$. A quantization index is shown above each cell. Source reconstruction is done at the middle of each cell, $\delta = 0.5$.

SAQ has a special relevance for quantization in many practical scalable multimedia compression systems [57], [100], [107]. In Section 2.4.2, it was mentioned that at high rates the uniform quantizer with center reconstruction is the optimal entropy-constrained quantizer. Moreover, at lower rates, for peaky source distributions, the performance of the uniform quantizer can be improved by widening the dead-zone cell [108]. Hence, one can conjecture that a quantizer having both these properties will not perform far from the optimal entropy-constrained quantizer for all rates [108]. In this context, the SAQ, which features

both these characteristics, can be regarded as a near-optimal embedded quantization system. Since the wavelet coefficients in the transformed multimedia sources, e.g., image, video, 3-D mesh, have peaky histograms, SAQ provides close to optimal results in quantizing such source [108].

The second important feature of SAQ is its very low cost of implementation. For SAQ, the encoder can compute the quantization index directly from the source sample using equation (2.60) and the reconstructed value using equation (2.61). Thus, the thresholds and the reconstruction points need not to be stored explicitly. This does not hold for non-uniform dead-zone quantizers for which the threshold and reconstruction values need to be stored in look-up tables.

In the next section, we will describe the use of SAQ in wavelet-based scalable image coding systems.

2.5 WAVELET-BASED SCALABLE IMAGE CODING

2.5.1 Transforms for Image Compression

Images consist of pixels which generally possess high degree of correlation due to a smooth spatial variation of intensity. The main function of the transform is to decompose the original image into a set of coefficients which are well decorrelated. As the transforms are generally reversible, this operation does not involve any loss of information and the inverse transform can be used to retrieve the original image losslessly. In the absence of a transform, an image coder needs to employ either relatively less efficient predictive coding based methods [108] or efficient but highly complex vector quantization based methods [44], to capture the statistical dependencies between pixels. On the other hand, the use of transform allows for less complex scalar quantization schemes, while still giving comparable performance with respect to the vector quantization based equivalents [108].

In images, the intensity contribution of different spatial frequencies is highly variable. Due to this reason, the frequency domain representation of the image signal is being used to achieve decorrelation of pixels intensities. A practical example making use of this principle is the discrete cosine transform (DCT) [5], which was employed in the JPEG image compression standard [3]. Transform variants that employ infinite length sinusoid basis functions, e.g., the Fourier transform or the cosine transform do not perform a localized analysis of the input signal, i.e., they cannot capture the spatial position at which the contribution of a certain frequency starts or ends. On the other hand, if we can distinguish image areas of intense activity from smooth areas, we can assign different rates to different types of regions, thus achieving efficient compression without sacrificing the visual

quality. Even though this is also possible with transforms based on sinusoids, e.g., the DCT, wavelets are more suitable for this purpose due to their inherent property of capturing the transients present in the signal [104]. As opposed to the cosine transform, the wavelet transform [69] performs a multi-scale analysis, where the transform coefficients are defined to contribute at various time scales with different spatial frequencies.

2.5.2 Continuous-time Wavelet Transform

For a continuous-time function $f(t)$ defined in $L^2(\mathbb{R}) \equiv \left\{ f(t) : \int_{-\infty}^{\infty} f^2(t) < \infty \right\}$, the continuous-time wavelet transform of $f(t)$ is defined as:

$$W(s, \tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-\tau}{s}\right) dt, \quad (2.62)$$

where, τ and s denote the shift (translation in time) and scale parameters, respectively. $\psi(t)$ is the transform's basis function, also referred to as the *mother wavelet* function. By defining $\psi_{s,\tau}(t) = 1/\sqrt{s} \psi((t-\tau)/s)$, the above equation can be written as:

$$W(s, \tau) = \int_{-\infty}^{\infty} f(t) \psi_{s,\tau}(t) dt = \langle f(t), \psi_{s,\tau}(t) \rangle. \quad (2.63)$$

In other words, the magnitude of the wavelet coefficient $W(s, \tau)$ is the measure of the similarity between $f(t)$ and the scaled and shifted version of the mother wavelet $\psi_{s,\tau}(t)$. The inverse continuous-time wavelet transform is given by the following expression:

$$f(t) = \frac{1}{C_{\Psi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W(s, \tau) \psi_{s,\tau}(t) \frac{dtds}{s^2}. \quad (2.64)$$

Equation (2.64) shows that $f(t)$ can be exactly written as the linear sum of translated and scaled versions of the wavelet function $\psi(t)$. The constant C_{Ψ} in equation (2.64) is given by:

$$C_{\Psi} = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega, \quad (2.65)$$

where, $\Psi(\omega)$ denotes the Fourier transform of the mother wavelet function $\psi(t)$. The necessary condition for the invertibility of the wavelet transform is $C_{\Psi} < \infty$, which is only possible if $\Psi(0) = 0$. In other words, the wavelet function always has a zero average value, i.e., $\int_{-\infty}^{\infty} \psi(t) dt = 0$.

2.5.3 Discrete Wavelet Transform (DWT) and Multiresolution Analysis

Multiresolution analysis aims at representing any arbitrary function

$f(t) \in L^2(\mathbb{R})$ as a set of successive approximations of the original signal at different resolutions. Mallat [69] considered the resolution scales of the form $s = 2^j$, $j \in \mathbb{Z}$. Let V^j denotes a function space at resolution 2^j . Since, the complexity of functions increases with the resolution, in general,

$$V^j \subset V^{j+1}, \forall j \in \mathbb{Z}. \quad (2.66)$$

Mallat [69] showed that there exists a function $\phi(t) \in L^2(\mathbb{R})$, referred to as the *mother scaling function*, such that if $\phi_j(t) = 2^j \phi(2^j t)$ denotes its dilation at scale j , then $2^{-j/2} \phi_j(t - 2^{-j} n)$ with $n \in \mathbb{Z}$ forms an orthonormal basis of V^j . Under this assumption, the continuous approximation $A_j f(t)$ of $f(t)$, at any resolution level j , can be computed as the orthonormal projection of the signal on V^j , i.e.,

$$\forall f(t) \in L^2(\mathbb{R}),$$

$$A_j f(t) = 2^{-j} \sum_{n=-\infty}^{\infty} \left\langle f(t), \phi_j(t - 2^{-j} n) \right\rangle \phi_j(t - 2^{-j} n). \quad (2.67)$$

The inner products terms in the above equation constitute the *discrete approximation* of $f(t)$ at level j , denoted as $A_j^d f$,

$$A_j^d f = \left(\left\langle f(t), \phi_j(t - 2^{-j} n) \right\rangle \right)_{n \in \mathbb{Z}}. \quad (2.68)$$

Mallat [69] showed that these approximations can be computed by using a fast algorithm, known as the *pyramidal algorithm*, which was originally proposed by Burth and Adelson [14] for pyramid-based multiresolution decomposition of images. In particular, by defining a discrete filter h with the impulse response

$$\forall n \in \mathbb{Z}, h(n) = \langle \phi_{-1}(t), \phi(t - n) \rangle, \quad (2.69)$$

and \tilde{h} as the mirror of h , i.e., $\tilde{h}(n) = h(-n)$, the discrete approximation $A_j^d f$ can be computed by convolving the approximation $A_{j+1}^d f$ with \tilde{h} followed by a decimation with a factor of 2, i.e.,

$$A_j^d f(n) = \sum_{k=-\infty}^{\infty} \tilde{h}(2n - k) A_{j+1}^d f(k). \quad (2.70)$$

Let W^j denote the orthogonal complement of V^j in V^{j+1} , i.e., $V^{j+1} = V^j \oplus W^j$. Mallat [69] proved that, similar to mother scaling function, a *mother wavelet function* $\psi(t) \in L^2(\mathbb{R})$ also exists such that $2^{-j/2} \psi_j(t - 2^{-j} n)$, for $n \in \mathbb{Z}$, is an orthonormal basis of W^j ; hence projection of $f(t)$ on W^j yields:

$$P_{W^j} f(t) = 2^{-j} \sum_{n=-\infty}^{\infty} \left\langle f(t), \psi_j(t - 2^{-j} n) \right\rangle \psi_j(t - 2^{-j} n), \quad (2.71)$$

where, the inner product forms the *discret detail* signal at scale j , given by

$$D_j^d f = \left(\left\langle f(t), \psi_j(t - 2^{-j} n) \right\rangle \right)_{n \in \mathbb{Z}}. \quad (2.72)$$

Similar to the equation (2.70), the discrete detail signal can be computed using a

discrete filter $\tilde{g}(n) = g(-n)$, where $g(n) = \langle \psi_{-1}(t), \psi(t-n) \rangle$, followed by a down-sampling by a factor 2. The equivalent of equation (2.70) for the detail signal is given by the following expression,

$$D_j^d f(n) = \sum_{k=-\infty}^{\infty} \tilde{g}(2n-k) A_{j+1}^d f(k). \quad (2.73)$$

$h(n)$ and $g(n)$, which denote the low-pass and the band-pass filters, are related to each other with the following expression [69]:

$$g(n) = (-1)^{1-n} h(1-n). \quad (2.74)$$

In the reverse process, the signal approximation $A_{j+1}^d f$ can be obtained by first up-sampling and then combining the signals $A_j^d f$ and $D_j^d f$ using the following mathematical expression [69]:

$$A_{j+1}^d f(n) = 2 \sum_{k=-\infty}^{\infty} h(n-2k) A_j^d f(k) + 2 \sum_{k=-\infty}^{\infty} g(n-2k) D_j^d f(k). \quad (2.75)$$

Figure 2.12 shows the block diagram of the forward and the reverse wavelet transform derived using equations (2.70), (2.73) and (2.75). The filter pairs (\tilde{h}, \tilde{g}) and (h, g) are termed as the *analysis* and the *synthesis* filter pair, respectively.

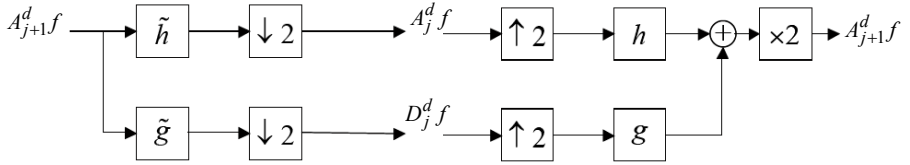


Figure 2.12: A pictorial representation of the forward and the reverse discrete wavelet transform.

Given that the function spaces are nested, V^{j+1} can be expanded as:

$$\begin{aligned} V^{j+1} &= V^j \oplus W^j, \\ &= (V^{j-1} \oplus W^{j-1}) \oplus W^j, \\ &\dots \\ &= V^{j-J} \oplus W^{j-J} \dots \oplus W^{j-2} \oplus W^{j-1} \oplus W^j. \end{aligned} \quad (2.76)$$

By using the convention that $j=0$ denotes the scale of the sampled original signal, $A_0^d f(n) = f(t-nT)$, it is intuitive from the concept given in equation (2.76) that the original signal can be written as the set:

$$A_0^d f = \left(A_{-J}^d f, (D_j^d f)_{-J \leq j \leq 1} \right). \quad (2.77)$$

The above discussed family of wavelets, proposed by Mallat [69], by which the synthesis filters are mirrored versions of the analysis filters, is known as the

orthonormal wavelet expansion due to its perfect reconstruction property. Other famous wavelet families are the *biorthogonal* [118], the *compactly supported orthonormal* [23] and the *compactly supported biorthogonal* [20] wavelet expansions. For brevity, details about these wavelet families are not covered here.

Above, we detailed the wavelet expansion procedure for a 1-D signal. For 2-D image signals, low-pass and band-pass filtering operations need to be carried out in both horizontal and vertical directions independently. Figure 2.13 depicts the wavelet decomposition of the *cameraman* image for two decomposition levels. Corresponding to the type of the filtering applied, detail signals are referred to as the low-low (LL), the low-high (LH), the high-low (HL) and the high-high (HH) *subbands* – see Figure 2.13. The subscript of a subband denotes the decomposition level it belongs to.

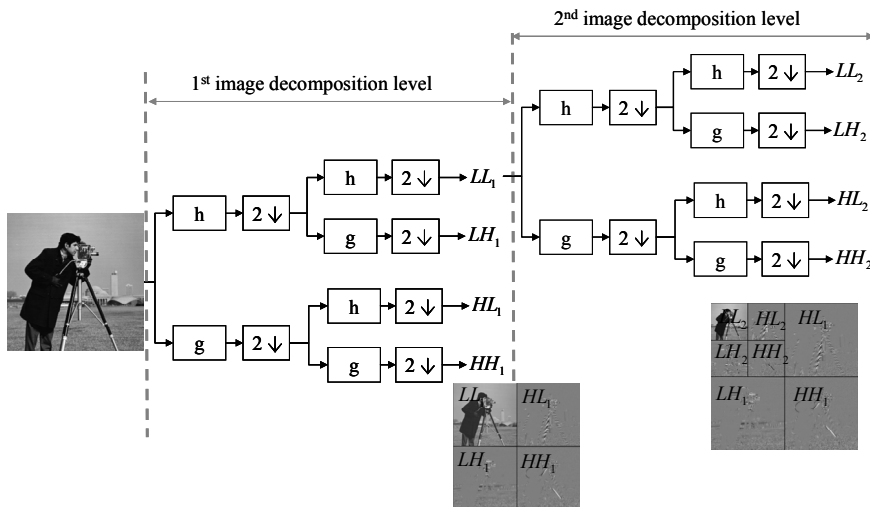


Figure 2.13: 2-D fast wavelet transform of the cameraman image with 2 decomposition levels.

2.5.4 Transform Domain Coefficients' Dependencies

In general, in the wavelet decomposed structure of an image, one can discriminate between three types of statistical dependencies among wavelet coefficients. These dependencies are referred to as the *interband*, the *intra-band*, and the *composite* coefficient dependencies [62], [63].

Interband dependencies are residual statistical dependencies between spatially related coefficients across different frequency subbands. These relationships are also referred to as the parent-descendants dependencies, as illustrated in Figure 2.14. Statistical dependencies exist between the parent and descendants due to the natural decay of the coefficients' magnitude for increasing frequencies. In other words, if a

parent coefficient magnitude is below a certain threshold then there is a high probability that the magnitude of its descendants will be also below this threshold. The wavelet coding paradigms that exploit the parent-descendants dependencies are known as the *interband wavelet codecs*. Practical examples of interband wavelet codecs include the popular embedded zero-tree wavelet (EZW) coding scheme of [100] and the set partitioning in hierarchical trees (SPIHT) coding technique of [87].

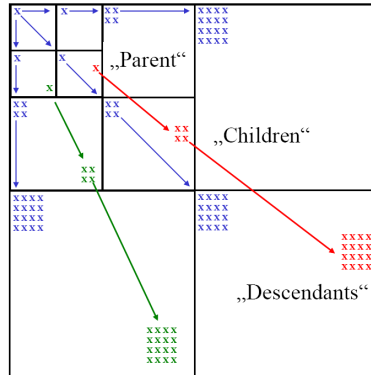


Figure 2.14: Parent-descendant relationship between the wavelet coefficients of different frequency subbands in a wavelet decomposed image.

In addition to interband dependencies, statistical intraband dependencies at any decomposition level exist due to the correlation between the neighboring coefficients of a given frequency subband. A general reason for the existence of intraband dependencies is the smooth variation of intensity values in images. Wavelet coding paradigms that exploit the intraband dependencies to achieve compression are known as *intraband wavelet codecs*. Typical examples of this category include the block-based coding technique of [73], quadtree coding approaches of [74, 75, 94], and the EBCOT codec [107] employed in the state-of-the-art JPEG-2000 [108] scalable image coding standard.

Finally, there is a third category of coding paradigms that exploit both the interband and the intraband statistical dependencies in a single image coding framework. The joint interband and intraband dependencies are referred to as composite dependencies and equivalently, codecs exploiting such dependencies are referred to as *composite wavelet codecs*. Typical codec examples belonging to this category are EZBC scheme of [54] and the ECECOW coding approach of [124].

From an information-theoretic point of view, in images, composite dependencies are significantly stronger than the interband dependencies, while they are only slightly stronger than intraband dependencies [62], [63]. Thus, one should exploit intraband or composite dependencies instead of interband dependencies in order to

achieve efficient compression of images. This argument was confirmed by the development of intraband [74, 75, 94, 107] and composite [54, 124] coding systems, which give better D-R performance compared to interband codec designs [87, 100]. Furthermore, if the encoding algorithm only captures intraband dependencies, i.e., when different frequency subbands are encoded independently, the receiver can always reconstruct the image at a lower (than original) spatial resolution level. This can be done by decoding only the subbands corresponding to the intended resolution scale and simply discarding the others. In other words, intraband models enable *resolution scalable decoding* (in short: *resolution scalability*) of the compressed image. This argument does not hold for interband codec designs due to the fact that different frequency subbands are jointly encoded. For example, in interband codecs [87, 100] the use of zero-trees, which span different frequency subbands, limit the resolution scalable decoding of images. In such codecs, resolution scalability cannot be enabled without sacrificing the compression performance. The other type of scalable image decoding is the *quality scalable decoding* (in short: *quality scalability*), which allows for decoding of the compressed image at various quality levels. In contrast to resolution scalable decoding, quality scalable decoding can be well supported by interband, intraband, and composite codecs.

In the next section, we describe interband and intraband embedded bit-plane coding based on the double dead-zone SAQ quantizer.

2.5.5 Embedded Bit-plane Coding using SAQ

When a 2-D DWT is applied on an image, the spatial frequency information is isolated in distinct frequency subbands of transform coefficients. In the decomposed structure of an image, the LL subband retains the lowest frequency information. Moving horizontally to the right increases the horizontal frequency, while moving vertically to the bottom increases the vertical frequency. Moving diagonally increases both vertical and horizontal spatial frequencies at the same time. In the decomposed structure of an image, wavelet coefficients corresponding to the high-frequency can be more coarsely quantized to save bitrate without sacrificing the image's visual quality. This is because the human visual system is less sensitive to high-frequency variations [108].

In general, the observed wavelet coefficient histogram for a given subband (except for the LL subband) has a larger magnitude close to its *mean* value. Away from the mean, the histogram's magnitude decreases as some exponential function of the distance [69]. For images, wavelet coefficient histograms can be closely modeled using the family of GG distributions [69]. In particular, the GG family of distributions allows for a parametric representation of several source distributions

including the Laplacian, the Gaussian, and the uniform PDF. From a theoretical point of view, SAQ is a near-optimal choice [108] for embedded quantization of generalized peaky source distributions – see Section 2.4.3. In the previous section the structure of the underlying quantizer partition in SAQ was discussed. In this section, representative bit-plane coding frameworks based on SAQ will be presented.

The bit-plane coding with SAQ is generally performed using (at least) two coding passes: the *significance pass* and the *refinement pass*. In SAQ terminology, a source sample x is said to be significant with respect to a threshold T_p if $|x| \geq T_p$, i.e.,

$$S_p(x) = \begin{cases} 0, & \text{if } |x| < T_p \\ 1, & \text{if } |x| \geq T_p \end{cases} \quad (2.78)$$

In the above equation T_p denotes the smallest positive threshold of the embedded quantizer at any bit-plane p . The output of the significance operator of equation (2.78), when applied to the whole wavelet image, results in a binary map also referred to as the *significance map*. The 1's in the significance map denote the wavelet coefficients that are significant with respect to T_p , while 0's denote the coefficients which are not significant. In general, encoding the position information of significant wavelet coefficients, constitutes a major proportion of the total rate needed to encode the wavelet decomposed image [100]. For each level p the encoder needs to encode the position of the wavelet coefficients that are found significant at the current bit-plane level. This position information is encoded using the significance coding pass. Due to a peaky distribution of wavelet coefficients, the significance map for $p = P$ generally contains a large number of 0's and few 1's. Bit-plane coding methods thus employ tools that can encode these 0's as efficiently as possible, while still localizing the occasionally found 1's. At the next bit-plane $p = P - 1$, the threshold is lowered and a few more coefficients become significant. The position information for newly found significant coefficients will be coded using the significant pass of $(P - 1)$ th bit-plane. To enable progressive decoding of coefficients, the sign information of a significant wavelet coefficient should be encoded immediately after coding its significance. For significant coefficients the trailing bits of its binary representations, i.e., all bits after the first 1, are encoded using the refinement coding pass.

Next, we use a toy example of a 4x4 wavelet image produced using two levels of wavelet decomposition, – see Figure 2.15(a), to demonstrate the progressive bit-plane coding using zero-tree based interband and quadtree-based intraband coders. The underlying SAQ in the demonstration examples is depicted in Figure 2.15(b).

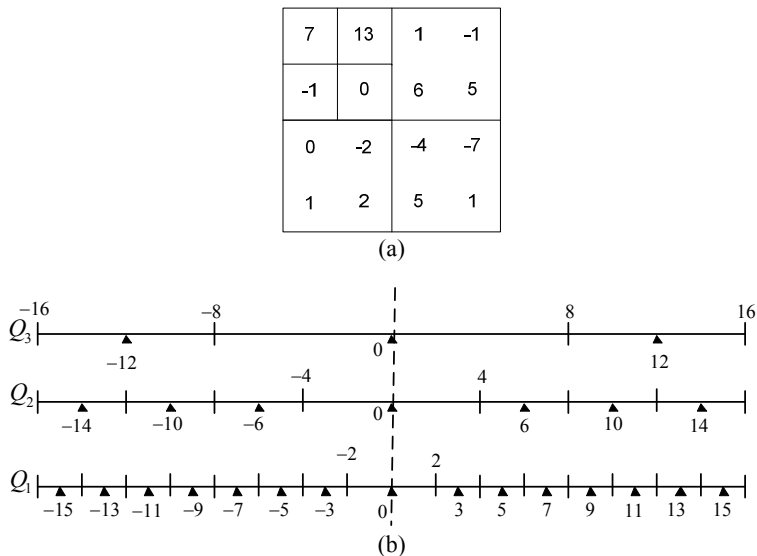


Figure 2.15: (a) A toy example of a 4x4 wavelet image created using two decomposition levels. (b) SAQ used for progressive bit-plane coding of the 4x4 example wavelet image. Only 3 highest quantization levels are shown in the figure.

2.5.5.1 Zero-tree based Interband Bit-plane Coding

In zero-tree based bit-plane coding, trees of 0's (also known as zero-trees) [100] are exploited to achieve efficient compression of the position information of significance coefficients. A tree of 0's in the significance map exists if a certain wavelet coefficient is non-significant with respect to a certain threshold and all its descendant wavelet coefficients (see Figure 2.14 for the definition of descendants) are also non-significant with respect to the same threshold. Each tree of 0's can be encoded using a single zero-tree root (ZTR) symbol. Note that, the root-node of a zero-tree may lie in any band HH_i , LH_i , or HL_i , $i > 1$. Thus, a zero-tree must span at least two wavelet decomposition levels. When a coefficient belonging to HH_i , LH_i , or HL_i for $i > 1$, is non-significant with respect to a threshold and one or more of its descendants are significant, then this coefficient is encoded using the isolated zero (IZ) symbol. A non-significant coefficient belonging to HH_1 , LH_1 , or HL_1 is coded using a single zero (Z) symbol. For encoding significance, the significant positive (POS) or the significant negative (NEG) symbols are used. The symbol stream produced using the above coding symbols can be transformed to a binary bit-stream by replacing each symbol with its binary equivalent, which are defined as follows:

1. POS: This symbol is equivalent to a binary 1, indicating that a particular positive wavelet coefficient is significant, followed by the positive sign indicated using a binary 0, so the binary equivalent of POS is 10.

2. NEG: The binary equivalent of this symbol is 11, where, like the POS symbol, the first bit denotes the significance while the second bit indicates the negative sign.
3. ZTR: The binary equivalent of this symbol is 00, where the first 0 indicates that the coefficient is non-significant while the second 0 bit indicates that all corresponding descendants are also non-significant at the current bit-plane level.
4. IZ: The binary equivalent of this symbol is 01, where, like the ZTR symbol, the first 0 bit indicate the non-significance of the coefficient while binary 1 indicates that at least one descendant has a coefficient which is significant at the current bit-plane level. Notice that the Z and IZ symbols have the same binary representation. In fact, the Z symbol is introduced just for a better understanding of the zero-tree operation.

In Figure 2.16, the bit-plane coding process is shown for the toy example of Figure 2.15(a). The highest bit-plane value P is computed as:

$$P = \lfloor \log_2(\max(x)) \rfloor, \quad (2.79)$$

and the corresponding threshold is determined as $T_p = 2^p$. For any lower bit-plane the threshold can be computed as:

$$T_p = \frac{T_P}{2^{P-p}}, \quad 0 \leq p < P. \quad (2.80)$$

For coding the significance map the following subband scanning pattern (proceeding from left to right) is used.

$$LL_J, HL_J, LH_J, HH_J, \dots, HL_2, LH_2, HH_2, HL_1, LH_1, HH_1. \quad (2.81)$$

Within each subband, a raster scan pattern is followed to encode the binary representation of wavelet coefficients using POS, NEG, ZTR, IZ (or Z) symbols.

In Figure 2.16, the coding begins with the significance pass of the MSB plane of the wavelet image. At most-significant bit-plane ($p = 3$), the single bit in LL_2 subband is 0. As the coefficients of LL subband do not have descendants, the first coded symbol is IZ. The single bit in the HL_2 is 1 and its corresponding sign is +, so the second coded symbol is POS. Note that descendants of the coefficient of HL_2 , i.e., the coefficients of HL_1 , are yet to be coded. Proceeding to LH_2 , the single coefficient is 0 and its four descendants in LH_1 are all 0, so the 3rd encoded symbol is ZTR. The same is also true for the only coefficient of HH_2 symbol, thus the 4th coded symbol is also ZTR. This is followed by four Z symbols each corresponding to one coefficient of HL_1 subband. At the end of the significance pass, only a single coefficient (i.e., 13) has been found significant, while all other coefficients are non-significant.

For the next bit-plane, the coefficients which are found significant are first refined

using the refinement pass. In the example case, the refinement symbol for the only significant coefficient (i.e., 13) is 1, which is coded next. For differentiation purposes, in Figure 2.16, the refinement symbols are shown within circles. The significance of all other coefficients except 13 is then coded using the significance pass for $p=2$, followed by the refinement of coefficients found significant for $p=3$ and $p=2$. The coding process proceeds this way till all the bit-planes of the decomposed image are coded. In general, the bit-streams produced by significant and refinement passes are arithmetic coded using separate adaptive models for each type of information [108].

The last column of Figure 2.16 shows the reconstructed wavelet image after progressively decoding each bit-plane. In general, the decoding process can also be stopped in between the bit-planes to meet a certain rate requirement. After finding that a given wavelet coefficient is significant, the decoder decodes the wavelet coefficient at bit-plane p with the value $\hat{x}^{(p)}$ as:

$$\hat{x}^{(p)} = \begin{cases} T_p + T_p/2 & \text{if } p \neq 0 \\ 1 & \text{if } p = 0 \end{cases} \quad (2.82)$$

Subsequently, receiving refinement symbols $\eta_p = \{0,1\}$ at any level p results in the reconstructed value:

$$\hat{x}^{(p)} = \begin{cases} \hat{x}^{(previous)} + T_p/2 & \text{if } p \neq 0 \text{ and } \eta_p = 1 \\ \hat{x}^{(previous)} - T_p/2 & \text{if } p \neq 0 \text{ and } \eta_p = 0 \\ \hat{x}^{(previous)} + 0 & \text{if } p = 0 \text{ and } \eta_p = 1 \\ \hat{x}^{(previous)} - T_p & \text{if } p = 0 \text{ and } \eta_p = 0 \end{cases} \quad (2.83)$$

where $\hat{x}^{(previous)}$ denotes the absolute value of the reconstructed sample before receiving the refinement symbol η_p . Note that, although the wavelet coefficients are not directly quantized using the SAQ, the above decoding rules comply with SAQ, shown in Figure 2.15(b). In fact, SAQ is built within the coding passes and can be more efficiently implemented using dyadically decaying thresholds [100], i.e.,

$$T_{p-1} = \frac{T_p}{2}, p = 1, 2, \dots, P, \quad (2.84)$$

as demonstrated in Figure 2.16.

Based on the zero-tree bit-plane coding scheme described above, a scalable image coding framework was proposed in [100]. Later in [87], the image coding system of [100] was improved using the set partitioning methods, also known as the SPIHT codec.

| | | | | | | | |
|--|--|----|----|----|----|--|--|
| | | 7 | 13 | 1 | -1 | | |
| | | -1 | 0 | 6 | 5 | | |
| | | 0 | -2 | -4 | -7 | | |
| | | 1 | 2 | 5 | 1 | | |

| Threshold | Bit-plane | | | | Encoded Symbols | Decoded Image | | | |
|----------------------|-----------|---|---|---|--|---------------|----|----|----|
| Threshold=8 $p=3$ | 0 | 1 | 0 | 0 | Significance Pass: IZ, POS, ZTR, ZTR, Z, Z, Z, Z | 0 | 12 | 0 | 0 |
| | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| Threshold=4 $p=2$ | 1 | 1 | 0 | 0 | Refinement Pass: 1 Values: 14 Significance Pass: POS, ZTR, IZ, Z, Z, POS, POS, NEG, NEG, POS, Z | 6 | 14 | 0 | 0 |
| | 0 | 0 | 1 | 1 | | 0 | 0 | 6 | 6 |
| | 0 | 0 | 1 | 1 | | 0 | 0 | -6 | -6 |
| | 0 | 0 | 1 | 0 | | 0 | 0 | 6 | 0 |
| Threshold=2 $p=1$ | 1 | 0 | 0 | 0 | Refinement Pass: 0, 1, 1, 0, 0, 1, 0 Values: 13, 7, 6, 5, -4, -7, 5 Significance Pass: IZ, IZ, Z, Z, Z, NEG, Z, POS, Z | 6 | 13 | 0 | 0 |
| | 0 | 0 | 1 | 0 | | 0 | 0 | 7 | 5 |
| | 0 | 1 | 0 | 1 | | 0 | -3 | -5 | -7 |
| | 0 | 1 | 0 | 0 | | 0 | 3 | 5 | 0 |
| Threshold=1 $p=0$ | 1 | 1 | 1 | 1 | Refinement Pass: 1, 1, 0, 1, 0, 1, 1, 0, 0 Values: 13, 7, 6, 5, -4, -7, 5, -2, 2 Significance Pass: NEG, IZ, POS, NEG, Z, POS, POS | 7 | 13 | 1 | -1 |
| | 1 | 0 | 0 | 1 | | -1 | 0 | 6 | 5 |
| | 0 | 0 | 0 | 1 | | 0 | -2 | -4 | -7 |
| | 1 | 0 | 1 | 1 | | 1 | 2 | 5 | 1 |

Figure 2.16: Interband zero-tree based bit-plane coding of the 4x4 example wavelet image. The coding of each bit-plane is shown using two coding passes.

2.5.5.2 Quadtree-based Intra-band Bit-plane Coding

In quadtree-based bit-plane coding the significant wavelet coefficients at a certain bit-plane p are localized using quadtree [36] splitting of the significance map. In particular, if one or more wavelet coefficients are significant at a certain level p , the significance map is split into four equally sized quadrants. Each quadrant is further split into four equal size sub-quadrants if it contains one or more significant coefficients. If all the coefficients inside a quadrant are non-significant no quadtree splitting is performed. The process is continued till each significant wavelet coefficient is localized to a single wavelet coefficient. The significance map of the 4x4 image of Figure 2.15(a) at bit-plane $p = 3$ is shown in Figure 2.17(a). The quadtree decomposition of the significance map of Figure 2.17(a) is depicted in Figure 2.17(b), where the significant and the non-significant leaves of the tree are represented using SGN and NSG coding symbols, respectively.

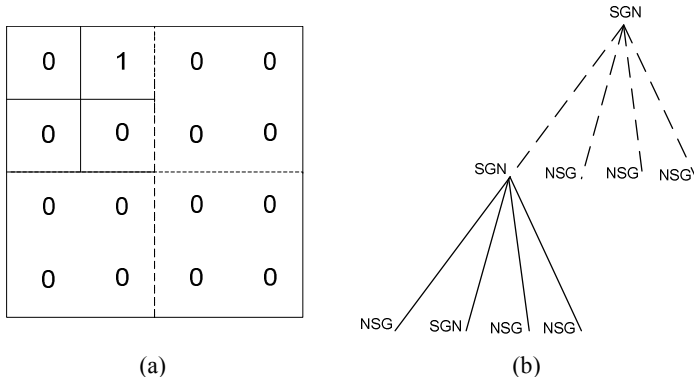


Figure 2.17: (a) Significance map of the 4x4 wavelet image of Figure 2.15(a) at $p = 3$. (b) Quadtree splitting of the significance map of (a). The dashed and solid lines differentiate between the levels of quadtree decomposition.

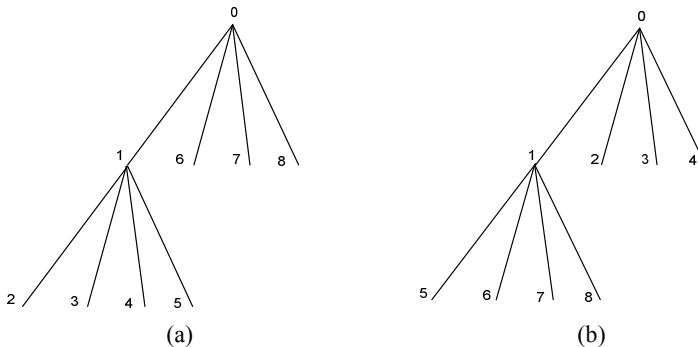


Figure 2.18: (a) Depth-first tree scanning pattern. (b) Breadth-first tree scanning pattern. The numbering determines the order in which the nodes of the tree are scanned.

For each bit-plane p , encoding the position of the significance coefficients translates to encoding the corresponding quadtree structure. A fixed tree scanning rule needs to be known both at the encoder and the decoder so that the tree (and hence the position of the significant coefficients) can be perfectly reconstructed at the decoder side. In general, either the depth-first or the breadth-first search methods, shown in Figure 2.18, are used to scan the quadtree nodes.

For any given intermediate tree-node the corresponding children nodes at the lower level are scanned using a fixed scanning order. For example, for any quadrant of a given intermediate tree-node, the corresponding sub-quadrants can be scanned in the order: top-left \rightarrow top-right \rightarrow bottom-left \rightarrow bottom-right.

In Figure 2.19, the significance pass for the MSB-plane of the considered example, i.e., the example of Figure 2.15(a), is shown. The depth-first tree scanning of Figure 2.18(a) is used to enumerate the quadtree nodes. In general, the depth-first tree scanning gives better encoding performance against the breadth-first scanning [76]. Note that, similar to zero-tree coding, the sign information is coded along with coding the significance of a particular wavelet coefficient. For the next lower bit-plane, the non-significant quadrants at the current bit-plane will be further quadtree decomposed and the significance of their wavelet coefficients is encoded in the significance pass. The significance pass is followed by the refinement pass, which encodes the refinement information of the coefficients that were found to be significant at the current bit-plane level – see Figure 2.19. The refinement pass in both zero-tree and quadtree coding schemes are exactly the same.

From a progressive image coding point of view, coding the significance of newly significant coefficients reduces the image distortion at a faster rate than refining the already significant coefficients [76]. For this reason, in quadtree coding, the refinement pass follows the significance pass when coding any bit-plane of image wavelet coefficients. Note that this is in contrast to zero-tree coding, which codes the refinement information prior to the significance information. The last column of Figure 2.19 shows the reconstructed wavelet coefficients after progressively decoding each bit-plane. The SAQ reconstruction rules of equations (2.82)-(2.83) are also valid for quadtree-based bit-plane coding. Note that for any level p , the decoded wavelet images for both zero-tree coding and quadtree coding are exactly the same – see the last column in Figure 2.16 and Figure 2.19, resulting in equal distortions for both cases when equal number of bit-planes are decoded. However, the encoding rates can be different as different coding strategies are utilized.

Based on the coding principle described in Figure 2.19, a square partitioning (SQP) image coder was proposed in [74]. An extension of the SQP coding is the wavelet-quadtree coding (WQT) approach of [75]. The difference between WQT

| | | | | | | | | | |
|--|--|--|--|----|----|----|----|--|--|
| | | | | 7 | 13 | 1 | -1 | | |
| | | | | -1 | 0 | 6 | 5 | | |
| | | | | 0 | -2 | -4 | -7 | | |
| | | | | 1 | 2 | 5 | 1 | | |

| | Bit-plane | | | | Encoded Symbols | Decoded Image | | | |
|----------------------|-----------|---|---|---|--|---------------|----|----|----|
| Threshold=8 $p=3$ | 0 | 1 | 0 | 0 | Significance Pass: SGN SGN NSG, SGN, +, NSG, NSG NSG, NSG, NSG | 0 | 12 | 0 | 0 |
| | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |
| Threshold=4 $p=2$ | 1 | 1 | 0 | 0 | Significance Pass: SGN, +, NSG, NSG SGN NSG, NSG, SGN, +, SGN, + NSG SGN SGN, -, SGN, -, SGN, +, NSG Refinement Pass: 1 Values: 14 | 6 | 14 | 0 | 0 |
| | 0 | 0 | 1 | 1 | | 0 | 0 | 6 | 6 |
| | 0 | 0 | 1 | 1 | | 0 | 0 | -6 | -6 |
| | 0 | 0 | 1 | 0 | | 0 | 0 | 6 | 0 |
| Threshold=2 $p=1$ | 1 | 0 | 0 | 0 | Significance Pass: NSG, NSG NSG, NSG SGN NSG, SGN, -, NSG, SGN, + NSG Refinement Pass: 0, 1, 1, 0, 0, 1, 0 Values: 13, 7, 6, 5, -4, -7, 5 | 6 | 13 | 0 | 0 |
| | 0 | 0 | 1 | 0 | | 0 | 0 | 7 | 5 |
| | 0 | 1 | 0 | 1 | | 0 | -3 | -5 | -7 |
| | 0 | 1 | 0 | 0 | | 0 | 3 | 5 | 0 |
| Threshold=1 $p=0$ | 1 | 1 | 1 | 1 | Significance Pass: SGN, -, NSG SGN, +, SGN, - NSG, SGN, + SGN, + Refinement Pass: 1, 1, 0, 1, 0, 1, 1, 0, 0 Values: 13, 7, 6, 5, -4, -7, 5, -2, 2 | 7 | 13 | 1 | -1 |
| | 1 | 0 | 0 | 1 | | -1 | 0 | 6 | 5 |
| | 0 | 0 | 0 | 1 | | 0 | -2 | -4 | -7 |
| | 1 | 0 | 1 | 1 | | 1 | 2 | 5 | 1 |

Figure 2.19: Intra-band quadtree-based bit-plane coding of the 4x4 example wavelet image. The vertical lines “|” indicate the transitions between the quadtree levels.

and SQP coding is that instead of two coding passes WQT uses three coding passes, namely, the significance, the non-significance and the refinement pass. Basically, during the significance pass of an arbitrary bit-plane, a certain neighborhood of the significant coefficients is identified and added in the list of non-significant coefficients (LNC). For the next lower bit-plane, the significance of the coefficients found in LNC is coded first. This is motivated by the fact that the coefficients recorded in LNC (being located in the neighborhood of the significant coefficients) have a high probability to become significant in the next coding step, due to the clustering property of the quadtree decomposition [74].

A further extension of SQP [74] and WQT [75] is the quadtree-limited (QT-L) coding technique proposed in [94]. The difference between them is that for any arbitrary bit-plane, the quadtree decomposition process in QT-L is stopped once the size of the significant quadrants is below a predefined minimal size [94]. In [94], it was shown that a 3-D variation of the QT-L coding coupled with context-based adaptive arithmetic coding yields competitive results against state-of-the-art coding systems for volumetric medical imaging datasets. In [76], a 2-D version of QT-L coding was evaluated for scalable compression of photographic images. Compared to JPEG-2000 [108], the QT-L based image codec gives competitive and in many cases better image compression results [76].

In contrast to zero-tree based bit-plane coding, which employs zero-trees that span different resolution subbands, quadtree-based bit-plane coding, realized by the SQP, WQT and QT-L codecs, can independently work on each frequency subband. In fact, due to this particular property, intraband coding schemes can enable resolution scalable decoding of wavelet coded images. In resolution scalable coding, different subbands are independently bit-plane coded in the order of their frequency, i.e., by following the coding order $LL_J, HL_J, LH_J, HH_J, HL_{J-1}, LH_{J-1}, HH_{J-1}, \dots$. In particular, all the bit-planes of a certain wavelet subband are coded before moving to the next higher frequency subband in the aforementioned order. At the decoder side, terminating the decoding process at any point in time will leave a certain resolution of the image decoded at a certain quality level. In general, with this type of bit-stream organization the decoder can decode the image at multiple resolution levels and even at multiple quality levels for a given resolution level. Hence, such a bit-stream organization allows for a greater flexibility to accommodate displays that vary in terms of resolution and complexity requirements. Note that such a flexibility cannot be guaranteed using zero-tree based systems due to the simple fact that bit-planes of different frequency subbands in zero-tree based methods are jointly coded in order to provide quality scalable decoding. In contrast, quality scalability is by default offered when using quadtree-based bit-plane coding

approaches [76]. Hence, one can conclude that quadtree-based coding extends over the scalability features of zero-tree coding, while still providing superior D-R performance in the quality scalable mode [76].

2.5.6 Practical Wavelet-based Scalable Image Coding

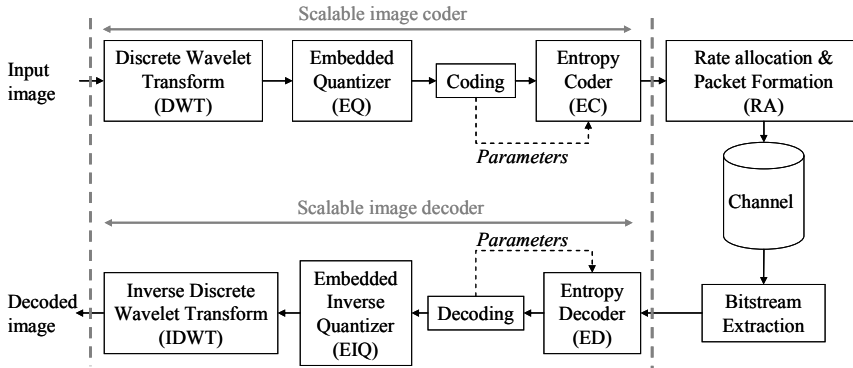


Figure 2.20: Wavelet-based scalable image codec.

Figure 2.20 depicts the block diagram of a wavelet-based scalable image codec. Practical transform-based image coding systems rely on subsequent application of an invertible transform, embedded quantization, bit-plane coding and entropy coding components [108], functions that have already been explained in the previous parts of this chapter. In Section 2.5.5, it was explained how bit-planes produced by an embedded quantizer are bit-plane coded. Bit-plane coding methods result in a binary bit-stream which is further losslessly compressed generally using adaptive arithmetic coding [123]. Entropy coding techniques exploit the inter-symbol correlation in the bit-plane coded bit-stream to further improve the compression efficiency of an image coding system. In practical image codecs, e.g., [94, 100, 107], bit-plane coding and entropy coding work in tandem. Namely, the entropy coder can encode symbols as soon as they appear at the output of the bit-plane coder.

A function that has not yet been discussed is the rate-allocation module in Figure 2.20. In general, for quality scalability, bit-planes can be simply coded in the order of their significance. However, such a predefined progression order may not be optimal in rate-distortion sense since bit-planes of different frequency subbands have a different impact on the overall image distortion. In other words, in some cases it may be beneficial to code different number of bit-planes from different subbands to achieve minimal distortion for a given bit budget. Thus, one can always make use of an additional level of prioritization to ensure a minimum image distortion for a given rate budget. This is the functionality provided by the rate-

allocation module. Without going into the details, we point out that the rate-allocation module computes the optimal truncation points for the bit-streams of non-overlapping disjoint spatial parts of the wavelet decomposed structure. These disjoint parts can be the distinctive frequency subbands or blocks of a certain fixed-size. For fixed-size blocks, a rate-allocation algorithm, known as the post compression rate-distortion optimization (PCRD-OPT) was proposed in [107]. The PCRD-OPT algorithm utilizes Lagrangian based methods to optimally compute the truncation point for each coding block, so that the overall image distortion is minimized. The final bit-stream is created using bit-stream contributions from different coding blocks. For details on PCRD-OPT, we refer to [107, 108].

The decoder part of a scalable image coder successively applies entropy decoding, bit-plane decoding and the inverse wavelet transform to reconstruct the compressed image.

In the next chapter, we will study the embedded quantization of coefficient subbands resulting from wavelet decomposition of a semi-regular 3-D mesh. With certain modifications, the tree-based bit-plane coding methods, described in Section 2.5.5, can also be used to encode the bit-planes of mesh wavelet subbands. Despite of the fact that 3-D meshes and images have different physical structure, efficient scalable compression for 3-D meshes (similar to images) demands (a) ensuring a lowest distortion at any intermediate bitrate, (b) the need to provide quality and/or resolution scalable decoding of the compressed mesh.

2.6 SUMMARY

Source coding, either lossless or lossy, is a well-developed process both in theory and in practice. The fundamental limits for both variations are well known in theory and are described in this chapter in order to provide some background knowledge to the reader who is not (or less) familiar with the concepts of source coding.

For lossless source coding, the fundamental lower rate limit is the entropy of a source, as proved by Shannon in his landmark paper [98] in 1948. Since then, a considerable amount of effort has been reserved to find practical source codes that can attain this limit. In this context, some well-known source codes, e.g., the Shannon-Fano, the Huffman and the arithmetic code, are briefly described in this chapter. As opposed to the Huffman code, arithmetic codes provide enhanced performance for binary sources. Additionally, arithmetic codes allow for easy source adaptation during the coding process. This property makes them the most suitable choice for entropy coding in practical multimedia coding standards, e.g., the JPEG-2000 [108] image coding standard and the MPEG AVC/SVC video

coding standard [2].

For lossy source coding, the fundamental limit is the source's $D(R)$ function, which was also first derived and explained by Shannon in [99]. In general, lossy source coding is performed by source quantization followed by entropy coding. Scalar quantization is the simplest way to perform quantization for which Lloyd and Max first derived the necessary conditions for optimality. Quantization can be classified into fixed-rate and embedded quantization. In the latter, quantization cells are embedded such that each level of quantization successively reduces the quantization noise. In this context, SAQ is the foremost quantization method forming the basis of wavelet-based scalable image coding systems. Based on SAQ, zero-tree based interband and quadtree-based intraband coding systems have been proposed in the past. These systems are also briefly described in this chapter. The state-of-the-art scalable image coding system JPEG-2000 is also an intraband system. Compared to interband image coding, intraband systems on average provide superior compression performance with the additional advantage of providing resolution scalable decoding of the compressed image.

Chapter 3

SCALABLE 3-D MESH COMPRESSION

3.1 INTRODUCTION

3-D graphics applications make use of polygonal 3-D meshes for object's shape representation. The recent introduction of high-performance laser scanners and fast microcomputer systems gave rise to high-definition graphics applications. In such applications, objects with complex textures are represented using dense 3-D meshes that consist of hundreds of thousands of vertices. Due to their enormous data size, such highly-detailed 3-D meshes are rather intricate to store, costly to transmit via bandwidth-limited transmission media and hard to display on end-user terminals with diverse display capabilities. Scalable compression, wherein the source representation can be adapted to the users' requests, available bandwidth and computational capabilities, is thus of paramount importance in order to make efficient use of the available resources to process, store and transmit high-resolution meshes.

State-of-the-art scalable mesh compression systems can be divided into two main categories. A first category includes codecs that directly compress the irregular topology meshes in the spatial domain. In such codecs, the connectivity information is encoded losslessly, while mesh simplification methods such as vertex coalescing [86], edge decimation [103] and edge collapsing [85] are employed to encode geometry. These mesh simplification methods progressively remove those mesh vertices which yield the smallest distortion. In order to enable the reconstruction of the original mesh at various levels of detail (LOD), the discarded vertices are encoded in the compressed bit-stream. Mesh compression systems belonging to this category include the progressive mesh coding schemes of [59, 82] and the topological surgery [106] based technique of [105]. These techniques generally exhibit two major drawbacks: first, due to the highly irregular topology of the input mesh, a large source rate is needed for lossless encoding of connectivity. Secondly, encoding the removed vertices in the compressed bit-stream is quite costly for high-resolution meshes. Therefore, such schemes are not useful for complex meshes

containing a large number of vertices. An alternative that solves the problem of the large source rates needed to encode the connectivity information, described above, is *remeshing*, which can be used to convert the original irregular mesh into a mesh consisting of regular elements, such as *B-spline* [33] or subdivision connectivity (wherein the connectivity can be realized using a predefined subdivision method) patches [32]. The regular mesh lends itself better to compression, and hence, compared to the irregular mesh, a much lower rate is needed to losslessly encode its connectivity information. Furthermore, multiresolution techniques alleviate the second problem of having to encode all the original vertices, because only detail information has to be encoded in order to create multiple LOD (or multiple resolution levels). Remeshing together with subdivision-based multiresolution [67] are the two major components of the second category of codecs which use space-frequency dilation methods such as wavelet transforms to decorrelate the input mesh data [67], [95]. The generated wavelet coefficients are compressed using tree-based bit-plane coding methods [100], [94], to achieve high compression efficiency. Multiresolution mesh compression techniques provide substantial compression gains compared to their competing schemes. In this chapter we will confine our discussion only to these techniques.

In the recent past, several multiresolution scalable mesh compression schemes have been proposed. The majority of these schemes use coding techniques specifically developed for image compression. However, in general, image and mesh data exhibit different statistical characteristics as images are consisting of pixels (with intensities), while mesh data involve geometry, i.e., the positions of vertices in a 3-D space. Thus, one must be cautious when extrapolating image compression techniques towards mesh geometry encoding.

In this chapter, we propose a constructive design methodology for multiresolution-based scalable mesh compression systems. The input mesh is assumed to possess subdivision connectivity [67], i.e., the connectivity in the mesh is built through subdivision⁵. A 3-D mesh with subdivision connectivity is also referred to as a semi-regular mesh. With respect to *design*, we address two major aspects of scalable wavelet-based mesh compression systems, namely, the optimality of embedded quantization in scalable mesh coding and the type of coefficient dependencies that can assure the best compression performance. In this context, thorough analyses investigating the aforementioned aspects are carried out to establish the most appropriate design choices. Later on, the derived design choices

⁵ In general, an initial remeshing step [32] is required to convert the original irregular mesh into a mesh with the required connectivity.

are integrated as components of the scalable mesh coding system to achieve state-of-the-art compression performance.

The remainder of this chapter is organized as follows. In Section 3.2, a brief overview of multiresolution analysis of the mesh geometry is given. Section 3.3 presents a model-based theoretical investigation of optimal embedded quantization in wavelet-based coding of 3-D semi-regular meshes. An information theoretic analysis of the statistical dependencies among wavelet coefficients and the conclusions regarding the best exploitable statistical dependency are detailed in Section 3.4. Section 3.5 gives an overview of the current state-of-the-art and proposed mesh compression systems.

3.2 MULTIREOLUTION ANALYSIS OF SEMI-REGULAR MESHES

A 3-D mesh $M = \{c, p\}$ is generally represented as a set of two components, a vertex list c and a polygon list p . c is a matrix whose i th row c_i contains the x , y and z position of the i th vertex, i.e., $c_i = [c_{i,x}, c_{i,y}, c_{i,z}]$. p is a list of polygons made up of edges where each edge is a line connecting two vertices. In computer graphics, 3-D meshes are constructed using different polygonal shapes, e.g., triangles, rectangles, etc. However, in this chapter, we will confine our discussion only to *triangular meshes*. Figure 3.1 shows few examples of 3-D meshes we will encounter in this chapter.

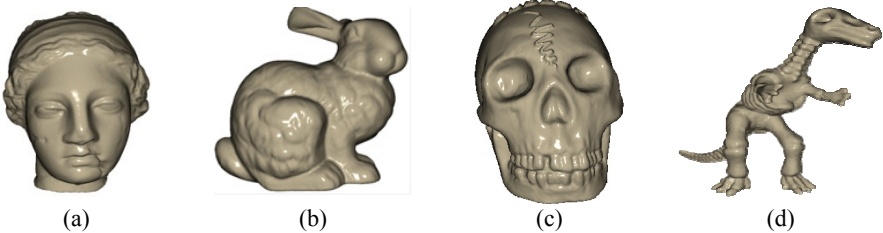


Figure 3.1: Few examples of irregular 3-D meshes. (a) Venus, 50002 vertices (b) Bunny, 34835 vertices (c) Skull, ~20000 vertices (d) Dino, ~14000.

The *valence* of a vertex is defined as the number of edges incident on that vertex. For a regular mesh, all vertices have valence six. In general, the scanned 3-D meshes are not regular. However, they can be efficiently converted to semi-regular equivalents by making use of the remeshing tools, e.g., [32], [52]. Figure 3.2(a) shows the semi-regular equivalents of 3-D meshes shown in Figure 3.1, E_r denotes the remeshing error with respect to the irregular mesh, measured using the Hausdorff distance measure [8]. For semi-regular meshes, which are built by

successive *triangle quadrisection*⁶ starting from a low resolution irregular mesh, almost all vertices have valence six (i.e., except the vertices of the irregular mesh). Due to the application of successive quadrisection, the final semi-regular mesh can be subsampled using the inverse-quadrisection to achieve a lower resolution version of the actual semi-regular mesh – see Figure 3.2(b).

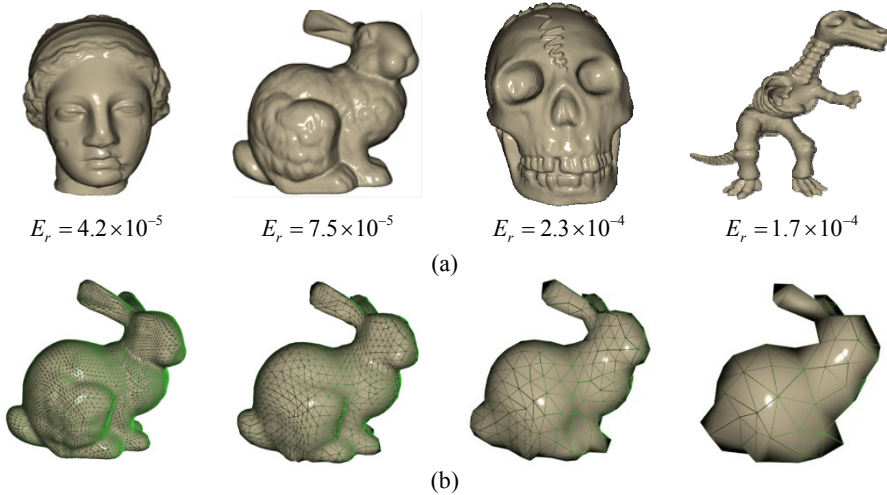


Figure 3.2: (a) Semi-regular equivalents of 3-D meshes shown in Figure 3.1. (b) Three levels of subsampling of the semi-regular Bunny using inverse-quadrisection.

In the following, a brief theoretical overview of semi-regular multiresolution analysis is presented. Later on, two practical transforms, namely the *lifting-based wavelet transform* and the *spatially adaptive wavelet transform* are detailed.

3.2.1 Theory

3.2.1.1 Subdivision Surfaces

Subdivision is a process of iteratively refining a control polyhedron M^0 into fine geometry polyhedra such that the refined polyhedra $M^1, M^2, M^3 \dots$ converge to a limit surface M^∞ . In general, subdivision schemes consist of *splitting* and *averaging* steps. In the *splitting* step, each triangular face is split into four sub-triangles by adding new vertices. This way, an intermediate polyhedron \bar{M}^j is created for any level j . The *averaging* step is used to determine the position of each vertex in M^j from its local neighborhood of vertices in \bar{M}^j , $j=1,2,\dots,J$. $P^j \in \mathbb{R}^{N_{j+1} \times N_j}$ and $Q^j \in \mathbb{R}^{N_{j+1} \times N_{j+1}}$ (where N_j denotes the number of vertices of M^j) are splitting and averaging matrices at level j . The subdivision process, expressed

⁶ Triangle quadrisection is a process in which each triangle is split into four sub-triangles by adding new vertices. One new vertex is added at the middle of each edge.

in the matrix form, can be written as:

$$\mathbf{c}^{j+1} = \mathbf{Q}^j \cdot \mathbf{P}^j \cdot \mathbf{c}^j, \quad j = 0, 1, 2, \dots, J-1.$$

Subdivision schemes can be regarded as either *interpolating* or *approximating*. In the interpolating subdivision, only the positions of new vertices, i.e., which are added in the splitting step, are changed in the averaging step. On the other hand, in the *approximating* subdivision, the positions of both newly introduced and existing vertices are adjusted during averaging. The simplest interpolating subdivision is the *polyhedral subdivision* in which the averaging step leaves all new vertices unmoved. However, the limiting surface in this case is the same as M^0 . A commonly used interpolating subdivision that can also produce a smooth surface is the Butterfly subdivision [31]. The subdivision stencil for Butterfly is shown in Figure 3.3, where the position of a newly introduced vertex p is computed as $c_p = \sum_{i=1}^8 a_i c_i$, where a_i 's denote the Butterfly weights [31], given by:

$$\left\{ \begin{array}{l} a_1 = a_2 = 1/2 \\ a_3 = a_4 = 1/8 \\ a_5 = a_6 = a_7 = a_8 = -1/16 \end{array} \right\}.$$

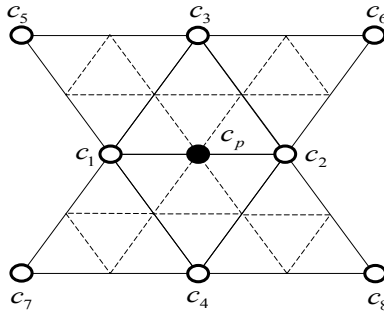


Figure 3.3: Butterfly subdivision stencil.

Loop [66] is an example of approximating subdivision. For Loop subdivision, the new vertices are introduced at the mid-points of triangle edges. Let $\{\bar{c}_i\}_{i=1}^n$ denote the neighboring vertices of a vertex \bar{c} in the split mesh \bar{M} , then for the Loop scheme, the modified position of \bar{c} can be computed using the following averaging relation:

$$c = \frac{\alpha(n)\bar{c} + \bar{c}_1 + \bar{c}_2 + \dots + \bar{c}_n}{\alpha(n) + n}, \quad (3.1)$$

where

$$\alpha(n) = \frac{n(1 - \beta(n))}{\beta(n)} \quad \text{and} \quad \beta(n) = \frac{5}{4} - \frac{(3 + 2 \cos(2\pi/n))^2}{32}.$$

The function $\alpha(n)$ is chosen carefully such that the limit surface has a continuous tangent plane [53, 66]. Catmull-Clark [16], Doo-Sabin [29, 30], and Chaikin [17] are among the other commonly used subdivision schemes for 3-D meshes.

3.2.1.2 Multiresolution Analysis

Lounsbery [67] first invented multiresolution analysis for arbitrary topology semi-regular surfaces using subdivision. He proved that refinable basis exist when a coarse mesh M^0 is refined through subdivision, i.e.,

$$\boldsymbol{\phi}^j(\mathbf{x}) = \boldsymbol{\phi}^{j+1}(\mathbf{x}) \cdot \mathbf{P}^j, \text{ for } \mathbf{x} \in M^0 \text{ and } 0 \leq j < J. \quad (3.2)$$

$\boldsymbol{\phi}^j(\mathbf{x})$ in the above equation denotes the row vector of scaling functions ϕ_i^j . Given these refinable scaling functions, scalar-valued function spaces associated with the coarsest geometry M^0 are defined as [67]:

$$V^j(M^0) := \text{Span}(\boldsymbol{\phi}^j(\mathbf{x})), \text{ for } 0 \leq j \leq J. \quad (3.3)$$

Equation (3.2) implies that these spaces are indeed nested, i.e.,

$$V^0(M^0) \subset V^1(M^0) \subset V^2(M^0) \subset \dots, \quad (3.4)$$

The wavelet space $W^j(M^0)$ is defined as a space that is the orthogonal complement of $V^j(M^0)$ in $V^{j+1}(M^0)$. Hence, $W^j(M^0)$ and $V^j(M^0)$ together can represent any scalar-valued piecewise function in the space $V^{j+1}(M^0)$. If $\boldsymbol{\psi}^j(\mathbf{x})$ is a row vector containing refinable basis functions of $W^j(M^0)$, the following stands [67]:

$$\boldsymbol{\psi}^j(\mathbf{x}) = \boldsymbol{\phi}^{j+1}(\mathbf{x}) \cdot \mathbf{Q}^j, \text{ for } \mathbf{x} \in M^0 \text{ and } 0 \leq j < J. \quad (3.5)$$

Combining (3.2) with (3.5) yields:

$$(\boldsymbol{\phi}^j(\mathbf{x}), \boldsymbol{\psi}^j(\mathbf{x})) = \boldsymbol{\phi}^{j+1}(\mathbf{x}) \cdot (\mathbf{P}^j, \mathbf{Q}^j), \text{ or } (\boldsymbol{\phi}^j(\mathbf{x}), \boldsymbol{\psi}^j(\mathbf{x})) \cdot (\mathbf{P}^j, \mathbf{Q}^j)^{-1} = \boldsymbol{\phi}^{j+1}(\mathbf{x}). \quad (3.6)$$

A set of scaling functions $\boldsymbol{\phi}^{j+1}(\mathbf{x})$ can then be used to decompose a surface S^{j+1} in $V^{j+1}(M^0)$ [67]:

$$S^{j+1}(\mathbf{x}) = \sum_i c_i^{j+1} \phi_i^{j+1} = \boldsymbol{\phi}^{j+1}(\mathbf{x}) \cdot \mathbf{c}^{j+1}, \quad (3.7)$$

where, c_i^{j+1} is the i th vertex in M^{j+1} . Since the analysis filters, \mathbf{A}^j and \mathbf{B}^j , are uniquely determined by the relationship:

$$\begin{pmatrix} \mathbf{A}^j \\ \mathbf{B}^j \end{pmatrix} = (\mathbf{P}^j | \mathbf{Q}^j)^{-1}, \quad (3.8)$$

by combining equation (3.6) and equation (3.7) and making the above substitution for $(\mathbf{P}^j | \mathbf{Q}^j)^{-1}$, we obtain:

$$S^{j+1}(\mathbf{x}) = \boldsymbol{\phi}^j(\mathbf{x}) \cdot \mathbf{A}^j \cdot \mathbf{c}^{j+1} + \boldsymbol{\psi}^j(\mathbf{x}) \cdot \mathbf{B}^j \cdot \mathbf{c}^{j+1}. \quad (3.9)$$

From equation (3.9), equation (3.10) derives the forward wavelet transform, given by:

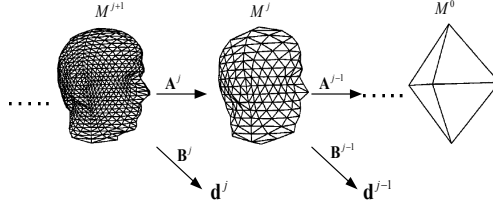


Figure 3.4: Pictorial representation of the forward wavelet decomposition, [67].

$$\mathbf{c}^j = \mathbf{A}^j \cdot \mathbf{c}^{j+1}, \quad \mathbf{d}^j = \mathbf{B}^j \cdot \mathbf{c}^{j+1}, \quad \forall j: 0 \leq j < J, \quad (3.10)$$

where, \mathbf{d}^j is a matrix containing the wavelet coefficients for the j th level of the transform. \mathbf{A}^j 's and \mathbf{B}^j 's are matrices representing the low and the high-pass filters, respectively, also referred to as the *analysis* filter pairs.

A similar reasoning as for equation can be used to formulate the inverse wavelet transform, expressed by:

$$\mathbf{c}^{j+1} = \mathbf{P}^j \cdot \mathbf{c}^j + \mathbf{Q}^j \cdot \mathbf{d}^j, \quad \text{for } \forall j: 0 \leq j < J. \quad (3.11)$$

Hence, \mathbf{P}^j 's and \mathbf{Q}^j 's jointly form the *synthesis* part of the decomposition for the lossless reconstruction of the input semi-regular mesh M^J . Note that the computation of the \mathbf{A}^j 's and \mathbf{B}^j 's involves the inversion of a large matrix, which makes the forward transform more complex than the inverse transform.

In general, after the transform, a fair amount of correlation still exist between x , y and z coordinate components of a wavelet coefficient. To reduce this correlation, the displacement vectors $d_k^j \in \mathbf{d}^j$ are represented in a local coordinate system, referred to as the *local frame representation* [128]. The local coordinate system to represent a wavelet coefficient at any level j is built using the parent edge in the coarser mesh M^{j-1} and the mean normal of the two triangles sharing this edge [57]. After the local frame representation, each wavelet coefficient consists of a *normal* and two *tangential* components.

We point out that the Venus and the Bunny meshes shown in Figure 3.2(a) are the examples of *non-normal* semi-regular meshes while the Skull and the Dino are the *normal* semi-regular meshes [52]. For non-normal meshes, the variance of the normal component is on average twice as large as the variance of two tangential components [57]. For normal meshes, the complete signal energy lies in the local normal direction and almost no energy lies in the tangential components (i.e., the tangential components are often zero). In other words, wavelet coefficients for normal meshes can potentially be represented using a single floating point number, instead of three such numbers for the non-normal meshes [52]. In practice, normal

meshes are created using a special transform-specific remeshing technique [52] which, starting from the base mesh, tries to build a good approximation of the original irregular mesh using only normal (or sometimes quasi-normal) displacement components.

3.2.2 Lifting-based Wavelet Transform

As explained earlier, the filter bank implementation of the multiresolution analysis is quite complex in the sense that the computation of analysis filters involves the computationally intensive inversion of large subdivision matrices. In this context, the lifting-based wavelet implementation [95] provides a low complexity construction of multiresolution methods. In lifting-based multiresolution analysis, each scaling function ϕ_i^j of the j th level exists so that $\{\phi_i^j \mid i \in M^j\}$ is a Riesz basis of $V^j(M^0)$ [95]. The refinement relation for the scaling functions is then:

$$\phi_i^j = \sum_l p_{i,l}^j \cdot \phi_i^{j+1}, \quad (3.12)$$

where, l is the set which defines all linear combination of scaling functions and $p_{i,l}^j$ forms the entries of a matrix similar to \mathbf{P}^j [95]. A similar refinement relation as equation (3.12) is also defined for wavelet functions, i.e., each wavelet function ψ_k^j exists so that $\{\psi_k^j \mid k \in K^j\}$ is a Riesz basis of $W^j(M^0)$ [95]:

$$\psi_k^j = \sum_l q_{k,l}^j \cdot \phi_k^{j+1}. \quad (3.13)$$

K^j and M^j are disjoint sets and they jointly form the scaling function index set of the next higher level, i.e., $M^{j+1} = M^j \oplus K^j$. The lifting-based forward decomposition is expressed by the following relations [95]:

$$\begin{aligned} \forall i \in M^j : c_i^j &= c_i^{j+1} \Rightarrow \text{subsample (i.e., } M^{j+1} \rightarrow M^j) \\ \forall k \in K^j : d_k^j &= c_k^{j+1} - \sum_{i \in M^j} a_i \cdot c_i^j \Rightarrow \text{prediction error (i.e., generate } d^j) \\ \forall k \in K^j : \begin{cases} c_1^j = c_1^j + \tilde{a}_1 \cdot d_k^j \\ c_2^j = c_2^j + \tilde{a}_2 \cdot d_k^j \end{cases} &\Rightarrow \text{update (i.e., modify } M^j \text{ based on } d^j) \end{aligned} \quad (3.14)$$

In the forward transform, the first step is to produce a lower-resolution mesh M^j starting from a higher-resolution version M^{j+1} . The wavelet coefficient d_k^j is the *prediction error* when a high-resolution vertex c_k^{j+1} is predicted based on its low-resolution neighborhood in M^j using a subdivision scheme, such as the Butterfly subdivision. After the prediction, an *update* step is used to modify the low resolution mesh M^j . The update step is carried out on a pair $\{c_1, c_2\}$ of low-resolution vertices joined by a parent edge [95] using the update weights $\{\tilde{a}_1, \tilde{a}_2\}$. The update step modifies the subsampled mesh M^j so as to produce the best low-resolution

approximation of the original input mesh [95]. In general, the prediction and update weights only depend on the connectivity with respect to the vertex to be predicted. However, specific multiresolution analyses, for which the weights depend on the specific resolution level and the underlying geometry, can be also constructed. More details on this are given in the following section.

The inverse transform can be formulated by following the forward-transform steps in the reverse order – see equation (3.15). Namely, the inverse update step is used to form the original subsampled mesh M^j . Using M^j , the prediction of the vertex c_k^{j+1} in K^j is performed and the corresponding prediction error vector d_k^j is added to reconstruct the original position of c_k^{j+1} (the inverse predict step). Finally, M^{j+1} is reconstructed by putting together the vertices of M^j and K^j . Since the connectivity can be exactly reproduced due to subdivision, no ambiguity remains and M^{j+1} can be losslessly reconstructed in the inverse transform.

$$\begin{aligned}
 \forall k \in K^j : & \begin{cases} c_1^j = c_1^j - \tilde{a}_1 \cdot d_k^j \\ c_2^j = c_2^j - \tilde{a}_2 \cdot d_k^j \end{cases} \Rightarrow \text{inverse update} \\
 \forall k \in K^j : & c_k^{j+1} = d_k^j + \sum_{i \in M^j} a_i \cdot c_i^j \Rightarrow \text{inverse predict} \\
 \forall i \in M^j : & c_i^{j+1} = c_i^j \Rightarrow \text{reconstruct (i.e., create } M^{j+1})
 \end{aligned} \quad (3.15)$$

In the course of this thesis we will discuss about non-normal as well as normal meshes. For non-normal meshes, transforms such as Loop, un-lifted Butterfly and lifted-Butterfly are considered. However, for normal meshes, only un-lifted Butterfly transform is considered, simply due to the fact that the original construction of normal meshes is only available for the un-lifted Butterfly transform [52].

3.2.3 Spatially Adaptive Wavelet Transform (SAWT)

As mentioned earlier, lifting-based transforms generally employ fixed prediction weights, independent of the spatial position and geometry around the vertex to be predicted. A simple observation reveals that a better prediction can result from adapting the prediction to the underlying geometry of the mesh. This argument is explained with a simple example: Figure 3.5, referring to the position variable of the vertices, shows a scenario where the vertex to be predicted c_p lies on the straight line joining the vertex pair $\{c_1, c_2\}$, while the remaining six coarser vertices $\{c_i\}_{i=3}^8$ lie on two different planes. In this situation, a prediction function for c_p involving all eight coarser vertices will not be optimal and a better prediction could result by using c_1 and c_2 only. This is logical since c_p lies on the edge formed by the vertex pair $\{c_1, c_2\}$ and is geometrically more correlated to vertices $\{c_1, c_2\}$. Thus, an efficient prediction can be achieved if the prediction process is adapted to the local

mesh geometry. Efficient prediction results in smaller energy of wavelet coefficients and hence an improved compression performance of the mesh coding system. To reverse the prediction operation, the decoder needs to know the weights used by the encoder for the prediction of each vertex c_p . Since additional rate (compared to the classical Butterfly) needs to be spent for coding the prediction weights, the total compression efficiency in the geometry adaptive case is a compromise between the bitrate saved due to the efficient prediction and the extra bitrate needed for signaling the prediction weights.

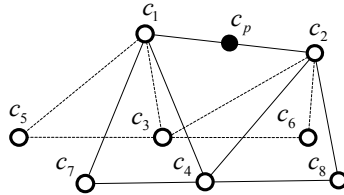


Figure 3.5: Butterfly footprint on an edge.

In the following, a finite set of prediction filters is proposed in the context of spatially-adaptive wavelet transforms (SAWT) [26]. The idea is to use one filter out of this set which best suits the geometry around the vertex to be predicted and which results in the smallest prediction error. A careful application of such an adaptive approach will provide an average rate gain if the reduction in the bitrate due to better prediction dominates the extra bitrate needed to signal the filter type to the decoder [26].

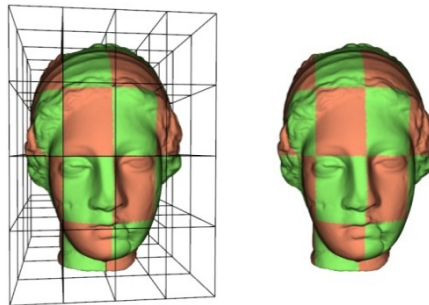


Figure 3.6: Mesh partitioning for $\alpha = 400$. The red and green patches indicate different regions for which different prediction filters will be selected.

In a first step, the input semi-regular mesh is segmented into regions as follows. Let $B(r, s)$ denote the bounding box of the input semi-regular mesh, where $r = (x_B, y_B, z_B)$ and $s = (s_x, s_y, s_z)$ represent the coordinates of the top-left corner and the size vector, respectively. Considering the bounding box as the root cell, each cell on a certain tree level is recursively split into eight equally sized sub-cells to create the next level of the octree. This recursive splitting continues until the number

of vertices in the highest-level cells are smaller than a user-defined threshold α [26]. This way, the semi-regular mesh is divided into regions of approximately the same size – see Figure 3.6.

For each region k , the wavelet analysis is performed by selecting one of the six candidate filters given below:

$$\begin{aligned}
 f_1 &= \frac{1}{2}(c_1 + c_2) + \frac{1}{8}(c_3 + c_4) - \frac{1}{16}(c_5 + c_6 + c_7 + c_8) \Rightarrow (\text{Butterfly}) \\
 f_2 &= \frac{1}{2}(c_1 + c_2) + \frac{1}{4}(c_3 + c_4) - \frac{1}{8}(c_5 + c_6 + c_7 + c_8) \Rightarrow (\text{Modified Butterfly}) \\
 f_3 &= \frac{3}{8}(c_1 + c_2) + \frac{1}{8}(c_3 + c_4) \Rightarrow (\text{Loop}) \\
 f_4 &= \frac{1}{2}(c_1 + c_2) \Rightarrow (\text{edge}) \\
 f_5 &= \frac{1}{2}(c_3 + c_4) \Rightarrow (\text{anti edge}) \\
 f_6 &= \frac{1}{4}(c_1 + c_2 + c_3 + c_4) \Rightarrow (\text{Hybrid of } f_4 \text{ and } f_5)
 \end{aligned}$$

Note that the above set of filters is defined using a mixture of Butterfly, Loop and midpoint subdivision schemes.

Similar to [18], a filter candidate for a particular region k in M^J , is chosen in an optimal D-R manner. More specifically, a predictor for each region k is selected such that the following Lagrangian cost function is minimized:

$$\Lambda_{M^J, k} = \arg \min_{l \in \{1, 2, \dots, 6\}} \left\{ E_k \left[\left(c_p - \tilde{c}_{p, f_l} \right)^2 \right] + \lambda \cdot R_k(f_l) \right\} \quad (3.16)$$

where, $R_k(f_l)$ denotes the necessary rate for encoding the filter index l used for prediction in region k .

In this dissertation, we will confine our discussion to isometric transforms, such as Butterfly or Loop, which do not adapt transform parameters with respect to the underlying geometry of the semi-regular mesh.

3.3 SCALABLE QUANTIZATION

In scalable mesh compression, the wavelet coefficients in the subbands are quantized using a generic family of embedded dead-zone scalar quantizers (EDSQs), see Chapter 2, Section 2.4.3, in which every wavelet coefficient X is quantized to:

$$q_{\xi, n} = \begin{cases} \text{sign}(X) \cdot \left\lfloor \frac{|X|}{\Delta_n} + \xi_n \right\rfloor & \text{if } \frac{|X|}{\Delta_n} + \xi_n > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.17)$$

where $n \in \mathbb{Z}_+$ denotes the quantization level. ξ_n and Δ_n denote the dead-zone control parameter and the step size for any $n \geq 0$, respectively, with $\xi_n = \xi_0/2^n$ and $\Delta_n = 2^n \Delta_0$, where ξ_0 and Δ_0 are the parameters for the highest rate quantizer ($n = 0$). Note that $\xi_0 = 0$ corresponds to the well-known SAQ [100, 108] in which

the dead-zone size is twice the step size Δ_n for any n .

3.3.1 Wavelet Coefficients' Histogram

In general, the observed histogram H_k^j of the k th coordinate component of the j th wavelet subband is symmetric around its center of mass, which is often zero or very close to zero. Moreover, the histogram is peaky around the mean and the frequency of occurrence decays as the magnitude of the coefficient's component increases. For *Rabbit* (*non-normal*) and *Dino* (*normal*) meshes, Figure 3.7 depicts the observed histograms of the normal component of d^{J-3} subband, obtained using the classical Butterfly transform. It is observed experimentally that, in general, $\sigma^2(H_k^{j+1}) < \sigma^2(H_k^j)$ for $1 \leq j < J$.

In the literature, the observed histogram of any component of a wavelet subband is generally modeled using a zero mean generalized Gaussian (GG) distribution [69], expressed by:

$$\forall x \in \mathbb{R} \quad f_{GG}(x, \sigma, \alpha) = \frac{\alpha \nu^{1/\alpha}}{2\Gamma(1/\alpha)} e^{-\nu|x|^\alpha}, \quad (3.18)$$

where α , $\alpha \in (0, 2]$, is the shape control parameter. $\nu > 0$ is the scaling factor and $\nu^{1/\alpha} = \sqrt{\Gamma(3/\alpha)/\sigma^2 \Gamma(1/\alpha)}$, where $\Gamma(\cdot)$ is the Gamma function. Note that, for $\alpha = 1$, equation (3.18) transforms into a zero-mean Laplacian PDF given by:

$$\forall x \in \mathbb{R} \quad f_L(x, \sigma) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}}{\sigma}|x|} = \frac{\lambda}{2} e^{-\lambda|x|} \text{ where } \lambda = \frac{\sqrt{2}}{\sigma}, \quad (3.19)$$

and for $\alpha = 2$ equation (3.18) corresponds to a zero-mean Gaussian PDF.

Although GG distributions closely approach the observed histogram of wavelet coefficients, only approximate rate and distortion expressions for a uniformly quantized GG random variable are known [37]. The extension of these expressions to embedded quantization is not evident as the rate and distortion functions for such distributions are not easily tractable and can only be computed numerically. Moreover, computing these quantities gets very cumbersome due to the slow numerical integration of expressions involving a GG probability function, especially for $\alpha \ll 1$.

3.3.2 Proposed Laplacian Mixture Model

In order to avoid the aforementioned drawbacks of GG distributions, we propose a novel Laplacian mixture (LM) model which not only gives an easy closed-form derivation of the distortion and rate quantities but also better approximates the observed histogram of wavelet coefficients in the majority of cases. The proposed LM is a linear combination of two Laplacian PDFs, i.e.,

$$\forall x \in \mathbb{R} \quad f_{LM}(x) = \beta \cdot f_L(x, \sigma_1) + (1 - \beta) \cdot f_L(x, \sigma_2). \quad (3.20)$$

Note that, $f_{LM}(x)$ indeed defines a probability function, as $\int_{-\infty}^{\infty} f_{LM}(x)dx = 1$.

The LM model is fitted over the observed data using the expectation maximization (EM) algorithm [24] in order to determine the parameters σ_1 , σ_2 and β . By definition, EM algorithm is used to iteratively estimate the *maximum likelihood* or *maximum a posteriori* estimates of the model parameters from the observed data. Let \mathbf{X} denote the set of observed data samples. Say the model is parameterized using a vector of parameters θ , the maximum likelihood estimate θ_{ML} would result from the following maximization problem

$$\begin{aligned}\theta_{ML} &= \arg \max_{\theta} L(\mathbf{X}; \theta), \\ &= \arg \max_{\theta} \sum_{\mathbf{R}} p(\mathbf{X}, \mathbf{R} | \theta),\end{aligned}\quad (3.21)$$

where $L(\mathbf{X}; \theta)$ denote the *likelihood function* [24]. \mathbf{R} in the above equation denotes the set of unobserved or *latent* parameters. The EM algorithm aims to find the maximum likelihood estimate θ_{ML} by iteratively executing the following two steps.

Expectation (E) Step:

The E step calculates the expected value of the log-likelihood function $\log L(\theta; \mathbf{X}, \mathbf{R})$, where $L(\theta; \mathbf{X}, \mathbf{R}) = p(\mathbf{X}, \mathbf{R} | \theta)$. The expectation is taken with respect to the conditional distribution $p(\mathbf{R} | \mathbf{X}, \theta^{(t)})$, i.e.,

$$Q(\theta | \theta^{(t)}) = E_{\mathbf{R} | \mathbf{X}, \theta^{(t)}} [\log L(\theta; \mathbf{X}, \mathbf{R})]. \quad (3.22)$$

Maximization (M) Step:

The M step computes the parameters that maximizes the quantity $Q(\theta | \theta^{(t)})$, i.e.,

$$\theta^{(t+1)} = \arg \max_{\theta} Q(\theta | \theta^{(t)}). \quad (3.23)$$

For the proposed LM model, the E step calculates the two responsibility factors which determine how responsible $f_L(x, \sigma_1)$ or $f_L(x, \sigma_2)$ are for generating the sample x . For each observation x_i , $1 \leq i \leq N$, the two responsibility factors $r_1(i)$, $r_2(i)$ are computed as follows:

$$r_1(i) = \frac{\beta \cdot f_L(x_i, \sigma_1)}{\beta \cdot f_L(x_i, \sigma_1) + (1 - \beta) \cdot f_L(x_i, \sigma_2)}, \quad r_2(i) = \frac{(1 - \beta) \cdot f_L(x_i, \sigma_2)}{\beta \cdot f_L(x_i, \sigma_1) + (1 - \beta) \cdot f_L(x_i, \sigma_2)},$$

The M step updates the parameters to be estimated as:

$$\sigma_m = \sqrt{2 \frac{\sum_{i=1}^N r_m(i) \cdot |x_i|}{\sum_{i=1}^N r_m(i)}}, \quad m=1,2, \quad \text{and} \quad \beta = \frac{1}{N} \sum_{i=1}^N r_1(i).$$

The *E* and *M* steps are executed in tandem till the algorithm achieves minimum KL distance between the observed and model histograms or when there is no significant change in the estimated model parameters between two consecutive iterations. A

better convergence rate is achieved by the initialization condition $\sigma_1^2 = 0.5\sigma_E^2$, $\sigma_2^2 = 2\sigma_E^2$ and $\beta = 0.9$, where σ_E^2 is the estimated data variance. Histogram fitting for GG distributions is done using the brute-force method, where parameters σ_1 , σ_2 and β are exhaustively computed for a minimum KL distance.

3.3.3 Distortion-Rate Function

Closed-form expressions for the output distortion D_L and the output rate R_L of a Laplacian source quantized using an n level EDSQ are derived in the Appendix (at the end of this chapter). In this section, we derive the D-R function for our proposed LM model. Since the distortion is a linear function of the source PDF, the output distortion D_{LM} of the LM PDF for any quantization level n can be written as:

$$D_{LM}(Q_{\delta_n, \Delta_n}) = \beta \cdot D_L(Q_{\delta_n, \Delta_n}) + (1 - \beta) \cdot D_L(Q_{\delta_n, \Delta_n}), \quad (3.24)$$

with $\delta_n = 1 - \xi_n$. The linearity condition does not hold for the output rate R_{LM} since the entropy involves the non-linear $\log(\cdot)$ function. Instead, R_{LM} can be computed as an infinite sum as:

$$p_0 = 2 \int_0^{\delta_n \Delta_n} f_{LM}(x) dx, \quad p_k = \int_{(k-1)\delta_n \Delta_n}^{(k+\delta_n)\Delta_n} f_{LM}(x) dx, \quad k = 1, 2, 3, \dots, \text{ and}$$

$$R_{LM}(Q_{\delta_n, \Delta_n}) = - \sum_{k=-\infty}^{\infty} p_k \log_2 p_k,$$

where p_k denotes the probability mass of the k th quantization cell ($k = 0$ corresponds to the dead-zone cell). Since the LM model is symmetric around its mean, $p_k = p_{-k}$. Note that the PMF p can be exactly computed due to the possibility of analytical integration of $f_{LM}(x)$. For the GG distribution, however, only numerical integration is possible.

3.3.4 Model Validation

This section demonstrates that the proposed LM model is able to approximate the observed histogram and the observed D-R function of 3-D wavelet coefficients more accurately compared to the commonly utilized GG distributions. For comparison purposes, results for the single Laplacian $f_{LM}(\beta = 0)$ case are also reported.

Figure 3.7 illustrates that the proposed mixture model provides a better fitting probability function for the observed histogram compared to the Laplacian and GG distributions. This is especially true for the middle range positive and negative coefficients values – see Figure 3.7. For the Rabbit mesh, LM gives only slightly better fitting than the other two models. However, for Dino, the LM can clearly model the fast decay of the observed histogram more accurately than the GG. The Laplacian PDF in this case only gives a very coarse approximation of the observed histogram.

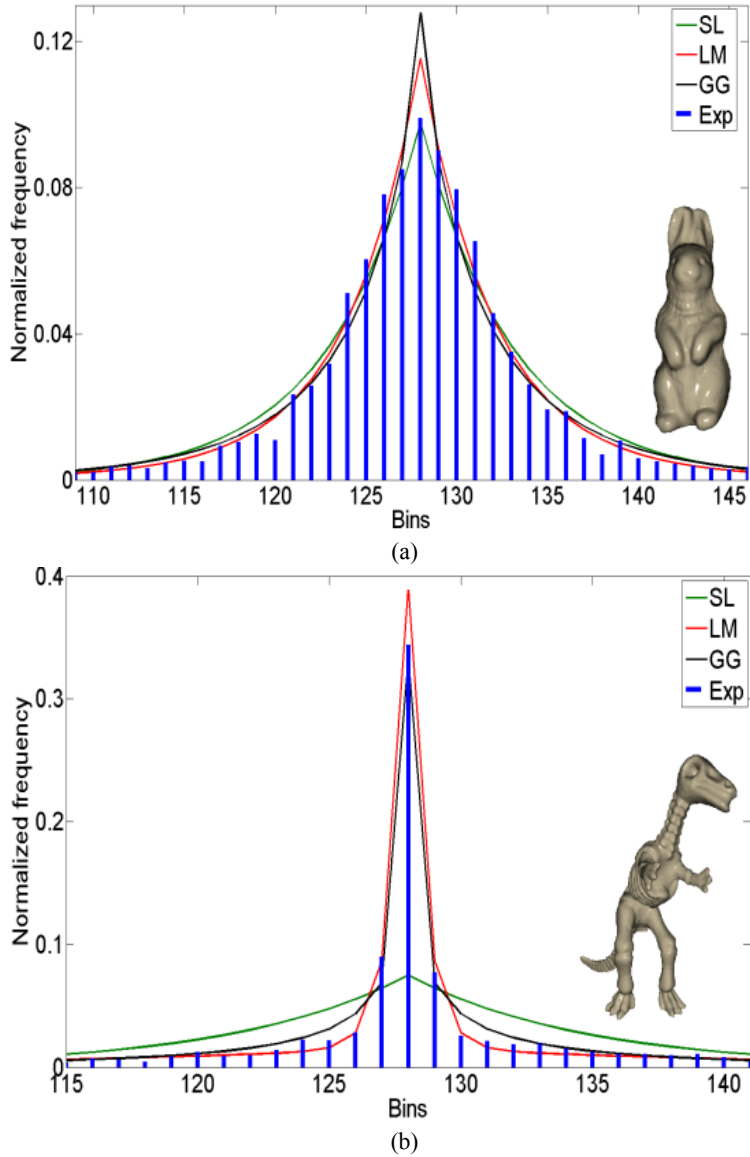
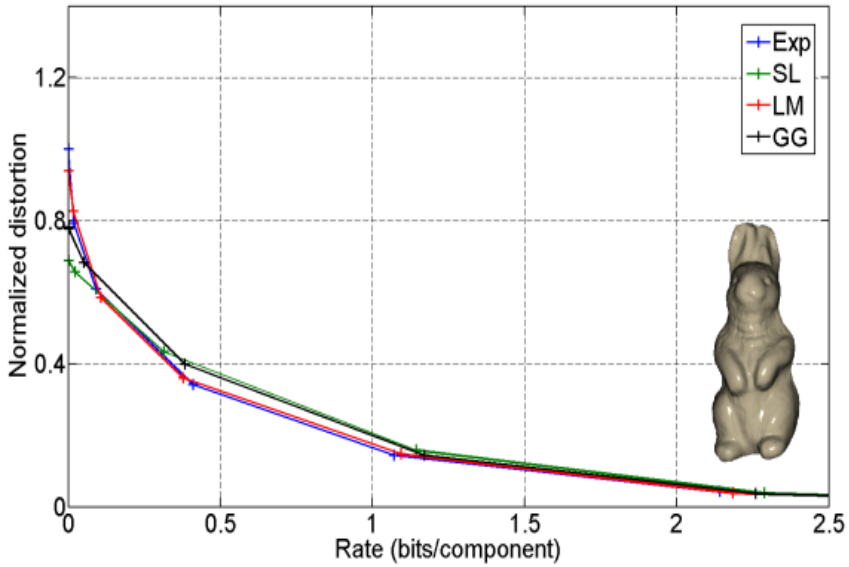
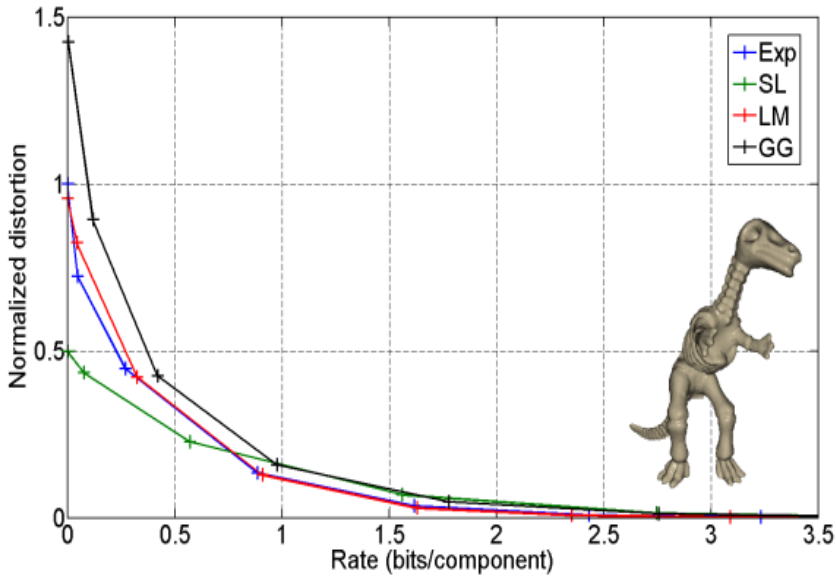


Figure 3.7: Probability function fitting over the observed histogram (Exp) for d^{J-3} -normal component for Rabbit (a) and Dino (b). SL is used as the abbreviation of single Laplacian PDF. The histogram is computed for 255 bins, the zero value correspond to the 128th bin.



(a)



(b)

Figure 3.8: Modeled and observed D-R functions for the histograms of Figure 3.7. Rate is taken as bits per spatial coordinate component.

Figure 3.8 plots the observed (Exp) and model D-R curves for meshes shown in Figure 3.7. For Rabbit, the LM's D-R almost completely overlaps the observed D-R curve. In both cases, the D-R function of the proposed LM model follows the experimental D-R curve more closely than the other two models.

In Table 3.1, the average KL divergence results for the Laplacian, GG and LM models for two non-normal (Venus, Rabbit) and three normal (Dino, Skull, Skrewdriver) meshes are tabulated. Each of the three coordinate components is separately considered. For each trial of Table 3.1, the average is taken over five highest resolution subbands. For the large majority of cases, the LM model gives better fitting of the observed histogram than the competing GG model. Note that the Laplacian model gives always the worst fitting results. Also, the LM model gives equally good fitting for both normal (NOR) and tangential (TAN 1 and TAN 2) components.

In Table 3.1, the percentage modeling error $ME(\%)$ relative to the KL divergence is shown in parenthesis of each table entry. The $ME(\%)$ is defined in order to gauge the D-R accuracy of the proposed mixture model with respect to other two models. $ME(\%)$ is defined as:

$$ME(\%) = \frac{\int_{R \in \mathfrak{R}} |D_M(R) - D_E(R)|}{\int_{R \in \mathfrak{R}} \max_R \{D_M(R), D_E(R)\}} \times 100, \quad (3.25)$$

where $D_E(R)$ and $D_M(R)$ denote the observed and the model D-R curves.

From Table 3.1, it is evident that on average the proposed LM model performs better than the GG and Laplacian models also in the ME sense. One notices that, a best histogram fitting in KL sense may not always yield the lowest ME. All in all, one can conclude that, compared to the contemporary models, the proposed LM model along with the derived D-R function is a better choice for modeling both the histogram and the D-R curve of mesh wavelet coefficients.

Table 3.1. KL (%ME, the modeling error as defined in equation (3.25)) for the *normal* (NOR) and the two *tangential* components (TAN1, TAN2) averaged over the five subbands. U-BF (Unlifted Butterfly), L-BF (Lifted Butterfly) [27].

| MESH TYPE | MESH (FILTER) | SL | | | LM | | | GG | | |
|------------|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | | NOR | TAN 1 | TAN 2 | NOR | TAN 1 | TAN 2 | NOR | TAN 1 | TAN 2 |
| NON-NORMAL | Venus(U-BF) | 0.097 (6.3) | 0.114 (10.3) | 0.103 (8.9) | 0.075 (1.3) | 0.100 (3.3) | 0.091 (2.8) | 0.086 (4.7) | 0.102 (5.0) | 0.089 (4.7) |
| | Venus(L-BF) | 0.137 (11.4) | 0.137 (10.1) | 0.104 (6.0) | 0.090 (2.0) | 0.108 (2.9) | 0.080 (1.9) | 0.112 (8.1) | 0.121 (5.4) | 0.092 (4.1) |
| | Venus(Loop) | 0.113 (9.4) | 0.102 (7.3) | 0.091 (6.7) | 0.085 (3.3) | 0.090 (1.8) | 0.069 (1.7) | 0.098 (7.1) | 0.092 (3.9) | 0.081 (5.0) |
| | Rabbit(U-BF) | 0.170 (8.6) | 0.171 (10.4) | 0.172 (10.2) | 0.134 (1.4) | 0.136 (2.0) | 0.132 (1.8) | 0.150 (5.7) | 0.143 (5.2) | 0.147 (6.2) |
| NORMAL | Rabbit(L-BF) | 0.208 (12.0) | 0.188 (10.7) | 0.177 (8.4) | 0.143 (2.5) | 0.140 (1.5) | 0.138 (1.8) | 0.160 (6.7) | 0.152 (5.1) | 0.153 (5.3) |
| | Rabbit(Loop) | 0.167 (11.2) | 0.207 (11.2) | 0.173 (8.3) | 0.115 (2.4) | 0.156 (1.8) | 0.135 (2.3) | 0.136 (7.9) | 0.177 (7.6) | 0.152 (5.2) |
| | Dino(U-BF) | 0.527 (16.2) | 0.656 (34.3) | 0.971 (42.7) | 0.145 (5.6) | 0.147 (7.8) | 0.154 (9.4) | 0.165 (7.5) | 0.132 (23.4) | 0.141 (30.3) |
| NORMAL | Skull(U-BF) | 1.108 (37.2) | 1.473 (44.9) | 1.877 (50.4) | 0.120 (3.9) | 0.138 (7.9) | 0.157 (20.5) | 0.145 (12.8) | 0.141 (15.4) | 0.141 (19.9) |
| | Skrewdriver(U-BF) | 0.529 (33.0) | 0.648 (42.4) | 0.638 (41.9) | 0.309 (14.4) | 0.251 (17.0) | 0.263 (20.0) | 0.315 (25.0) | 0.262 (34.8) | 0.249 (35.2) |

Table 3.2: KL (%ME) for three resolution subbands averaged over the three coordinate components.

| MESH TYPE | MESH (FILTER) | SL | | | LM | | | GG | | |
|------------|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| | | $J-1$ | $J-2$ | $J-3$ | $J-1$ | $J-2$ | $J-3$ | $J-1$ | $J-2$ | $J-3$ |
| NON-NORMAL | Venus(U-BF) | 0.044 (13.6) | 0.038 (8.4) | 0.039 (3.5) | 0.014 (2.9) | 0.008 (0.85) | 0.025 (1.5) | 0.027 (9.5) | 0.024 (5.6) | 0.033 (2.8) |
| | Venus(L-BF) | 0.050 (13.8) | 0.070 (13.9) | 0.076 (7.6) | 0.010 (2.1) | 0.010 (1.5) | 0.027 (1.7) | 0.029 (9.9) | 0.041 (10.1) | 0.049 (4.1) |
| | Venus(Loop) | 0.051 (13.8) | 0.054 (11.0) | 0.038 (4.0) | 0.016 (2.4) | 0.009 (1.6) | 0.023 (0.80) | 0.036 (11.0) | 0.031 (6.8) | 0.032 (2.7) |
| | Rabbit(U-BF) | 0.064 (14.0) | 0.062 (11.4) | 0.082 (8.1) | 0.008 (1.5) | 0.011 (1.0) | 0.035 (1.3) | 0.029 (8.6) | 0.032 (6.8) | 0.054 (4.9) |
| | Rabbit(L-BF) | 0.069 (14.2) | 0.093 (13.9) | 0.111 (11.2) | 0.007 (1.5) | 0.011 (1.1) | 0.035 (1.8) | 0.029 (8.6) | 0.040 (8.6) | 0.058 (5.9) |
| | Rabbit(Loop) | 0.082 (16.4) | 0.088 (14.7) | 0.085 (9.0) | 0.011 (2.0) | 0.013 (1.9) | 0.034 (2.0) | 0.042 (11.5) | 0.038 (8.7) | 0.058 (5.4) |
| | Dino(U-BF) | 1.208 (56.7) | 0.873 (46.7) | 0.623 (34.7) | 0.029 (13.7) | 0.074 (12.8) | 0.049 (4.6) | 0.031 (41.3) | 0.042 (32.9) | 0.058 (20.4) |
| | Skull(U-BF) | 2.039 (65.7) | 1.981 (49.9) | 1.832 (32.4) | 0.054 (34.2) | 0.040 (7.6) | 0.076 (5.1) | 0.036 (40.2) | 0.068 (15.1) | 0.066 (8.3) |
| | Skrewdriver(U-BF) | 0.536 (67.0) | 0.696 (61.6) | 0.483 (39.7) | 0.067 (53.2) | 0.074 (18.2) | 0.065 (7.3) | 0.035 (64.0) | 0.064 (54.3) | 0.101 (25.4) |

Table 3.2 reports the model validation results for different resolution subbands. For each trial the average is taken across the three spatial coordinate components. It is observed that the GG model performs slightly better for the low-resolution subbands of some meshes. The observed histograms in such cases are more Gaussian-alike, i.e., they have a round top. In general, the LM model faces difficulty in approximating such a round-top histogram due to the peaky nature of each of its Laplacian components; the GG fits well such histograms, as it corresponds to a Gaussian distribution for $\alpha = 2$. Nevertheless, the results show that, on average, the LM model outperforms the Laplacian and the GG models in KL as well as in ME sense.

3.3.5 Optimal Embedded Quantization

In this section, conclusions regarding the optimal EDSQ to be used in scalable wavelet-based coding of meshes are drawn. Let z denote the ratio between the dead-zone size and the cell size of a general EDSQ at $n = 0$. The total average SNR difference which is utilized to measure the performance gap of different embedded quantizers is defined as:

$$\overline{\Delta SNR} = \frac{1}{N} \sum_{\mathfrak{R}} \left(SNR(R)_{z=1} - SNR(R)_{z>1} \right),$$

which is computed over a rate range \mathfrak{R} for N rate points, where $SNR(R)$ denotes the discrete SNR-rate function. The $SNR = 10 \log_{10}(\sigma^2/D)$ is computed in dB, where D is the total distortion in the transform domain. The difference in SNR is computed relative to the uniform embedded quantizer (UEQ), i.e., $z = 1$. $\overline{\Delta SNR}$ for five embedded dead-zone quantizers is plotted in Figure 3.9 over a wide range of standard deviation ratios σ_2/σ_1 . In Figure 3.9, the commonly observed proportion $\beta = 0.9$ is considered.

We experimentally determined that at lower standard deviation ratios, $\overline{\Delta SNR}$ is positive and the UEQ is optimal for $\sigma_2/\sigma_1 < 120$. For $120 < \sigma_2/\sigma_1 < 290$, the quantizer with $z = 1.5$ performs better compared to all other quantizers. Similarly, $z = 2$ (i.e. the SAQ) performs the best in the range $290 < \sigma_2/\sigma_1 < 600$, while $z = 2.5$ performs the best for $600 < \sigma_2/\sigma_1$. In general, small standard deviation ratios correspond to α close to 1, observed in non-normal meshes, while higher ratios correspond to $\alpha \ll 1$, observed in normal meshes. These results show that one cannot determine a single embedded quantizer that provides the best performance for *all* 3-D meshes. However, an optimal quantizer per wavelet coordinate can be determined based on the corresponding σ_2/σ_1 extracted from the model. Overall, for $\sigma_2/\sigma_1 < 120$, the difference between SAQ and the UEQ is significant, and hence UEQ is the optimal choice. For $\sigma_2/\sigma_1 \geq 120$, which is the

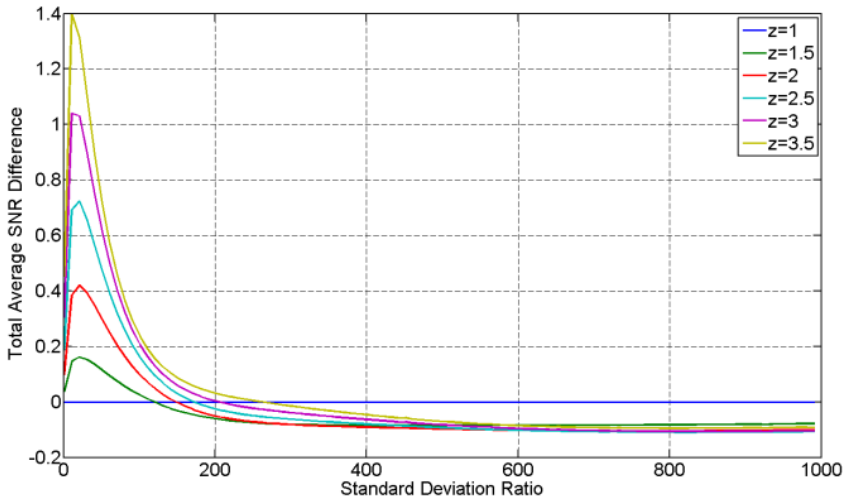


Figure 3.9: SNR difference for five EDSQs with respect to UEQ.

practical range of σ_2/σ_1 values for real meshes, SAQ is not always the optimum but lies not far from the optimum.

Given the fact that SAQ is closely linked to bit-plane coding and that it can be implemented using simple binary arithmetic, one concludes that SAQ is not an optimal, but an acceptable solution in scalable coding of meshes.

3.4 ANALYSIS OF WAVELET COEFFICIENT DEPENDENCIES

In Figure 3.10(middle, right), the positions of the wavelet coefficients at different levels of the subdivision transform are shown with the help of white and dark circles. In particular, wavelet coefficients have a one-to-one correspondence with the edges of the coarser mesh. In the decomposed mesh structure, for each wavelet coefficient, there are rings of neighboring coefficients that lie in the same wavelet subband – see Figure 3.10(right). Also, a set of four wavelet coefficients have a parent coefficient at the next coarser resolution – see Figure 3.10(middle, right).

Similar to images, neighboring (intra-band) and parent-descendant (inter-band) wavelet coefficient dependencies also exist in the wavelet decomposed structure of a mesh also. We point out that the strength of these dependencies in meshes may be completely different than in images. Particularly, in images, which consist of pixels (with intensities), intra-band dependencies exist due to the smooth variation of pixel intensity over the spatial regions while the inter-band dependencies occur due to the natural decay of wavelet coefficients towards higher frequencies. However, 3-D meshes consist of vertices in the 3-D space and the wavelet coefficients are basically the displacement vectors. Due to different statistical properties of images and

meshes, one cannot decide about the relative strength of wavelet coefficient dependencies for meshes from the results available for images [62], [63]. Thus, a separate analysis of these dependencies for a mesh wavelet transform needs to be conducted.

In the following, an information theoretical analysis of the aforementioned coefficient dependencies is presented. Our aim is to single out the type of dependencies which can ensure best compression performance in the context of wavelet-based mesh compression.

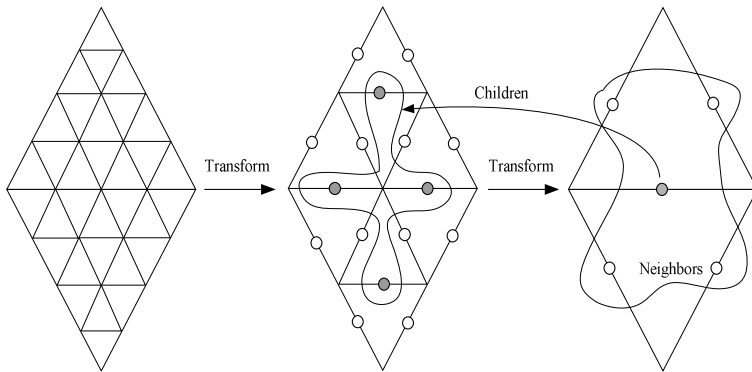


Figure 3.10: Parent-children and neighboring wavelet coefficients: actual mesh (left); coarser meshes after one (middle), and after two wavelet decomposition levels (right).

3.4.1 Statistical Information Analysis

As explained in Section 2.2.5, the mutual information is the reduction in the entropy of one random variable due to the knowledge of another random variable that statistically depends on the former [21], i.e.,

$$I(X;Y) = H(X) - H(X/Y) , \quad (3.26)$$

To perform the mutual information analysis, we define the following mutual information quantities in the wavelet domain:

1. $I(X;P_X)$ denotes the mutual information between a wavelet coefficient X and its parent coefficient P_X .
2. $I(X;n_X)$ denotes the mutual information between a wavelet coefficient X and its neighboring wavelet coefficients $n_X = [n_{1,X}, n_{2,X}, \dots, n_{N,X}]$.
3. $I(X;P_X, n_X)$ denotes the composite mutual information.

In the above quantities, the realizations of different random variables are the magnitudes⁷ of the corresponding wavelet coefficient(s). From the basics of information theory [21], we know that:

⁷ The magnitude of the vector valued wavelet coefficient $v = (v_x, v_y, v_z)$ is taken as $\|v\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$.

$$I(X; P_X, \mathbf{n}_X) \geq I(X; \mathbf{n}_X) \quad \text{and} \quad I(X; P_X, \mathbf{n}_X) \geq I(X; P_X). \quad (3.27)$$

For the estimation of $I(X; \mathbf{n}_X)$, we need to estimate the joint PDF $p(x, \mathbf{n}_x)$ which can have high dimensionality depending on the number of considered neighbors. Since the amount of data needed to accurately estimate a PDF increases exponentially with its dimensionality, it is difficult to reliably estimate a high-dimensional PMF. To alleviate this problem, dimensionality reduction method of [62], [63], is reused here. We summarize the neighborhood of X through a so-called summarizing function $T = g(\mathbf{n}_X)$. This function maps the neighboring wavelet coefficients to a single value. We note that such a many-to-one summarizing function cannot increase the mutual information [21], i.e.,

$$I(X; \mathbf{n}_X) \geq I(X; T). \quad (3.28)$$

Equality in the above equation holds if $\mathbf{n}_X \rightarrow T \rightarrow X$ forms a Markov chain, in which case, T provides the *sufficient statistics* [21] for the neighborhood vector random variable \mathbf{n}_X . The summarizing function used in our analysis is:

$$T = g(\mathbf{n}_X) = \sum_{i=1}^N (n_{i,X})^2. \quad (3.29)$$

Due to this summarizing function, it is sufficient to compute the joint PDF $p(x, t)$, where t is a realization of T .

In our analysis, the mutual information for the defined quantities is estimated using the adaptive partitioning method [22] instead of the traditional histogram method. This is because the histogram method highly depends on the bin size and for a small bin size there may not be sufficient number of observations in some bins to make a correct estimate. The adaptive partitioning method [22] on the other hand, ensures that there are always sufficient numbers of observations in each bin, and provides reliable estimates of the mutual information.

Table 3.3 shows the average mutual information results for interband, intraband and composite dependencies for various mesh models. Since in mesh coding three different components need to be coded for each vertex position in space, the average mutual information $I_{avg} = (I_X + I_Y + I_Z)/3$ is reported instead of the mutual information for the three components individually. It is observed from Table 3.3 that, for both normal and non-normal meshes, the mutual information of interband models is the least and is independent of the employed wavelet transform. On the other hand, mutual information for intraband models is significantly higher than for the interband models. Finally, the composite models, which gather the characteristics of both interband and intraband models, exhibit even higher mutual information than interband or intraband models alone. Mathematically we can summarize our numerical findings as:

Table 3.3: Average mutual information (in bits) for several non-normal and normal meshes. For intraband and composite cases, the first-ring neighboring wavelet coefficients are considered. Additionally, in the composite case, the corresponding parent wavelet coefficient is also considered. As the normal meshes are designed for the un-lifted Butterfly transform [52], for normal meshes, no results are computed for the Loop transform case.

| MESH TYPE | MESH | BUTTERFLY | | | LOOP | | |
|------------|--------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | Intraband | Interband | Composite | Intraband | Interband | Composite |
| NON-NORMAL | Venus | 0.3727 | 0.1902 | 0.6886 | 0.8320 | 0.5591 | 1.5847 |
| | Bunny | 0.3960 | 0.1992 | 0.6844 | 0.8033 | 0.5628 | 1.5427 |
| | Horse | 0.5615 | 0.2869 | 0.9873 | 1.0482 | 0.6943 | 1.9684 |
| | Rabbit | 0.4048 | 0.2017 | 0.7089 | 0.8996 | 0.6450 | 1.7425 |
| | Feline | 0.8277 | 0.2134 | 1.0696 | 1.1471 | 0.6285 | 2.0287 |
| NORMAL | Venus | 0.3052 | 0.2130 | 0.5741 | - | - | - |
| | Skull | 0.3381 | 0.2922 | 0.7001 | - | - | - |
| | Dino | 0.3043 | 0.2804 | 0.6672 | - | - | - |

$$I(X; P_X) \ll I(X; n_X) \ll I(X; P_X, n_X). \quad (3.30)$$

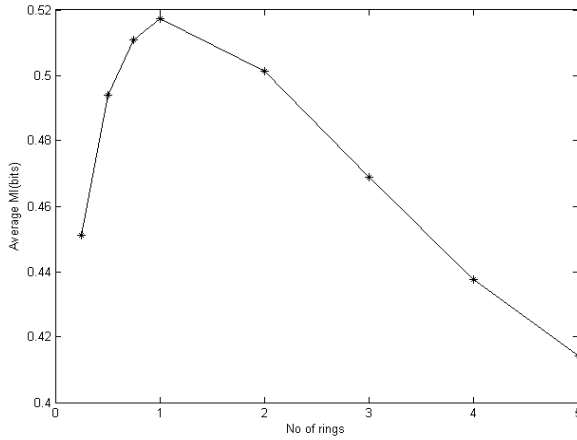
Experimental results for the mutual information based estimation of interband, intraband and composite dependencies seem to indicate that exploiting the composite dependencies should be preferred over the other two. Additionally, it is important to point out that favoring intraband over zero-tree based interband models brings along the additional benefit of resolution scalability. Specifically, by following an intraband codec design, only those wavelet subbands that are needed in order to reconstruct a target mesh resolution-level need to be encoded, while the others can be discarded. This does not hold in case of interband and composite codec designs, due to the tree-structures that span all the wavelet decomposition levels. Since composite models cannot be discarded altogether due to their highest mutual information property, a careful implementation of a composite mesh coding system needs to be carried out in order to simultaneously get the benefit of both higher compression efficiency and resolution scalable decoding.

In Figure 3.11, the intraband mutual information is plotted for a varying number of neighboring wavelet coefficients n_X . Due to the random positions of the vertices in space, neighboring coefficients are extracted as rings. The figure plots the mutual information up to the first 5 rings. In the 1st ring, there are always 4 neighboring wavelet coefficients, as shown in Figure 3.10. The most-left points in Figure 3.11 (a)-(b) correspond to the estimated mutual information taking the neighboring wavelet coefficient in the 1st ring that has the highest magnitude. The 2nd point is the estimated mutual information by taking the two wavelet coefficients with the highest magnitudes in the 1st ring, a.s.o.

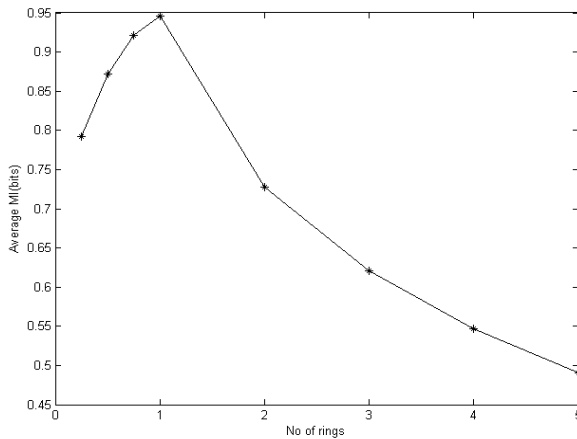
According to information theory, two or more random variables, say $\beta_1, \beta_2 \dots \beta_n$, convey equal or more information about a given random variable γ than a subset $\beta_1, \beta_2 \dots \beta_{n-1}$; in other words, the mutual information must satisfy the so-called chain rule [21]:

$$I(\alpha; \beta_1, \beta_2 \dots \beta_n) \geq I(\alpha; \beta_1, \beta_2 \dots \beta_{n-1}). \quad (3.31)$$

The plots in Figure 3.11 are contradictory to the above relation, as the mutual information decreases if more and more neighboring wavelet coefficients are considered when estimating the intraband mutual information $I(X; n_X)$. Such a behavior is due to the inappropriateness of the summarizing function T for large number of neighbors. The summarizing function T can only be used for reducing the dimensionality of the joint PDF $p(x, \mathbf{n}_x)$ if and only if, given T , the wavelet coefficient X is independent of its neighborhood. In Figure 3.11, the decay in the mutual information for two and more rings indicates that T is not an appropriate



(a)



(b)

Figure 3.11: Average intraband mutual information (MI) for varying number of neighboring wavelet coefficients, averaged over 5 non-normal meshes. (a) Butterfly (b) Loop.

summarizing function anymore for more than four neighboring wavelet coefficients (i.e., 1st ring). We notice that the average intraband mutual information results in Table 3.3 are provided by taking all the 1st ring coefficients, for which T is certainly an appropriate summarizing function.

Finally, it is important to point out that the differences in terms of mutual information do not give any indication about the final performance differences between interband, intraband and composite coding systems. Hence, an actual development and comparison of such coding systems is needed in order to experimentally validate the conclusions of this information-theoretic analysis of wavelet-based mesh coding designs, which is presented next.

3.5 SCALABLE MESH COMPRESSION: OVERVIEW AND DESIGN

In this section, we give a brief overview of the scalable mesh compression systems that existed before our study. Moreover, we describe the design of novel scalable intraband and composite mesh coding systems, following our information-theoretic analysis of the statistical dependencies between the coefficients. In our proposed codecs, the wavelet subbands are independently encoded using an extension of quadtree method explained in Section 2.5.5.2. Furthermore, context-based entropy coding employing either intraband or composite models is applied. The proposed codecs provide both resolution and quality scalability. This lies in contrast to the state-of-the-art interband zero-tree based semi-regular mesh coding techniques, which supports only quality scalability. Additionally, the experimental results show that, on average, the proposed codecs outperform the interband state-of-the-art for both normal and non-normal meshes. Moreover, compared to a zero-tree coding system, the proposed coding schemes are better suited for software/hardware parallelism, due to the independent processing of wavelet subbands.

3.5.1 Progressive Geometry Compression (PGC)

Before our study, the state-of-the-art in scalable wavelet-based compression of semi-regular meshes was the progressive geometry compression (PGC) codec proposed by Khodakovsky et al. in [57]. PGC system was originally proposed to progressively compress highly-detailed, densely sampled, arbitrary topology meshes. The main insight given by the work of [57] was the separation of the vertex's position information into two partially independent types of information, namely *parameter* and *geometry*. Prior to the work of [57], such a separation was not considered during the compression of arbitrary topology meshes. The differentiation between the parameter and geometry information can be explained using the following example: Imagine a vertex in a certain 3-D mesh (say Venus, see Figure 3.2(a)). Slightly moving this vertex within the surface of the mesh does not affect the shape of the object, however, moving this vertex normal to the surface will certainly do. The parameter information describes vertex displacement in the plane tangential to the mesh surface, while geometry describes its displacement in the plane normal to the surface. From the rate-distortion point of view, such a differentiation has a particular significance as it allows to assign different amount of bits to different types of information depending on their importance. In particular, relative to the tangential, the normal information should be given more rate, and hence coded with less error. This is

because, towards lowering the overall distortion of the mesh, the geometry information contributes more than the parameter information.

Based on the unique differentiation of the parameter and the geometry information, Khodakovsky et al. proposed the PGC scalable mesh codec [57]. At first the wavelet-decomposition of the input semi-regular mesh is carried out in order to transform the mesh to subbands of wavelet coefficients and a coarsest level mesh M^0 referred to as the *base mesh*. Each vector-valued wavelet coefficient is then represented in the local coordinate frame [128] to separate its geometric and parametric components. The normal component, which correspond to the geometry information, is finely quantized while the two tangential components, which correspond to the parameter information, are coarsely quantized using the SAQ. PGC makes use of the well-known SPIHT coder [87], which is based on embedded zero-tree bit-plane coding [100], in order to encode bit-planes of wavelet coefficients. The normal and two tangential components are successively and independently bit-plane coded. The produced bit-plane coded bit-stream is further losslessly coded using adaptive arithmetic coding. To compress the base mesh, the PGC compression coder uses the state-of-the-art single rate Touma and Gotsman (TG) lossless codec of [112].

Significant improvements in the compression performance against the contemporary scalable [82] as well as non-scalable mesh coding systems [106, 112] were reported in [57]. Unlike PGC, which operates on semi-regular meshes, contemporary schemes were made to run on irregular mesh directly. As the PGC can at best reconstruct the semi-regular mesh losslessly, a quality saturations in its rate-distortion curve occurs when approaching close to the remeshing error [57]. With respect to the scalable codec of [82], at the same rate, PGC codec was shown to provide approximately 12dB improvement in the reconstructed mesh quality. When compared against fixed-rate coders [106, 112], even higher performance gains are observed for PGC. However, beyond the remeshing error, the fixed-rate codecs outperform the PGC scheme. This is completely understandable as, with respect to the original irregular mesh, PGC cannot reduce the error in the decoded semi-regular mesh below the remeshing error. For normal meshes [52], the extension of PGC codec was published in [58]. Normal meshes are built using a special type of transform-specific remeshing which ensures that wavelet coefficients have minimum (almost no) energy in the tangential components. In other words, for normal meshes, the total energy lies in the normal geometry component. Thus, effectively, each wavelet coefficient can be encoded using a single floating point number, instead of three floating point numbers for non-normal meshes. Note that, unlike non-normal meshes [57], no local frame representation [57] is required for normal meshes as it is already build in the remeshing process [52]. It was shown in [58] that compressing normal

(instead of non-normal) meshes can bring marginal compression gains for some meshes, while significantly high gains (2 to 5 dB) for others.

A major drawback of PGC schemes is their inability to provide resolution scalability. This is caused by the zero-tree structure which, for a given bit-plane, spans all the wavelet decomposition levels. The only form of scalability PGC enables is the quality scalability which results from the joint progressive coding of bit-planes belonging to different subbands. However, we point out that despite of its inability to provide resolution scalability, PGC is a very efficient mesh compression framework which could bring enormously high compression gains with respect to the contemporary schemes – see [57], [58].

3.5.2 Scalable Intra-band Mesh (SIM) Compression

Despite the great success of zero-tree based coding techniques in image coding, the choice of an interband codec design is not necessarily the best option in the context of scalable mesh coding. This was illustrated in Section 3.4.1, where different types of dependencies among wavelet coefficients were studied. Based on this analysis, we opt for an intra-band dependency model in our codec design. As mentioned before, favoring intra-band models over interband models brings along the additional benefit of resolution scalability. Specifically, by following an intra-band codec design, only those wavelet subbands that are needed in order to reconstruct a target mesh resolution-level need to be encoded, while the others can be discarded. Figure 3.12 depicts the block diagram of the proposed SIM encoder. Note that the coding pipeline is quite similar to the one for the wavelet image coding, given in Figure 2.20. Namely, the input semi-regular mesh is wavelet decomposed and the produced J wavelet coefficients are bit-plane coded by employing SAQ. The produced bit-plane coded symbols are further losslessly entropy coded using binary arithmetic coding. The base mesh is coded using a fixed-rate coder, e.g., the TG codec of [112].

As mentioned before, in our mesh coding systems, the generated coefficient subbands are coded using an extension of quadtree coding, presented in Section 2.5.5.2. Note that, in contrast to images, wavelet coefficients for meshes are irregularly spaced and lie in a 3-D space. Given this property of mesh wavelet coefficients, it is not immediately clear how a 2-D quadtree coding method, which was originally designed for images, can be adapted to encode coefficients generated by a 3-D mesh wavelet transform.

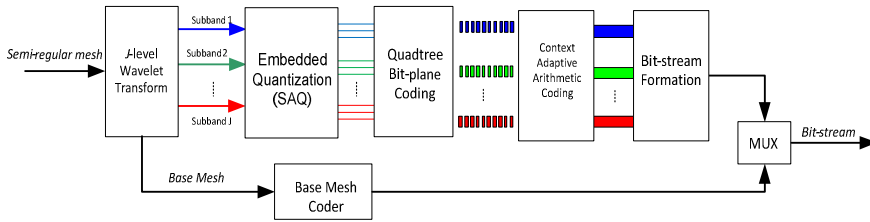


Figure 3.12: The block diagram of the proposed SIM encoder. The decoder follows the block operations in the reverse direction.

The adaptation of quadtree coding algorithm to irregularly spaced wavelet coefficients, which is my main contribution towards the design of a novel intraband mesh codec, is described next.

In the designed scalable intraband mesh (SIM) compression system [27, 28] each resolution subband is encoded independently of the others. Similar to [57], SAQ is applied to each resolution subband to determine the significance of the wavelet coefficients with respect to a series of monotonically decreasing thresholds. The *bounding box* of the 3-D mesh is considered as the root-node of the tree and based on the significance outcome, a tree node is split into eight equal volume nodes. Figure 3.13 shows the example of octree decomposition of a single mesh wavelet subband. Other than the difference of quadtree and octree, the bit-plane coding algorithm is exactly the same as the QT-L coding proposed in [94].

Note that, unlike quadtree decomposition in images, octree decomposition in meshes produces nodes that contain an unequal number of wavelet coefficients. This causes a major difficulty in the bit-plane decoding process. In images, due to their regular structure, the number of wavelet coefficients contained in all tree nodes of a given depth are fixed and can be perfectly determined at the decoder side. This is not the case for meshes as the number of wavelet coefficients inside the tree nodes of the same tree-depth are generally not equal. Having received a NSG node symbol, the decoder does not know how many and which wavelet coefficients are actually non-significant. To resolve this ambiguity, the octree decomposition at the encoder side is done based on the *template* mesh rather than based on the original mesh. Any mesh which has the same number of vertices as the original mesh and which can be automatically constructed at the decoder side can be used as the template mesh. Ideally, the template mesh should have the same geometry as the original semi-regular mesh, which is of course not feasible, since this would imply that all vertices of the input mesh are pre-known at the decoder side. In our method, we utilized the J -level subdivided base mesh as the template mesh. Note that the same template mesh can be constructed both at the encoder and the decoder sides. Using the template mesh, an octree is independently constructed for each resolution subband

and magnitude bit-planes are sequentially coded using the non-significance, the significance and the refinement coding passes, as in [94]. Similar to [57], [58], all three coordinate components of the wavelet coefficients are bit-plane coded independently.

The bit-stream produced by the bit-plane coding module is then entropy coded using adaptive arithmetic entropy coding. In the SIM codec, similar to QT-L based image codec of [94], we utilize the more efficient context-based adaptive arithmetic coding. The significance, the non-significance, and the sign information is entropy coded using context-based arithmetic coding [27]. In this regard, only the first-ring (see Figure 3.10-right for the definition of the first-ring) neighboring wavelet coefficients are considered for the context modeling. In total six contexts are used to encode the significance of a given wavelet coefficient [27]. The contexts used to encode the refinement symbols were initially adopted from JPEG-2000. However, due to the uniform distribution of the refinement symbols, no satisfactory compression improvements were achieved. Therefore, to limit the complexity, no context-conditioning is employed when entropy coding the refinement symbols. Empirical observations showed that the tangential and normal components in the same context exhibit different statistical characteristics. We therefore assign two different sets of context models to entropy code the normal and tangential components [27].

Using the codec structure described above, separate symbol streams are first generated for all bit-planes of each resolution subband. Depending on the type of scalability, i.e., resolution or quality scalability, the encoded symbol streams are entropy coded using a predefined progression order of bit-planes. For quality scalability, bit-planes of certain significance, from all resolution subbands, are first encoded before encoding the bit-planes of lower significance. A template for the generated quality scalable bit-stream is given by:

$$\overbrace{(B_{0,p}, B_{1,p}, \dots, B_{J-1,p})}^{\text{Quality level } P \text{ (lowest)}}, \overbrace{(B_{0,p-1}, B_{1,p-1}, \dots, B_{J-1,p-1})}^{\text{Quality level } P-1}, \dots, \overbrace{(B_{0,0}, B_{1,0}, \dots, B_{J-1,0})}^{\text{Quality level } 0 \text{ (highest)}} \quad (3.32)$$

where $B_{j,p}$ denotes the bit-stream corresponding to the p th bit-plane, $0 \leq p \leq P$, of the j th, $0 \leq j < J$, resolution subband. In resolution scalability mode, bit-planes of a lower resolution subband are progressively encoded before encoding the next higher resolution subband. The progression order template for this mode is given by:

$$\overbrace{(B_{0,p}, B_{0,p-1}, \dots, B_{0,0})}^{\text{Subband } 0}, \overbrace{(B_{1,p}, B_{1,p-1}, \dots, B_{1,0})}^{\text{Subband } 1}, \dots, \overbrace{(B_{J-1,p}, B_{J-1,p-1}, \dots, B_{J-1,0})}^{\text{Subband } J-1} \quad (3.33)$$

Empirical findings showed that, a resolution-scalable bit-stream constructed based on equation (3.33) is not competitive against its quality scalable counterpart, constructed by equation (3.32) in the rate-distortion sense [27]. In the following,

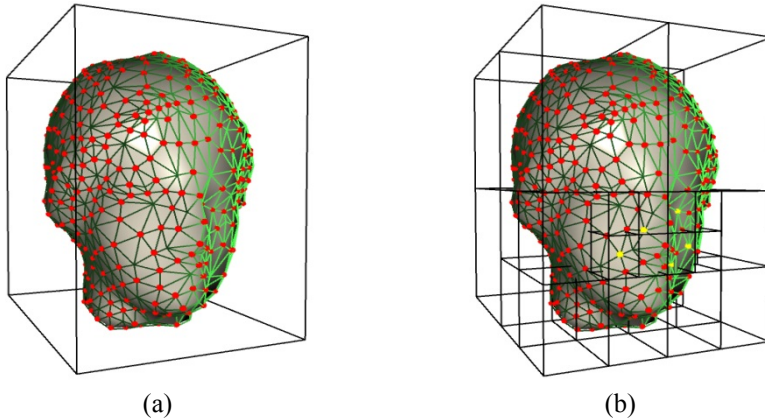


Figure 3.13: Bit-plane coding of a wavelet decomposed mesh subband. (a) The bounding box of a mesh (b) Octree decomposition of wavelet coefficients.

the compression performance of the SIM codec is evaluated in the quality scalable mode.

We compared the SIM codec with the PGC codec for both normal and non-normal 3-D meshes. The decoded meshes are compared against the original semi-regular input meshes using the peak signal-to-noise ratio (PSNR) as the distortion metric, which is defined as:

$$PSNR = 20 \log_{10} \left(\frac{peak}{RMS} \right) (\text{dB}), \quad (3.34)$$

where, *peak* and *RMS* denote the size of the *space diagonal* of the bounding box and the root mean squared error calculated on the distances between the decoded vertex positions with respect to the original ones, respectively. Figure 3.14 depicts PSNR versus bitrate (bits per semi-regular vertex, i.e., *bpv*) plots, evaluated for the semi-regular non-normal Venus and Bunny meshes, using the Butterfly transform. The results demonstrate that for both meshes, SIM codec yields superior performance when compared to PGC. The average gain in PSNR when compressing the Venus and Bunny meshes goes up to 2.22 dB and 2.35 dB, respectively. One may also notice the increasing performance difference with increasing bitrates; this indicates that the SIM coder tends to code the high frequency information more efficiently.

Figure 3.15 shows compression performance plots for two normal meshes, namely, Skull and Dino. One notices that at low bitrates, PGC tends to compress better. However, the ability of SIM codec to capture and code more efficiently the high-frequency components is noticeable at high bitrates and leads to an improved performance with respect to the PGC system.

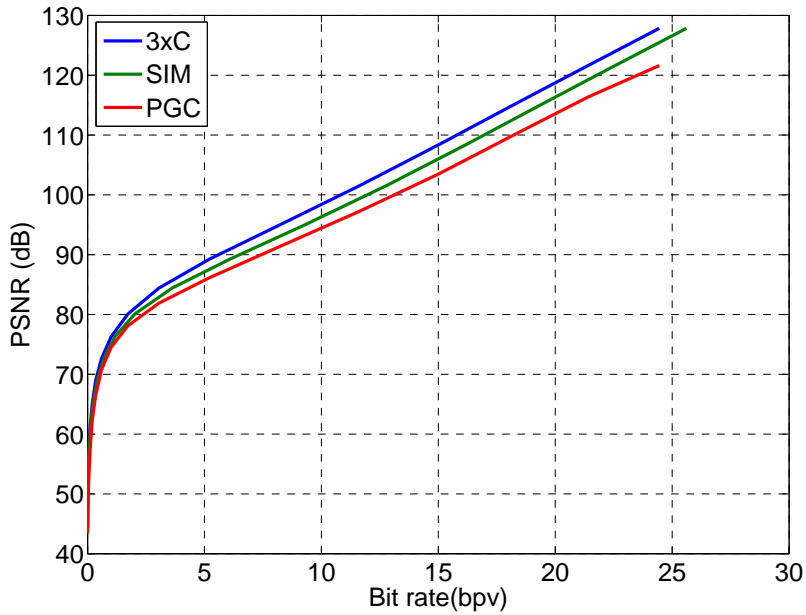
3.5.3 Composite Context-conditioned (3xC) Compression

The mutual information analysis presented earlier showed that the composite dependencies between the wavelet coefficients are by far the strongest. Hence, employing composite models during context-conditioning is expected to lead to an overall improvement of the compression performance. However, one may notice that, similar to interband models, employing composite models may hinder the possibility of providing resolution scalability. Additionally, exploiting composite dependencies may lead to an increased complexity of the mesh coding system. Thus, one must be careful in exploiting the parent-children dependencies within composite models. In this section, we detail the design of proposed novel composite context-conditioned mesh codec, which was originally named: composite context-conditioned (3xC) compression [25].

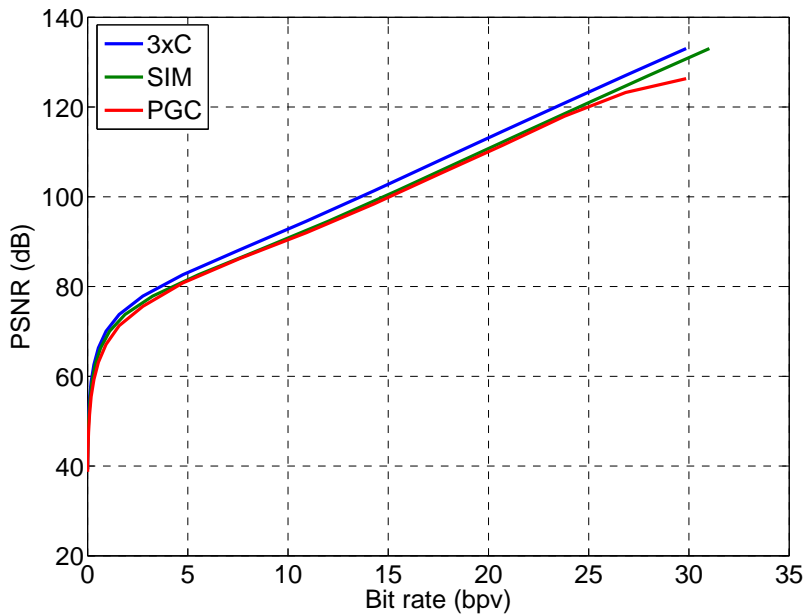
Empirical results showed that the statistical characteristics of neighboring intraband wavelet coefficients are partially characterized by the interband relation between that coefficient and its parent coefficient. Moreover, exploiting parent-children dependencies in a causal fashion does not limit resolution scalable decoding of the compressed mesh. In order to benefit from this observations, we proposed 3xC mesh compression system in [25, 27]. The proposed composite codec uses three identical sets of context-models to entropy code the significance of a given wavelet coefficient. These contexts are identical to the ones employed in the SIM codec, presented in Section 3.5.2. However, in a first conditioning phase, a specific set is chosen based on the significance of the parent coefficients. Except this difference, the bit-plane coding modules of the SIM codec (see Figure 3.12) and the 3xC codec are identical. For a detailed presentation of the 3xC codec the interested reader is referred to [25, 27].

The proposed 3xC codec, enhances over the SIM codec in terms of the compression performance [25, 27], as promised by our information theoretic analysis of Section 3.4.1. Moreover, like SIM codec, it also retains the capability of resolution scalable decoding of the compressed mesh due to exploiting causal cross-band coefficient dependencies. Figure 3.14 also depicts the PSNR curves computed for the non-normal Venus and Bunny meshes using our implementation of the unlifted Butterfly based 3xC mesh compression system. The curves clearly demonstrate that, when dealing with non-normal meshes, 3xC systematically yields superior performance compared to PGC as well as SIM.

In the case of normal meshes (Figure 3.15) our coder employs the same transform as PGC. Both PGC and 3xC perform the same at very low bitrates. However, overall, 3xC yields significantly better compression performance than PGC. 3xC

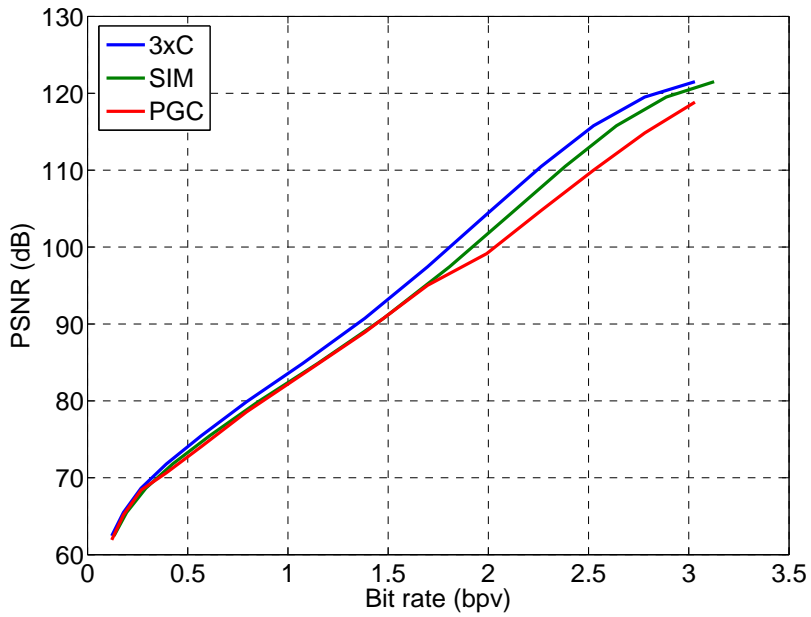


(a)

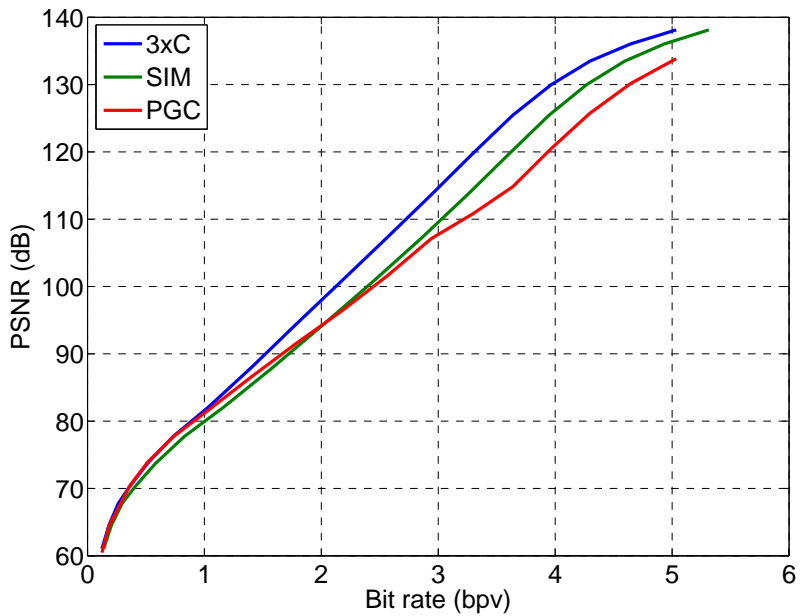


(b)

Figure 3.14: PSNR versus bitrate for non-normal mesh models in the quality scalability mode: (a) Venus, (b) Bunny . The lifted Butterfly transform is employed for all three codecs.

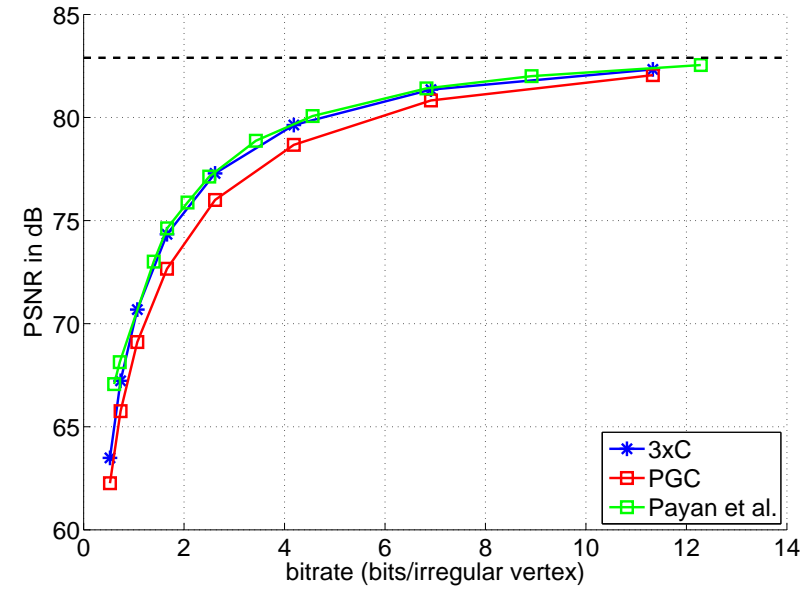


(a)

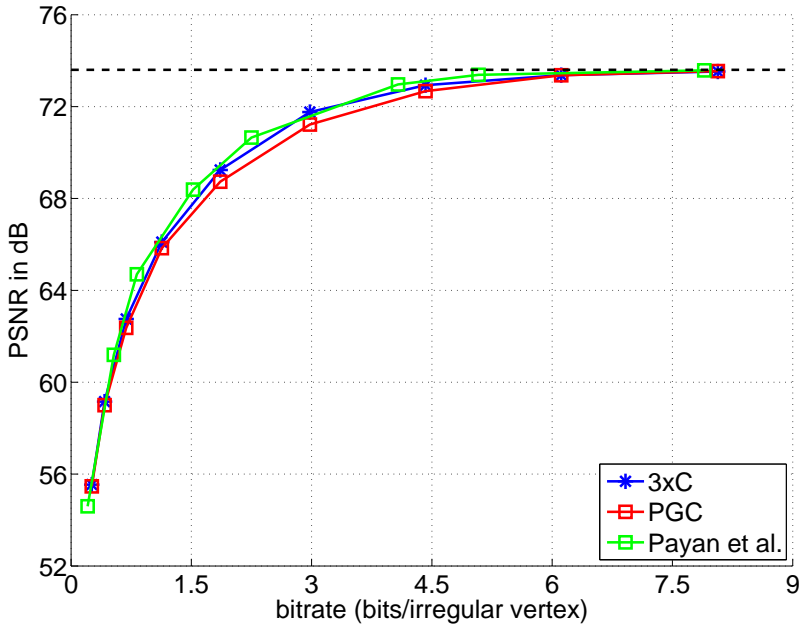


(b)

Figure 3.15: PSNR versus bitrate for normal mesh models in the quality scalability mode: (a) Skull, (b) Dino. The un-lifted Butterfly transform is employed for all three codecs.



(a) Venus



(b) Skull

Figure 3.16: Performance comparison between the proposed codec 3xC, scalable PGC codes [57], [58], and the state-of-the-art fixed-rate codec of [83] (Payan et al.). The compressed meshes are compared against the original irregular mesh using MESH tool of [8]. The rates include the compressed base mesh.

also gives better results when compared with the intraband SIM codec for normal meshes. This is because 3xC, unlike SIM, employs the parent context-conditioning for the normal component of the vector valued wavelet coefficients. Overall, it is clear that the proposed 3xC codec provides similar or superior performance compared to PGC and SIM codecs.

In addition, we compared 3xC against the state-of-the-art fixed-rate codec of [83]. For this set of experiments, we employed the MESH [8] tool for calculating the Hausdorff error between the reconstructed and the irregular mesh. The error computed by the MESH tool is substituted at the place of *RMS* in equation (3.34) to compute the Hausdorff PSNR. In Figure 3.16, the proposed 3xC codec is compared against the PGC codecs [57], [58], and the fixed-rate codec of [83] for the non-normal Venus and the normal Skull meshes. Since the error is calculated directly with respect to the irregular mesh the bitrate is also reported in bits per vertex of the irregular mesh. From Figure 3.16, it is clear that our codec outperforms PGC in this case as well. Moreover, we notice that 3xC, which is scalable, and the codec of [83], which is not scalable, yield similar performance. This is in spite of the fact that 3xC uses SAQ (which is not optimal), while [83] uses fixed-rate quantizers optimized for each wavelet subband.

3.5.4 Visual Comparisons

Visual comparisons of Bunny (non-normal) and Skull (normal) meshes, compressed and reconstructed using 3xC at different bpv, are presented in Figure 3.17 and Figure 3.18, respectively. The red colored regions highlight the distortions introduced due to lossy compression. For bitrates below 0.314 bpv for Bunny and below 1.073 bpv for Skull, the pure red color indicates areas where the distance between the original and decoded vertex is larger than 0.1% of the space diagonal⁸ of the bounding box. For bitrates corresponding to the last two rows of Figure 3.17 and Figure 3.18, the distortion is visualized with respect to 0.02% of the length of the space diagonal. The mesh is shaded greener as the distortion lowers, with pure green indicating no distortion.

When visually comparing the compressed Bunny and Skull meshes produced by 3xC and PGC, it is very clear that 3xC yields superior performance for all bitrates. For the Bunny mesh, taking the result at 0.050 bpv as an example, we observe that many areas that are shaded red for PGC, are green for 3xC. At high rates, the differences between mesh geometries may not be visually significant, yet the colors reveal that 3xC is able to approximate the original mesh much more accurately when

⁸ The space diagonal of a rectangular box is a line that goes from a corner of the box through its center to the opposite corner.

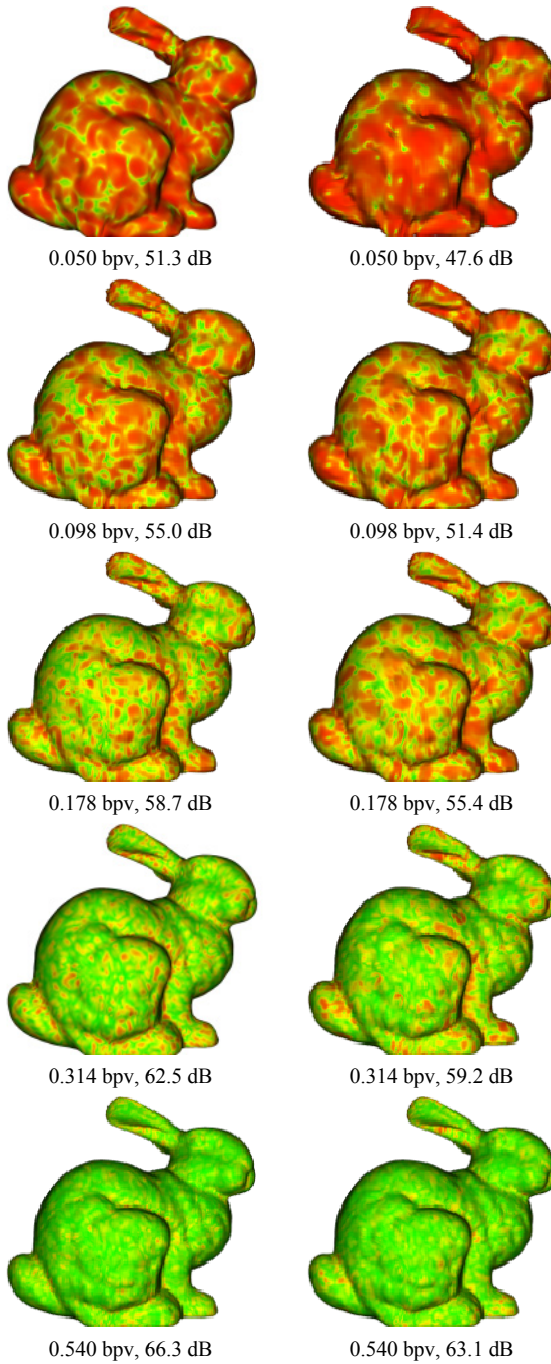


Figure 3.17: Visual comparison of non-normal Bunny mesh using (left column) the 3xC codec and (right column) the PGC codec. The red color intensity reflects the distortion with respect to the uncompressed semi-regular mesh. The rate for the base mesh is not included in the reported rate values.

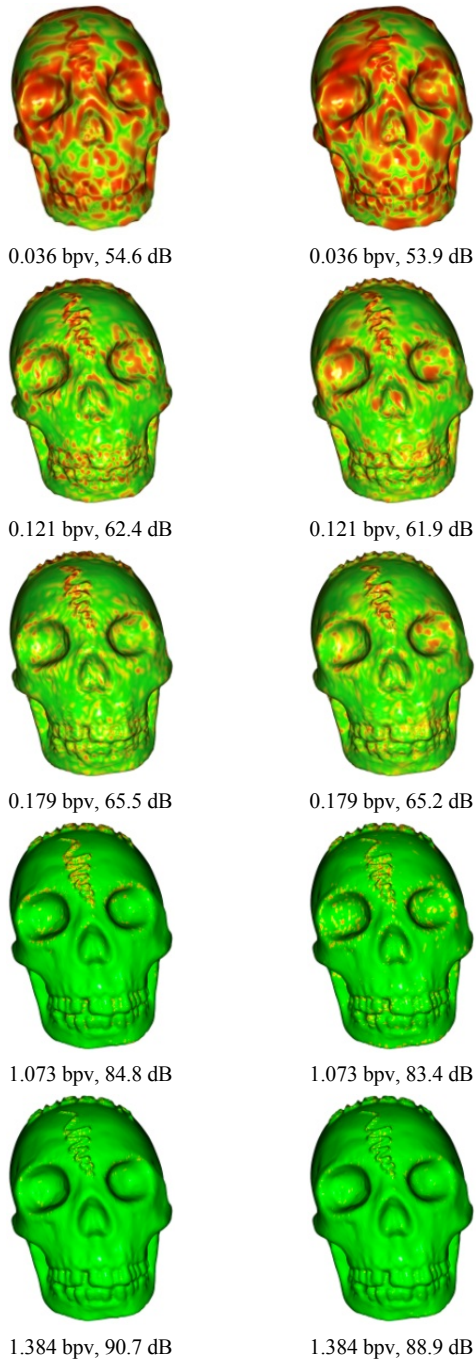


Figure 3.18: Visual comparison of normal Skull mesh using (left column) the 3xC codec and (right column) the PGC codec. The red color intensity reflects the distortion with respect to the uncompressed semi-regular mesh. The rate for the base mesh is not included in the reported rate values.

compared to the PGC system.

The visual comparisons of the normal mesh Skull at different bpv are shown in Figure 3.18. Though, at first glance it may appear that both codecs perform very similar, small differences are noticeable when investigating the meshes more closely. When examining the comparison at 0.036 bpv, we notice that the PGC codec preserves more details in Skull's teeth. The green shade for 3xC at rate 0.179 bpv, however, seems more pure compared to PGC for which it is rather yellowish green. We also observe that no red regions are present for 3xC at rate 1.073 bpv, whereas some are visible for PGC at the same rate.

3.6 CONCLUSIONS

In this chapter, a constructive methodology for the design of scalable wavelet-based mesh compression systems is proposed. Our design strategy differs from conventional designs which, without any theoretical or practical justification, simply opt for reusing methods from wavelet-based image coding for the design of mesh coding systems. In particular, our methods are motivated by an information-theoretic analysis of the statistical dependencies between wavelet coefficients, which shows that (i) intraband dependencies are systematically stronger than the interband ones for both normal and non-normal meshes, and (ii) composite models are by far the strongest.

We also investigate the optimality of successive approximation quantization, commonly used in scalable image compression, in the context of wavelet-based mesh compression. Using a Laplacian mixture model, it is shown that successive approximation quantization is an acceptable yet in general not an optimal solution.

Anchored in these results, novel intraband and composite coding systems were presented, which improve over the state-of-the-art in scalable mesh compression both in terms of scalability and compression efficiency. With the significantly improved performance results given in this chapter, we claim that the proposed coding systems provide a new state-of-the-art in the area of scalable 3-D semi-regular mesh compression.

As stated before, embedded quantization entails a coding framework wherein a certain embedded quantization level can only be decoded if all coarser levels are available (and are decoded) at the decoder side. Such a scheme is beneficial if the transmission rate needs to be adapted on the fly with respect to the available channel bandwidth. However, if transmission also involves packet losses, which is the case under error-prone network conditions, such a scheme results in an unpredictable

drop in source reconstruction quality. This is due to the fact that same number of packets lost from coarser or finer layers contribute unequally towards the overall source distortion. Moreover, packets belonging to a certain layer are only useful if the corresponding packets belonging to all coarser layers are received at the decoder side, otherwise, even if correctly received, they cannot be used to lower the source distortion. Conventionally, FEC is used in conjunction with scalable coding to provide protection against packet erasures. However, FEC schemes, due to a fix code strength, suffers from what is known as the *cliff effect* and a graceful degradation of the source reconstruction quality is not viable. An alternative solution to this problem, referred to as the multiple description coding is studied in the next chapter.

3.7 APPENDIX

The output distortion D_L of a Laplacian PDF, quantized using an n level EDSQ and reconstructed using midpoint reconstruction, can be written as:

$$D_L(Q_{\xi_n, \Delta_n}) = \underbrace{2 \int_0^{(1-\xi_n)\Delta_n} x^2 \frac{\lambda}{2} e^{-\lambda x} dx}_{D_{DZ}} + \sum_{k=1}^{\infty} \underbrace{2 \int_{(k-\xi_n)\Delta_n}^{(k+1-\xi_n)\Delta_n} (x - (k+0.5-\xi_n)\Delta_n)^2 \frac{\lambda}{2} e^{-\lambda x} dx}_{D_k},$$

where D_{DZ} and D_k denote the distortion contributions of the dead-zone and the k th quantization cell, respectively. Since the dead-zone cell has equal sized positive and negative parts, D_{DZ} is the twice the distortion caused by the positive part of the source PDF, hence the factor 2 in the first part of the above equation. Moreover, since $D_k = D_{-k}$, so $D_k + D_{-k} = 2D_k$, hence the factor 2 in the second part of the above equation. Substituting $x - (k+0.5-\xi_n)\Delta_n = y$ in the above equation, we can write:

$$\begin{aligned} D_L(Q_{\xi_n, \Delta_n}) &= \lambda \int_0^{\frac{\Delta_n}{2}} x^2 e^{-\lambda x} dx + \lambda \int_{\frac{\Delta_n}{2}}^{(1-\xi_n)\Delta_n} x^2 e^{-\lambda x} dx \\ &\quad + \left(\sum_{k=1}^{\infty} (e^{-\lambda \Delta_n})^k \right) \left(\lambda \int_{\frac{\Delta_n}{2}}^{\frac{\Delta_n}{2}} y^2 e^{-\lambda y} dy \right) e^{-\lambda \Delta_n (0.5-\xi_n)}. \end{aligned} \quad (3.35)$$

Let $e^{-\lambda \Delta_n k} = (e^{-\lambda \Delta_n})^k$, since $e^{-\lambda \Delta_n} \leq 1$, the summation in the above equation reduces to:

$$\sum_{k=1}^{\infty} e^{-\lambda \Delta_n k} = \frac{e^{-\lambda \Delta_n}}{1 - e^{-\lambda \Delta_n}}. \quad (3.36)$$

Simplification of the above equation results in the following closed-form expression:

$$D_L(Q_{\delta_n, \Delta_n}) = \frac{2}{\lambda^2} + e^{-\lambda \Delta_n \delta_n} \left\{ \left(\frac{1}{4} - \delta_n^2 \right) \Delta_n^2 - \left(2\delta_n + \coth \left(\frac{\lambda \Delta_n}{2} \right) \right) \frac{\Delta_n}{\lambda} \right\}, \quad (3.37)$$

where $\delta_n = 1 - \xi_n$.

Similarly, the output rate R_L of a Laplacian PDF, quantized using an n level EDSQ can be written as:

$$R_L(Q_{\xi_n, \Delta_n}) = -2 \underbrace{\left(\frac{\lambda}{2} \int_0^{(1-\xi_n)\Delta_n} e^{-\lambda x} dx \right) \log_2 2 \left(\frac{\lambda}{2} \int_0^{(1-\xi_n)\Delta_n} e^{-\lambda x} dx \right)}_{R_{DZ}} \dots$$

$$\dots - \sum_{k=1}^{\infty} \underbrace{2 \left(\frac{\lambda}{2} \int_{(k-\xi_n)\Delta_n}^{(k+1-\xi_n)\Delta_n} e^{-\lambda x} dx \right) \log_2 \left(\frac{\lambda}{2} \int_{(k-\xi_n)\Delta_n}^{(k+1-\xi_n)\Delta_n} e^{-\lambda x} dx \right)}_{R_k}.$$

Substituting $x - (k + 0.5 - \xi_n)\Delta_n = y$ in the above equation, we can write:

$$R_L(Q_{\xi_n, \Delta_n}) = - \left(1 - e^{-\lambda \Delta_n (1-\xi_n)} \right) \log_2 \left(1 - e^{-\lambda \Delta_n (1-\xi_n)} \right)$$

$$- \sum_{k=1}^{\infty} \left(e^{-\lambda \Delta_n (k-\xi_n)} - e^{-\lambda \Delta_n (k+1-\xi_n)} \right) \log_2 \left(\frac{e^{-\lambda \Delta_n (k-\xi_n)} - e^{-\lambda \Delta_n (k+1-\xi_n)}}{2} \right).$$

(3.38)

Again, making use of the summation reduction identity of (3.36) together with the following identity

$$\sum_{k=1}^{\infty} e^{-\lambda \Delta_n k} \log_2 \left(e^{-\lambda \Delta_n k} \right) = \log_2 \left(e^{-\lambda \Delta_n} \right) \sum_{k=1}^{\infty} k \left(e^{-\lambda \Delta_n} \right)^k$$

$$= \frac{\log_2 \left(e^{-\lambda \Delta_n} \right) e^{-\lambda \Delta_n}}{\left(1 - e^{-\lambda \Delta_n} \right)^2}, \quad \text{since } \sum_{k=1}^{\infty} k \alpha^k = \frac{\alpha}{(1-\alpha)^2} \text{ when } \alpha < 1,$$

the expression for the rate in equation (3.38) can be reduced to the following closed-form:

$$R_L(Q_{\delta_n, \Delta_n}) = c_{\delta_n} \log_2 \left(\frac{2d_{\delta_n}}{d_1 c_1^{1/d_1} e^{\lambda \Delta_n (1-\delta_n)} d_{\delta_n}^{1/c_{\delta_n}}} \right),$$

(3.39)

where, $c_{\delta_n} = e^{-\lambda \Delta_n \delta_n}$ (hence $c_1 = e^{-\lambda \Delta_n}$) and $d_{\delta_n} = 1 - c_{\delta_n}$ (hence $d_1 = 1 - c_1$).

Chapter 4

MULTIPLE DESCRIPTION CODING: AN OVERVIEW

4.1 INTRODUCTION

So far, our discussion entails single description coding, where a source is quantized to produce a single sequence of quantization indices. In the recent past, a different framework, known as *multiple description coding* (MDC), is shown to be beneficial for communication systems that use diversity to overcome channel impairments [48]. MDC generates multiple mutually refinable source representations (or descriptions), which are then transmitted via separate on/off channels. Any subset of generated descriptions, received at the decoder, can be used to reconstruct the source with certain fidelity and the reconstruction quality strictly improves with the number of description received at the decoder. In this setting, the reliability of the source transmission increases, as in case in which one or more channels fail, the receiver can still reconstruct the source by using the information received via the successful channels. In general, the descriptions transmitted over different channels must be different since the reception of multiple redundant copies of the same information will have no additional value. Additionally, the produced descriptions should also be good individually so that even the reception of a single description can result in some meaningful reconstruction of the source. In general, in MDC, there lies a trade-off between the standalone and joint accuracy of the produced descriptions. This is because the descriptions can be good either individually or jointly but not both [48]. Hence, an MDC system has to operate between the limits set by the individual and joint accuracy of descriptions. In theory, MDC falls under the category of joint source-channel coding (JSCC) paradigm, wherein both source and channel coding can be performed in a combined framework.

At the first look, one may find similarities between MDC and scalable source coding due to the fact that the latter consists of different refinement layers which are

analogous to different source descriptions of MDC. However, a closer look reveals a major difference between the two coding frameworks. In case of scalable coding, a given refinement layer can be decoded and used to improve the reconstruction quality of the input source only if all previous layers are available. In contrast to scalable coding, descriptions in MDC have no dependency and can be individually decoded. In general, scalable coding is a framework where, by using a single coded bit-stream, the source quality can be traded for the source rate in the absence of channel losses. On the other hand, MDC represents a coding paradigm where a smooth rate-distortion trade-off can be achieved in the presence of channel erasures.

In the recent years, MDC has emerged as an attractive solution for error-resilient coding in error-prone packet-switched networks [48]. MDC avoids re-transmission of lost packets, and thus descriptions, if a decoder's quality constraint is met upon reception of a subset of descriptions. In real networks with a large number of receivers, re-transmission of lost packets is not always feasible since it can lead to network congestion [6]. Moreover, in low delay applications, such as video conferencing, peer-to-peer video streaming [81], and real-time delivery of video [13], [117], the use of re-transmission for error resilience is typically not an option since it introduces a significant structural delay which is prohibitive for such applications. In these cases, MDC forms an appealing alternative.

MDC finds its roots in what is known as the *multiple description problem* [48] in information theory. In particular, the multiple description problem is an extension of source coding subject to the fidelity criterion problem [99]. In the scientific community, the multiple description problem has been extensively studied from information theoretic as well as from practical point of views. In this chapter, we will provide a general overview of MDC considering both of these aspects. Since multiple description quantization is one of the main topics of this dissertation, special emphasis is put on the quantization based MDC frameworks.

4.2 MULTIPLE DESCRIPTION PROBLEM

The so-called multiple description problem was first posed by Gersho, Ozarow, Witsenhausen, Wolf, Wyner, and Ziv in 1979's information theory workshop. Before, it was known as the *channel splitting problem* within Bell Laboratories, and later, it was introduced as the multiple description problem to the information theory community. For a two-description case the multiple description problem states as follows: If an information source is described using two descriptions with a total rate of R bps, what is the lowest fidelity of source reconstruction that can be achieved by separate and joint decoding of the produced descriptions?

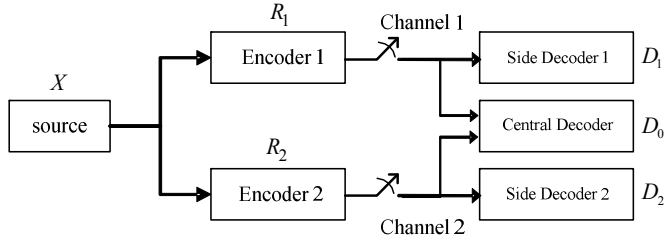


Figure 4.1: Two-description lossy source coding. For the general M description case there will be 2^{M-1} decoders [48].

A pictorial representation of the multiple description problem is shown in Figure 4.1. A source random variable X is encoded by two encoders, also referred to as the *side encoders*, to produce two source descriptions at rates R_1 and R_2 bps, respectively. The produced descriptions are sent over two independent on/off channels. Let D_1 , D_2 and D_0 denote the reconstructed distortions corresponding to the reception of description 1 alone (decoded using the side decoder 1), of description 2 alone (decoded using the side decoder 2), and of both the descriptions (decoded using the central decoder), respectively. In the following section, the achievable performance bounds of the multiple description problem, i.e., the range of achievable values for the quintuple $(R_1, R_2, D_1, D_2, D_0)$, where MSE is taken as the distortion metric, are presented.

4.2.1 Distortion-rate Bound

Let \hat{X}_1 , \hat{X}_2 and \hat{X}_0 denote the reconstruction random variables for the side decoder 1, the side decoder 2 and the central decoder, respectively. Also, let D_1 , D_2 and D_0 be formulated as MSE distortions as follows:

$$D_m = E\left[\left(X - \hat{X}_m\right)^2\right], \text{ for } m = 0, 1, 2. \quad (4.1)$$

Shannon's D-R function $D(R) = \sigma^2 2^{-2R}$ [21] imposes that the decoder m receiving the rate R_m bps cannot have distortion less than $D(R_m)$, i.e.,

$$D_1 \geq D(R_1), \quad D_2 \geq D(R_2), \quad (4.2)$$

$$D_0 \geq D(R_1 + R_2). \quad (4.3)$$

Ahlsvede [4] investigated the tightness of the above inequalities in the no *access rate* regime, i.e., the situation when equality is achieved in equation (4.3). Later, Zang [125] studied the above inequalities for the access rate sum. Achieving equality in both equations (4.2), (4.3) means that the optimum distortion for the sum rate, i.e., $D(R_1 + R_2)$ and the optimum distortion for individual rates, i.e., $D(R_1)$, $D(R_2)$, can be simultaneously reached. In general, descriptions achieving equality in (4.2) should be quite similar and fairly or totally redundant and thus cannot

further improve on $D(R_1)$ (or $D(R_2)$) to reach $D(R_1 + R_2)$. Thus, the fundamental trade-off of MDC is to generate descriptions which are individually good and yet sufficiently different.

El Gamal and Cover [34] first proposed the achievable MDC rate-distortion regions for a memoryless source. Later, Ozarow [80] demonstrated that the rate-distortion region of [34] is tight for the Gaussian sources when evaluated using the MSE distortion metric. He derived the following rate-distortion bound for a unit-variance Gaussian source,

$$D_1 \geq 2^{-2R_1}, \quad D_2 \geq 2^{-2R_2}, \quad (4.4)$$

$$D_0 \geq \frac{2^{-2(R_1+R_2)}}{1 - (\sqrt{\Pi} - \sqrt{\Delta})^2}, \quad (4.5)$$

where

$$\Pi = (1 - D_1)(1 - D_2), \quad (4.6)$$

and

$$\Delta = D_1 D_2 - 2^{-2(R_1+R_2)}. \quad (4.7)$$

4.2.2 Discussion

In order to understand the achievable MDC rate-distortion region of (4.4), (4.5), we consider the extreme case when the source descriptions can individually achieve the Shannon D-R bound, i.e., $D_1 = 2^{-2R_1}$, $D_2 = 2^{-2R_2}$, and hence $\Delta = 0$. In this case, D_0 is given by the following relation:

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

which can be further solved [50] to yield

$$D_0 \geq \frac{\min(D_1, D_2)}{2}. \quad (4.9)$$

Equation (4.9) means that joint decoding of descriptions can only result in a slightly better distortion (half of the lesser side distortion) when compared to standalone decoding. On the other hand, if we want to make the joint decoding as efficient as possible, i.e., by taking

$$D_0 = 2^{-2(R_1+R_2)}, \quad (4.10)$$

implies from (4.5) that $\Pi = \Delta$, which leads to

$$D_1 + D_2 = 1 + 2^{-2(R_1+R_2)}. \quad (4.11)$$

Equation (4.11) suggests that the only way for joint decoding to approach the Shannon limit is by spending the total rate $R_1 + R_2$ in coding a single description (achieving the distortion $2^{-2(R_1+R_2)}$) while not coding the other description at all

(achieving the distortion equal to the variance, i.e., 1). Thus, with this configuration, efficient central distortion at the cost of poor side distortion can be achieved. The intermediate rate-distortion region, with respect to the aforementioned extremes, can be better understood in the *balanced* case, which is presented next.

4.2.3 Balanced MDC

In the *balanced* MDC case, $R_1 = R_2$ and $D_1 = D_2$. In this case, we have:

$$1 - (\sqrt{\Pi} - \sqrt{\Delta})^2 = 1 - \left((1 - D_1) - \sqrt{D_1^2 - 2^{-4R_1}} \right)^2. \quad (4.12)$$

We describe the balanced MDC case under high-rate assumptions, i.e., when $R_1 = R_2 \gg 1$ and $D_1 = D_2 \ll 1$. For $R_1 \gg 1$, $2^{-4R_1} \approx 0$, and equation (4.12) can be approximated as:

$$\begin{aligned} 1 - (\sqrt{\Pi} - \sqrt{\Delta})^2 &\approx 1 - \left((1 - D_1) - D_1 \right)^2, \\ &= 4D_1 - 4D_1^2, \\ &\approx 4D_1. \end{aligned}$$

Substituting the above derived value of $1 - (\sqrt{\Pi} - \sqrt{\Delta})^2$ in equation (4.5) yields the following approximate relationship:

$$D_0 D_1 \geq \frac{1}{4} 2^{-4R_1}. \quad (4.13)$$

Equation (4.13) describes the lower bound on balanced MDC for a unit-variance Gaussian source. In the context of balanced MDC, the product of side and the central distortions, derived in (4.13), is also referred to as the *granular distortion* [10].

In the following, we analyze the balanced MDC from another point of view. Assume,

$$D_1 = D_2 = 2^{-2(1-\alpha)R_1}, \text{ for } 0 < \alpha \leq 1, \quad (4.14)$$

i.e., the rate-of-decay of the side distortions is sub-optimal. Note that the optimal decay rate in (4.14) with respect to Shannon's $D(R)$ function is achieved at $\alpha = 0$. Under this assumption, the approximate expression for the central distortion D_0 can be computed from equation (4.13) as:

$$D_0 \approx \frac{1}{4} 2^{-2(1+\alpha)R_1}. \quad (4.15)$$

From equation (4.15) it is evident that the penalty α in the rate-of-decay of D_1 and D_2 with respect to Shannon's $D(R)$ function is exactly the gain in the rate-of-decay of D_0 . In other words, for a fixed description rate, one can trade-off side and central distortion at the cost of each other in a balanced MDC setting.

4.2.4 Redundancy

In general, for a fixed description rate, the trade-off between the side and the central distortion is controlled using the *redundancy* between the descriptions. As a rule of thumb, the redundancy between the descriptions should be proportional to the probability of channel failure. In particular, a high (respectively low) redundancy between descriptions is suitable for a high (respectively low) failure probability. This is analogous to channel codes, which require more parity (redundancy) bits for channels that have high likelihood of failing.

The redundancy is the additional rate employed in MDC with respect to single description coding. Thus, for a two-description case, the MDC redundancy can be defined as:

$$\rho = R_1 + R_2 - R_0(D_0), \quad (4.16)$$

where $R_0(D_0)$ is the optimal rate for the single description coding achieving D_0 , as given by the Shannon rate-distortion bound:

$$R_0(D_0) = -\frac{1}{2} \log_2 \left(\frac{\sigma^2}{D_0} \right). \quad (4.17)$$

For a two-description MDC case, we have $0 \leq \rho \leq 2R_0$. To make the redundancy range independent of R_0 one can compute the *normalized redundancy*, given by:

$$\rho' = \frac{\rho}{2R_0}, \quad (4.18)$$

where ρ' ranges in the interval $[0, 1]$.

A different way of looking at the multiple description problem is by partitioning the total rate into the base rate R_0 and the redundancy rate $\rho = R_1 + R_2 - R_0$, where R_0 corresponds to the central distortion D_0 and the redundancy rate ρ is employed to reduce the side distortions D_1, D_2 as much as possible. Using this analogy the side distortion D_1 in the balanced case is lower bounded by the *redundancy-distortion* function $D_1(\rho, D_0)$ [49, 50] given by:

$$D_1(\rho, D_0) \geq \begin{cases} \frac{1}{2} \left[\left(1 - \sqrt{1 - 2^{-2\rho}} \right) + D_0 \left(1 + \sqrt{1 - 2^{-2\rho}} \right) \right], & \text{for } \rho \leq R_0 - 1 + \log_2(1 + 2^{-2R_0}) \\ \left[1 - \sqrt{1 - 2^{-2\rho}} \right], & \text{for } \rho > R_0 - 1 + \log_2(1 + 2^{-2R_0}) \end{cases} \quad (4.19)$$

In the above equation $D_0 = 2^{-2R_0}$. For the case when the base rate R_0 is sufficiently large, i.e., $R_0 \rightarrow \infty$, and the redundancy is far below R_0 , the bound in equation (4.19) becomes:

$$D_1(\rho) = \lim_{\substack{D_0 \rightarrow 0 \\ R_0 \rightarrow \infty}} D_1(\rho, D_0) = \frac{1}{2} \left(1 - \sqrt{1 - 2^{-2\rho}} \right). \quad (4.20)$$

For ρ large enough to have $2^{-2\rho} \ll 1$, $\sqrt{1-2^{-2\rho}}$ can be approximated by $(1-2^{-2\rho-1})$, yielding:

$$D_1(\rho) \approx \frac{1}{4} 2^{-2\rho}, \text{ for } 2^{-2\rho} \ll 1. \quad (4.21)$$

Comparing equation (4.14) with equation (4.21), we can conclude that ρ and α exhibit an inverse relationship, i.e., a small value of α corresponds to high redundancy and vice versa.

4.3 MULTIPLE DESCRIPTION SCALAR QUANTIZATION

The focus of the previous section was to enable a theoretical understanding of the multiple description problem. In this section, we highlight the practical code constructions for MDC.

In 1993, Vaishampayan proposed the first practical scheme to realize a balanced MDC framework [113]. The scheme of [113], which was based on scalar quantization methods, later gained popularity under the name multiple description scalar quantizer (MDSQ). Conceptually, an MDSQ can be viewed as a combination of two independent scalar quantizers, which produce equal side distortions. However, when joined together, they realize a finer scalar quantizer, which can be used to achieve a lower central distortion.

For explanatory purposes, we consider a simple MDSQ example depicted in Figure 4.2. Let the two uniform scalar quantizers Q_1, Q_2 , which are shifted with respect to each other with half the quantization cell size, form two side encoder-decoder pairs. Each source sample is quantized by Q_1 and Q_2 to produce two quantization indices, which identify the quantization cells of Q_1 and Q_2 that the sample belongs to. These two quantization indices jointly identify a particular cell in Q_0 , where Q_0 is created by the intersections of the quantization cells of Q_1 and Q_2 . In this example, due to the fact that the cell size of Q_0 is half of that of Q_1 (or Q_2), the central distortion D_0 is approximately a quarter of the side distortions D_1 (or D_2) if the reconstruction is performed at the center of each cell for both side and the central quantizers.



Figure 4.2: An example of a balanced two-description MDSQ with the center of the cell reconstruction.

4.3.1 Index Assignment of the Central Quantizer's Indices

In this section, we describe an MDSQ [113] in a more formal manner. Let $\mathcal{I}_1 = \{1, 2, 3, \dots, M_1\}$ and $\mathcal{I}_2 = \{1, 2, 3, \dots, M_2\}$ denote the index sets of Q_1 and Q_2 , respectively. In the particular case of a balanced MDSQ, i.e., $M_1 = M_2 = M$, Q_1 and Q_2 map the real-valued source sample to the indices i and j , where $i \in \mathcal{I}_1$ and $j \in \mathcal{I}_2$. Let \mathcal{C} , a subset of $\mathcal{I}_1 \times \mathcal{I}_2$, i.e., $|\mathcal{C}| = N < M_1 \times M_2$, denotes the index set of Q_0 . A two-description MDSQ can then be defined as a combination of two side encoders $Q_1: \mathbb{R} \rightarrow \mathcal{I}_1$, $Q_2: \mathbb{R} \rightarrow \mathcal{I}_2$, two side decoders $Q_1^{-1}: \mathcal{I}_1 \rightarrow \hat{\mathcal{X}}_1$, $Q_2^{-1}: \mathcal{I}_2 \rightarrow \hat{\mathcal{X}}_2$ and a central decoder $Q_0^{-1}: \mathcal{C} \rightarrow \hat{\mathcal{X}}_0$, where $\hat{\mathcal{X}}_1 = \{\hat{x}_i^1, i \in \mathcal{I}_1\}$, $\hat{\mathcal{X}}_2 = \{\hat{x}_j^2, j \in \mathcal{I}_2\}$ and $\hat{\mathcal{X}}_0 = \{\hat{x}_{ij}^0, (i, j) \in \mathcal{C}\}$ are the reconstruction codebooks for Q_1 and Q_2 and Q_0 , respectively.

In the example of Figure 4.2, $M_1 = M_2 = 6$, $N = 11$ and the redundancy for fixed-length coding of indices is $2 \log_2 6 - \log_2 11 = 1.71$ bps. The codebooks for Q_1 , Q_2 and Q_0 can be chosen as the centers (sub-optimal) or the centroids (optimal) of the corresponding quantization cells. For the MDSQ example in Figure 4.2, the rate-of-decay of side and central distortions is asymptotically $O(2^{-2R_1})$ [113], which is optimal for side distortions but not optimal for the central distortion. In Section 4.2.3, it was shown that D_0 can be reduced by reducing the redundancy, which in turn will increase the side distortion D_1 . The practical implementation of such a trade-off is presented next.

The MDSQ of Figure 4.2 can also be created by first quantizing the source with Q_0 and then mapping the produced central quantization index c for each sample to two side indices (i, j) through a mapping function $[i, j] = a(c)$, also referred to as the index assignment (IA) function. The block diagram for such a construction is shown in Figure 4.3(a). For two-description scalar quantizers the IA function can be represented in the form of a matrix, generally referred to as the IA matrix. The IA matrix for the MDSQ example of Figure 4.2 is shown in Figure 4.3(b).

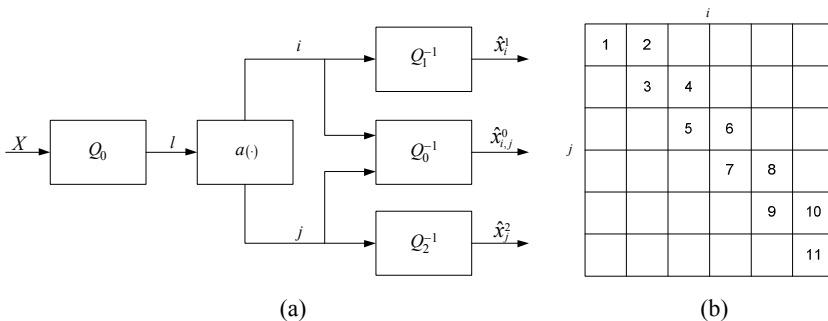


Figure 4.3: An alternative understanding of MDSQ. (a) central quantizer followed by index assignment (b) index assignment matrix for the MDSQ of Figure 4.2.

In Figure 4.3(b) the index entries of Q_0 are organized as two diagonals in the IA matrix. For any M , such an arrangement yields $N = 2M - 1$ quantization bins of Q_0 . To increase the rate of decay $O(2^{-2R})$ of D_0 , we require N to grow at a faster rate with respect to the number of cells of Q_1 (or Q_2). The additional number of entries can be arranged along the remaining diagonals of the IA matrix in Figure 4.3(b). One such arrangement is depicted in Figure 4.4(a), for which $N > 2M - 1$, and the corresponding MDSQ partitions are shown in Figure 4.4(b). The resulting MDSQ has the redundancy of $2 \log_2 6 - \log_2 16 = 1.17$ bps, which is smaller than that of the MDSQ of Figure 4.2. As a result, the quantizer of Figure 4.4 will have an improved central distortion at the cost of increased side distortion. Note that, in contrast to the MDSQ example of Figure 4.2, the MDSQ of Figure 4.4 consists of disconnected side partition cells – see Figure 4.4(b).

We know that, for a fixed M , the number of diagonals (and hence N) can be increased to reduce the redundancy and in turn the central distortion D_0 at the cost of D_1 . The extreme case will occur for $N = M^2$, which completely fills the IA matrix. This corresponds to the lowest redundancy case, for which D_0 has the smallest, while D_1 has the highest magnitude. After this point, the redundancy and hence D_0 can only be further reduced by increasing M , i.e., by increasing the description rate. Opposite to the lowest redundancy case is the full redundancy case which occurs when there are only M entries in the IA matrix which are organized along its main diagonal.

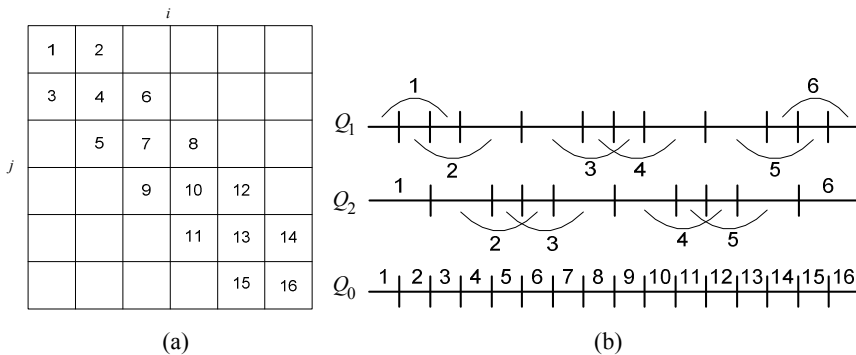


Figure 4.4: MDSQ with disconnected side partition cells. (a) The IA matrix (b) The side and the central quantizers' partitions.

4.3.2 Good Index Assignments

The MDSQ construction, as presented in Figure 4.3(a) and Figure 4.4, gives a practical meaning to the theoretical idea of the base rate R_0 and the redundancy rate ρ introduced in Section 4.2.4. Precisely, the base-rate fixes the number of quantization bins the central quantizer Q_0 must possess to achieve a certain D_0 ,

attain a spread that is close to the theoretically promised one [113]. The corner adjusted nested and linear IAs are referred to as the *modified nested* and *modified linear* IAs, respectively. Few examples of the modified nested and the modified linear IA matrices for different number of diagonals are shown in Figure 4.6 and Figure 4.7, respectively.

| | | | | | | |
|---|---|---|----|----|----|----|
| 1 | 3 | | | | | |
| 2 | 4 | 5 | | | | |
| | 6 | 7 | 9 | | | |
| | | 8 | 10 | 11 | | |
| | | | 12 | 13 | 15 | |
| | | | | 14 | 16 | 17 |
| | | | | | 18 | 19 |
| | | | | | | 20 |
| | | | | | | 22 |

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

| | | | | | | |
|---|----|----|----|----|----|----|
| 1 | 3 | 5 | 7 | | | |
| 2 | 8 | 10 | 12 | 14 | | |
| 4 | 9 | 15 | 17 | 19 | 21 | |
| 6 | 11 | 16 | 22 | 23 | 25 | 27 |
| | 13 | 18 | 24 | 29 | 30 | 32 |
| | | 20 | 26 | 31 | 36 | 37 |
| | | | 28 | 33 | 38 | 41 |
| | | | | 35 | 40 | 42 |
| | | | | | 43 | 44 |

Figure 4.6: Modified nested IA matrices. (left) $k = 1$, (center) $k = 2$, (right) $k = 3$.

| | | | | | | |
|---|---|---|----|----|----|----|
| 1 | 2 | | | | | |
| 3 | 4 | 6 | | | | |
| | 5 | 7 | 8 | | | |
| | | 9 | 10 | 11 | | |
| | | | 12 | 14 | | |
| | | | 13 | 15 | 16 | |
| | | | | 17 | 18 | 20 |
| | | | | | 19 | 21 |

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

| | | | | | | |
|----|----|----|----|----|----|----|
| 1 | 2 | 4 | 7 | | | |
| 3 | 5 | 8 | 11 | 14 | | |
| 6 | 9 | 12 | 15 | 18 | | |
| 10 | 13 | 16 | 19 | 21 | 24 | 28 |
| | 17 | 20 | | 23 | 27 | 31 |
| | | | 22 | 26 | 30 | 34 |
| | | | | 25 | 29 | 33 |
| | | | | | 32 | 36 |
| | | | | | | 37 |
| | | | | | | 38 |
| | | | | | | 39 |
| | | | | | | 40 |
| | | | | | | 41 |

Figure 4.7: Modified linear IA matrices. (left) $k = 1$, (center) $k = 2$, (right) $k = 3$.

4.3.3 Source Specific MDSQ Optimization

Besides designing efficient IA strategies, the other main contribution of Vaishampayan's work was the optimization of MDSQs with respect to the source PDF [113, 114]. To give a detailed insight into the optimization algorithm, we define the quantization distortions D_1 , D_2 and D_0 as:

$$D_1 = E\left[\left(X - \hat{X}_1\right)^2\right] = \sum_{i \in \mathcal{I}_1} \int_{\mathcal{A}_i} \left(x - \hat{x}_i^1\right)^2 f(x) dx, \quad (4.23)$$

$$D_2 = E\left[\left(X - \hat{X}_2\right)^2\right] = \sum_{j \in \mathcal{I}_2} \int_{\mathcal{A}_j} \left(x - \hat{x}_j^2\right)^2 f(x) dx, \quad (4.24)$$

$$D_0 = E\left[\left(X - \hat{X}_0\right)^2\right] = \sum_{(i,j) \in \mathcal{C}} \int_{\mathcal{A}_{i,j}} \left(x - \hat{x}_{ij}^0\right)^2 f(x) dx. \quad (4.25)$$

\mathcal{A}_i , \mathcal{A}_j and $\mathcal{A}_{i,j}$ specify the sides and central quantizer cells. Similarly, the entropies of the side quantizers indices are defined as:

$$H_1 = - \sum_{i \in \mathcal{I}_1} p_i \log_2(p_i), \text{ with } p_i = \int_{\mathcal{A}_i} f(x) dx, \quad (4.26)$$

$$H_2 = - \sum_{j \in \mathcal{I}_2} q_j \log_2(q_j), \text{ with } q_j = \int_{\mathcal{A}_j} f(x) dx. \quad (4.27)$$

The aim is to jointly optimize the side and central quantizers. The optimization is formulated as a minimization problem as follows:

$$\min D_0 \text{ subject to } D_1 \leq D'_1, D_2 \leq D'_2, H_1 \leq R'_1, H_2 \leq R'_2. \quad (4.28)$$

D'_1 (respectively D'_2) and R'_1 (respectively R'_2) in the above equation respectively denote the target values of distortion and rate of the first (respectively the second) side description. The Lagrangian functional for the optimization problem is then,

$$L(\mathbf{f}, \mathbf{g}, \lambda_1, \lambda_2, \mu_1, \mu_2) = D_0 + \lambda_1 (D_1 - D'_1) + \lambda_2 (D_2 - D'_2) + \mu_1 (H_1 - R'_1) + \mu_2 (H_2 - R'_2) \quad (4.29)$$

Similar to the design of optimal scalar quantizers, i.e., the Lloyd and Max case [65, 70], one can employ an *iterative descent algorithm* to find an optimum encoder-decoder pair $(\mathbf{f}^*, \mathbf{g}^*)$ for which the cost in the equation (4.29) is minimum [113]. In particular, using a training sequence $x(t)$, $t=1,2,3,\dots,T$, of samples drawn from the source distribution for which quantizer optimization needs to be carried out, the encoder-decoder pair is iteratively optimized by keeping one component fixed while changing the other [65, 70]. Such a design algorithm generates a non-increasing sequence of Lagrangian cost values. The algorithm must converge since the minimum Lagrangian cost value is lower bounded by zero. However, the finally obtained limit point may be a local minimum or just a saddle point and a good initialization condition is necessary to realize a good final solution.

4.3.3.1 Optimum decoder (for a given encoder)

For a given encoder \mathbf{f} the optimal decoder \mathbf{g}^* is the one that reconstructs all source samples within a given quantization cell using the centroid value, i.e., the centroid condition – see Section 2.4.2.2. Based on this analogy, the optimum sides and central decoder are given by [113, 114],

$$\hat{x}_i^1 = E[X/i] = \frac{\int_{\mathcal{A}_i} xf(x) dx}{\int_{\mathcal{A}_i} f(x) dx} = \frac{1}{|x(t) \in \mathcal{A}_i|} \sum_{x(t) \in \mathcal{A}_i} x(t), \text{ for } i \in \mathcal{I}_1, \quad (4.30)$$

$$\hat{x}_j^2 = E[X/j] = \frac{\int_{\mathcal{A}_j} xf(x) dx}{\int_{\mathcal{A}_j} f(x) dx} = \frac{1}{|x(t) \in \mathcal{A}_j|} \sum_{x(t) \in \mathcal{A}_j} x(t), \text{ for } j \in \mathcal{I}_2, \quad (4.31)$$

$$\hat{x}_{i,j}^0 = E[X|i,j] = \frac{\int_{\mathcal{A}_{i,j}} xf(x) dx}{\int_{\mathcal{A}_{i,j}} f(x) dx} = \frac{1}{|x(t) \in \mathcal{A}_{i,j}|} \sum_{x(t) \in \mathcal{A}_{i,j}} x(t), \text{ for } (i,j) \in \mathcal{C} \quad (4.32)$$

4.3.3.2 Optimum encoder (for a given decoder)

The optimum encoder \mathbf{f}^* for a given decoder \mathbf{g} is given by the minimum of the scalar cost function of equation (4.29), i.e.,

$$\mathbf{f}^* = \arg \min_{\mathbf{f}} L(\mathbf{f}, \mathbf{g}, \lambda_1, \lambda_2, \mu_1, \mu_2). \quad (4.33)$$

The Lagrangian cost function in the above minimization problem can be simplified as follows:

$$\begin{aligned} & L(\mathbf{f}, \mathbf{g}, \lambda_1, \lambda_2, \mu_1, \mu_2) \\ &= \left\{ \begin{aligned} & \sum_{(i,j) \in \mathcal{C}} \int_{\mathcal{A}_{i,j}} (x - \hat{x}_{ij}^0)^2 f(x) dx + \lambda_1 \sum_{i \in \mathcal{I}_1} \int_{\mathcal{A}_i} (x - \hat{x}_i^1)^2 f(x) dx + \\ & \lambda_2 \sum_{j \in \mathcal{I}_2} \int_{\mathcal{A}_j} (x - \hat{x}_j^2)^2 f(x) dx - \mu_1 \sum_{i \in \mathcal{I}_1} p_i \log_2(p_i) - \mu_2 \sum_{j \in \mathcal{I}_2} q_j \log_2(q_j) \end{aligned} \right\}, \\ &= \int_{-\infty}^{\infty} \left[(x - \hat{x}_{ij}^0)^2 + \lambda_1 (x - \hat{x}_i^1)^2 + \lambda_2 (x - \hat{x}_j^2)^2 - \mu_1 \log_2 p_i - \mu_2 \log_2 q_j \right] f(x) dx, \end{aligned} \quad (4.34)$$

or

$$L(\mathbf{f}, \mathbf{g}, \lambda_1, \lambda_2, \mu_1, \mu_2) = E \left[\begin{aligned} & (x - \hat{x}_{ij}^0)^2 + \lambda_1 (x - \hat{x}_i^1)^2 + \lambda_2 (x - \hat{x}_j^2)^2 \\ & - \mu_1 \log_2 p_i - \mu_2 \log_2 q_j \end{aligned} \right]. \quad (4.35)$$

From equation (4.35) it is evident that the minimum of the cost function $L(\mathbf{f}, \mathbf{g}, \lambda_1, \lambda_2, \mu_1, \mu_2)$ is achieved when the expression inside the expectation operator is individually minimized for each sample of the training sequence. This can be accomplished by optimally mapping each sample $x(t)$ to those side quantizer bins (i, j) that result in the smallest value of the expression within the expectation operator. Since the central quantizer together with the IA matrix defines the side quantizer cells, such an optimal mapping is equivalent to creating an optimal central quantizer partition. Using the training sequence, the expression inside the expectation operator can be minimized by defining the central quantizer Q_0 as:

$$\mathcal{A}_{i,j} = \left\{ \begin{aligned} & x : (x - \hat{x}_{i,j}^0)^2 + \lambda_1 (x - \hat{x}_i^1)^2 + \lambda_2 (x - \hat{x}_j^2)^2 - \mu_1 \log_2 p_i - \mu_2 \log_2 q_j \\ & \leq \\ & (x - \hat{x}_{i',j'}^0)^2 + \lambda_1 (x - \hat{x}_{i'}^1)^2 + \lambda_2 (x - \hat{x}_{j'}^2)^2 - \mu_1 \log_2 p_{i'} - \mu_2 \log_2 q_{j'}, \\ & \quad \forall (i, j) \neq (i', j') \end{aligned} \right\} \quad (4.36)$$

where $p_i = |x(t) \in i|/T$ denotes the probability that a training sample belong to the i th side quantizer cell. Defining

$$\alpha_{ij} = (\hat{x}_{i,j}^0) + \lambda_1 (\hat{x}_i^1) + \lambda_2 (\hat{x}_j^2), \quad (4.37)$$

and

$$\beta_{ij} = (\hat{x}_{i,j}^0)^2 + \lambda_1 (\hat{x}_i^1)^2 + \lambda_2 (\hat{x}_j^2)^2 - \mu_1 \log_2 p_i - \mu_2 \log_2 p_j, \quad (4.38)$$

equation (4.36) simplifies to [113, 114]:

$$\mathcal{A}_{i,j} = \left\{ x : 2\alpha_{ij} - \beta_{ij} \geq 2\alpha_{i'j'} - \beta_{i'j'}, \forall (i, j) \neq (i', j') \right\}. \quad (4.39)$$

The above central quantizer partition, for a given set of reconstruction values, i.e., $\hat{x}_{i,j}^0$'s, \hat{x}_i^1 's and \hat{x}_j^2 's, can be determined using the *extreme point algorithm* described in [113].

4.3.3.3 Discussion

In case the aim is to have equal side entropies $H_1 = H_2$ and side distortions $D_1 = D_2$, i.e., the case of balanced MDSQ, we can take $\lambda_1 = \lambda_2 = \lambda$ and $\mu_1 = \mu_2 = \mu$ in equations (4.34)-(4.38). However, $\lambda_1 = \lambda_2 = \lambda$ and $\mu_1 = \mu_2 = \mu$ can only lead to balanced side distortions if the un-optimized MDSQ is balanced [113]. For a symmetric source PDF, perfectly balanced descriptions are created if for every quantization cell in one side quantizer there exists a corresponding cell in the second side quantizer with exactly the same but opposite signed thresholds. In this case, such two cells contribute equal distortion and entropy contributions hence resulting in $D_1 = D_2$ and $H_1 = H_2$. We notice that, for the linear IAs in Figure 4.7, the side cell pair $(i, M - i + 1)$ from two side quantizers have exactly opposite threshold values for a uniform central quantizer. This is not true for the nested IAs depicted in Figure 4.6. However, due to a roughly identical behavior of the two side quantizers of the nested IA, the side distortions are roughly balanced [113], i.e., $D_1 \approx D_2$ and $H_1 \approx H_2$. Since the linear IA yields perfectly balanced side quantizers for symmetric sources, in such cases, one need to only compute the distortion and the rate quantities corresponding to a single side quantizer in equations (4.34)-(4.38). In this case, when $\lambda_1 = \lambda_2 = \lambda$ and $\mu_1 = \mu_2 = \mu$, α_{ij} and β_{ij} simplifies to:

$$\alpha_{ij} = (\hat{x}_{i,j}^0) + 2\lambda(\hat{x}_i^1), \quad \beta_{ij} = (\hat{x}_{i,j}^0)^2 + 2\lambda(\hat{x}_i^1)^2 - 2\mu \log_2 p_i.$$

For a constant value of λ and μ , the iterative solution to the optimization problem of equation (4.28) is given in [113, 114]. The appropriate λ , μ are any set of values which fulfill the constraints on side rates and side distortions, as given in (4.28). In general, it is quite tedious to find one combination of λ , μ values that gives a true minimum of D_0 , while still meeting the distortions and rate constraints. One method is by successively narrowing the possible range of values between two extreme points using the bisection method [108].

When the quantization indices are encoded using a fixed-length code, e.g., NBC or gray code, the two entropy constraints in equation (4.28) become irrelevant, i.e., one can set $\mu = 0$. However, when a variable-length code is employed, e.g., a Huffman code or an arithmetic code, both λ and μ need to be carefully searched to figure out a solution that meets both the distortion and the entropy constraints given in (4.28).

The method of Vaishampayan [113, 114] generalizes the concept of optimal single description scalar quantization to the multiple description scalar quantization case. In Vaishampayan's optimization method, a certain IA serves as an initialization condition to the algorithm. We can conjecture that, using this algorithm, a true locally optimal MDSQ performance can only be achieved if the IA is optimal. In general, searching for an optimal IA is a complex combinatorial problem [11] even for the rather simple case of $M = 2$. The nested and linear IAs proposed in [113] are not optimal. Theoretically speaking, a large gap may exist between the performance of a locally optimal MDSQ based on a non-optimal IA and a locally optimal MDSQ based on an optimal IA.

4.3.4 High-Resolution D-R Function of MDSQ

For generalized smooth source densities and assuming a generic IA matrix, the high-rate D-R function of a balanced MDSQ is derived in [10]. For a generic source of variance σ^2 the D-R function is given by the following set of equations [10]:

$$D_1(R) = D_2(R) \approx S\sigma^2 2^{-2R(1-\alpha)}, \quad (4.40)$$

$$D_0(R) \approx C\sigma^2 2^{-2R(1+\alpha)}. \quad (4.41)$$

where C and S are constant factors depending on the source probability density function and the encoding method. Also, $\alpha \in (0,1]$ denotes the redundancy control factor which is related to the total number of diagonals $d = (2k+1)$, in the IA matrix. k and α are related with $k = 2^{\alpha R}$ [10]. It is explained before that small values of α correspond to high redundancy and vice versa.

In the level-constrained MDSQ case [113], for a Gaussian source, $C = \sqrt{3}\pi/2$ and $S = \sqrt{3}\pi/8$ are derived in [10]. The granular distortion (which is defined as the product of side and the central distortion) for a unit-variance Gaussian source can then be approximated as [10]:

$$D_0(R)D_1(R) \approx \frac{3\pi^2}{16} 2^{-4R}. \quad (4.42)$$

Since the lower bound on the granular distortion is given by $D_0(R)D_1(R) \approx (1/4)2^{-4R}$ (see Section 4.2.3), there is a 8.69 dB performance gap between the optimum level-constrained quantizer and the multiple description D-R bound.

In the entropy-constrained case, for a Gaussian source, $C = \pi e/6$, $S = \pi e/24$ [10] and the granular distortion for a unit-variance Gaussian source can be approximated as:

$$D_0(R)D_1(R) \approx \frac{\pi^2 e^2}{144} 2^{-4R}. \quad (4.43)$$

When compared to $D_0(R)D_1(R) \approx (1/4)2^{-4R}$, there is a 3.07 dB gap, which is a 5.62 dB improvement over the level-constrained case. The gap between the actual performance of the quantizer and the bound can be reduced by making use of lattice based vector quantizers [47, 96, 116] for multiple description coding.

In a setting where both on/off channels are equally likely to fail with a failure probability p_L , the average distortion \bar{D} at the decoder for a σ^2 -variance source is given by:

$$\bar{D} = (1 - p_L)^2 D_0(R) + 2p_L(1 - p_L)D_1(R) + p_L^2\sigma^2. \quad (4.44)$$

By substituting for the central and the side distortion from equations (4.40), (4.41), \bar{D} can be written as:

$$\bar{D} = (1 - p_L)^2 C\sigma^2 2^{-2R(1+\alpha)} + 2p_L(1 - p_L)S\sigma^2 2^{-2R(1-\alpha)} + p_L^2\sigma^2 \quad (4.45)$$

Minimizing the above equation with respect to the redundancy factor α , i.e., by solving $\partial\bar{D}/\partial\alpha = 0$, the optimum value of α can be found as:

$$\alpha^* = -\frac{1}{4R} \log_2 \left(\frac{2p_L S}{(1-p_L)C} \right) = -\frac{1}{4R} \log_2 \left(\frac{8p_L}{1-p_L} \right). \quad (4.46)$$

Since $k = 2^{\alpha R}$, the optimal value of k is given by:

$$k^* = 2^{-\frac{1}{4} \log_2 \left(\frac{8p_L}{1-p_L} \right)}. \quad (4.47)$$

Note that the value of k^* computed using the above equation may not to be an integer value. Since the number of diagonals of an MDSQ can only be an integer number, the fractional value of k^* obtained from equation (4.47) is rounded to the nearest integer to compute the optimum number of diagonals for a given p_L . The positivity of α^* demands

$$\frac{8p_L}{1-p_L} \leq 0, \quad (4.48)$$

which is only satisfied if and only if $p_L \leq 1/9 \approx 0.11$. Thus, one can conclude that, in the asymptotic sense, an MDSQ based MDC system cannot provide *optimum protection* in transmission situations wherein the failure probability is higher than 11%. However, this does not mean that MDSQs are not beneficial if $p_L > 1/9$. In general, MDSQ based MDC can be used to enable error-resilience for packet loss rate above 11%.

4.3.5 Two-stage MDSQ

Interestingly, one can achieve the performance of an entropy-constrained MDSQ [114] using a much simpler IA defined using a two-stage framework, as proposed by Tian et al. in [109]. The first stage in a two-stage MDSQ consists of two equal step size uniform quantizers displaced by half of their step size. An example of such

quantizers is shown in Figure 4.2 and the equivalent IA matrix is depicted in Figure 4.3(b). The quantization indices from the first stage side quantizers are entropy coded to form the first parts of two side descriptions. In the second stage, cells of the central quantizer of the first stage are refined using N smaller equal-sized sub-cells. The second stage's refinement information is entropy coded and then evenly split into two halves to form the second parts of the source descriptions. The resulting IA

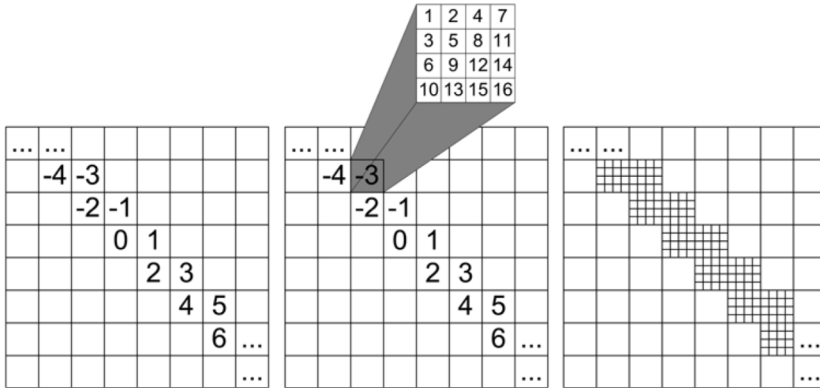


Figure 4.8: IA matrix for a two-stage MDSQ. (Left) The IA matrix for the first stage. (Middle) A central cell of the first stage is subdivided into $N = 16$ sub-bins. (Right) The combined IA matrix for two stages. The figure is directly copied from [109].

matrix for the two-stage MDSQ [109] is depicted in Figure 4.8.

At the decoder, only the first part of a received description is used to perform side decoding and the second part is simply discarded. Since the first part of the description corresponds to the first stage, the side distortion of the two-stage MDSQ for a unit-variance Gaussian source can be approximated as [45]:

$$D_1(R_1) = D_2(R_1) \approx \frac{\pi e}{6} 2^{-2R_1}, \quad (4.49)$$

where R_1 denotes the side rate (in bits per sample per description) for the first stage.

The central decoder uses both parts of the source descriptions to perform central decoding. In other words, the first parts of the two descriptions decodes the source sample to a single central cell, while the second stage refinement information is used to further refine the source sample to one of the N second stage fine sub-cells. Since there are $2N$ fine sub-cells within each side quantizer cell, the central distortion, employing high-rate assumptions, can be approximated as:

$$D_0(R_1 + R_2) \approx \frac{D_1(R)}{(2N)^2}. \quad (4.50)$$

Considering high-rate quantization, the PDF within each side cell can be assumed

approximately constant. With this assumption the rate R_2 (bits per sample per description) for coding the second stage information is $2R_2 = \log_2 N$ or $N = 2^{2R_2}$. By substituting $N = 2^{2R_2}$ in equation (4.50), the central distortion becomes:

$$D_0(R_1 + R_2) \approx \frac{\pi e}{24} 2^{-(2R_1 + 4R_2)}. \quad (4.51)$$

Using equations (4.49) and (4.51), the granular distortion for the two-stage MDSQ, for the total side rate $R = R_1 + R_2$, can be written as [109]:

$$D_0 D_1 \approx \frac{\pi^2 e^2}{144} 2^{-4R}. \quad (4.52)$$

The right hand side of the above expression is identical to the one given in equation (4.43), hence, we can say that the performance of the entropy-constrained MDSQ [113] can be asymptotically achieved by the above described two-stage MDSQ using a much simpler IA strategy.

In [60], a method is proposed for utilizing the second stage information to further reduce the side distortion attained using the first stage alone. In [60], the splitting of the refinement information into two parts is done prior to entropy coding. In particular, the second stage's refinement information corresponding to the odd sub-cells is added to the first description while the information corresponding to the even sub-cells is added to the second description. At the decoder, the second stage information is used to figure out the two odd (upon the reception of the first description) or the two even (upon the reception of the second description) numbered refinement sub-cells that the source sample belongs to. Note that the two refinement bins will always lie within the side quantizer cell, which can be specified using the first stage information. For side decoding, the average of the centers of the two odd or two even numbered refinement bins is used as a reconstruction point [60]. The asymptotic rate-distortion performance of this modified side decoding rule is analyzed in [64]. In [64], it is shown that, with respect to equation (4.49), the modified side decoding rule of [60] asymptotically improves the side distortion performance by 0.58 dB for a unit-variance Gaussian source.

4.4 OTHER PRACTICAL METHODS FOR MDC

Besides the MDSQ framework, which was the initial approach for practical MDC, there exist several other methods to perform MDC of generic sources. In this section, we will briefly sketch an overview of these methods. In addition to the methods described in this section, other MDC algorithms are specific to the type of synthetic sources, e.g., image, video and other types of multimedia data [97], [19], [7], [12], etc. For brevity we will not discuss these methods here.

4.4.1 Multiple Description Transform Codes

Multiple description transform codes [50, 121] are based on correlating transforms that steer the redundancy between transformed coefficients. In the literature, the MDC scheme based on correlating transform is generally referred to as multiple description transform coding (MDTC).

Conceptually, in MDTC, two independent random variables X_1 and X_2 are transformed to two coefficients Y_1 and Y_2 using a correlating, unit-determinant, transform matrix \mathbf{T} [50], i.e.,

$$\underbrace{\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} a & b \\ c & \frac{1+bc}{a} \end{bmatrix}}_{\mathbf{T}} \underbrace{\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}}_{\mathbf{X}}. \quad (4.53)$$

The purpose of the transform matrix \mathbf{T} is to add a controlled amount of redundancy between Y_1 and Y_2 . Since \mathbf{T} is a non-singular matrix, X_1 and X_2 can be uniquely recovered using the relation $\mathbf{X} = \mathbf{T}^{-1}\mathbf{Y}$. If Y_1 (respectively Y_2) is erased by a channel error, an estimate \check{Y}_1 (respectively \check{Y}_2) of the lost coefficient, due to the introduced redundancy between Y_1 and Y_2 , can be created using the successfully received coefficient. Using the estimated lost component, the approximate decoding of X can be performed as $\check{\mathbf{X}}_1 = \mathbf{T}^{-1}[\check{Y}_1 \ Y_2]$, or as $\check{\mathbf{X}}_2 = \mathbf{T}^{-1}[Y_1 \ \check{Y}_2]$, depending on the loss of Y_1 or Y_2 respectively. We note that, in practice, we need to introduce quantization to reduce the precision of the real-valued source samples. In the context of MDTC, quantization is done prior to transformation. This is justified since quantization of the components of \mathbf{Y} is sub-optimal due to non-orthogonal transformation [50].

We explain the MDTC operation using an example. Let X_1 and X_2 are independent zero-mean Gaussian random variables with variances σ_1^2 and σ_2^2 , where $\sigma_1^2 > \sigma_2^2$. First, we evaluate the distortions when no transform coding is employed. Consider single description coding of X_1 and X_2 assuming high-rate entropy-coded uniform quantization. Let's assume that the coded information for X_1 is sent over channel 1 and the coded information for X_2 is sent over channel 2. In this case, the per component squared error at rate R bits per two source samples would be given by [50]:

$$D_0 = \frac{\pi e}{6} \sigma_1 \sigma_2 2^{-2R}. \quad (4.54)$$

Since X_1 and X_2 are independent, no knowledge about the lost random variable can be derived from the received one. In this case, the lost random variable can be best approximated by its mean value. For this case, the single channel distortions D_1 , D_2 , can be modeled using the high-rate approximations as [50]:

$$D_1 \approx \frac{1}{2} \left(\frac{\pi e}{6} \sigma_1 \sigma_2 2^{-2R} + \sigma_2^2 \right), \quad (4.55)$$

$$D_2 \approx \frac{1}{2} \left(\frac{\pi e}{6} \sigma_1 \sigma_2 2^{-2R} + \sigma_1^2 \right). \quad (4.56)$$

Further assuming that each channel is equally likely to break-down, the average side distortion is:

$$\bar{D}_1 \approx \frac{1}{2} (D_1 + D_2) \approx \frac{1}{4} (\sigma_1^2 + \sigma_2^2) + \frac{1}{2} \left(\frac{\pi e}{6} \sigma_1 \sigma_2 2^{-2R} \right). \quad (4.57)$$

We now show that the distortion D_0 can be increased at the cost of \bar{D}_1 using a transform matrix \mathbf{T} . Assuming $a = b = c = 1/\sqrt{2}$ in equation (4.53). Since the variance for both Y_1 and Y_2 is $(\sigma_1^2 + \sigma_2^2)/2$, D_0 can be approximated using the high-rate approximations as [50]:

$$D_0 \approx \frac{\pi e}{6} \left(\frac{\sigma_1^2 + \sigma_2^2}{2} \right) 2^{-2R}, \quad (4.58)$$

which is worse compared to D_0 given in (4.54) by a constant factor Γ :

$$\Gamma = \frac{(\sigma_1^2 + \sigma_2^2)}{2\sigma_1\sigma_2}. \quad (4.59)$$

Now consider the distortion at decoder 1. D_1 is equal to the quantization distortion plus the squared error in estimating Y_2 from Y_1 . Since Y_1, Y_2 are jointly Gaussian random variables, the optimal estimator for Y_2 is given by [50]:

$$E[Y_2 | Y_1 = y_1] = (\sigma_1^2 - \sigma_2^2) (\sigma_1^2 + \sigma_2^2)^{-1} y_1.$$

The estimation error of the above estimator is $2(\sigma_1^2 + \sigma_2^2)^{-1} \sigma_1^2 \sigma_2^2$. Since the total single channel distortion is the average of quantization (of Y_1) and estimation (of Y_2 from Y_1) distortion components, then:

$$\bar{D}_1 \approx \underbrace{\left(\frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \right)}_{\frac{1}{2} E[|Y_2 - \hat{Y}_2|^2]} + \underbrace{\frac{\pi e}{12} \left(\frac{\sigma_1^2 + \sigma_2^2}{2} \right) 2^{-2R}}_{\frac{1}{2} E[|Y_1 - \hat{Y}_1|^2]}. \quad (4.60)$$

Compared to equation (4.57), the first term in (4.60) has reduced by a factor Γ^2 . However, the second term has increased by a factor Γ . Thus we can say that the correlating transform decreased the average side distortion while increasing the central distortion by a related amount. In this example the parameters of the correlating transform (a , b , and c) are fixed. By varying the transform parameters other trade-offs between the side and the central distortion can be obtained.

For a pair of Gaussian random variables X_1 and X_2 , the optimal balanced

correlating transform is derived by Goyal et al. in [50] and is of the form:

$$\mathbf{T}_\alpha = \begin{bmatrix} \alpha & 1/(2\alpha) \\ -\alpha & 1/(2\alpha) \end{bmatrix}, \text{ for } \alpha = \sqrt{\frac{\sigma_2}{2\sigma_1(2^\rho - \sqrt{2^{2\rho} - 1})}}, \quad (4.61)$$

where, ρ denotes the redundancy rate.

The extension of MDTC to produce $M > 2$ descriptions gets computationally involved as M increases. This is due to the fact that one needs to determine $M \times M - 1$ elements of the correlating transform. The optimal solution for $M = 3$ has also been provided in [50]. For $M > 3$, cascading of $M = 2$ optimal transforms can be done to form a general $M \times M$ transform [50]. For $M = 4$, such a cascaded structure is shown in Figure 4.9. For $M = 4$ the cascaded structure reduces the computation of 15 parameters to 3 parameters. However, we notice that the optimality of such a cascaded structure is not proven and the only argument motivating such a design is the less involved computation of the transform parameters [50].

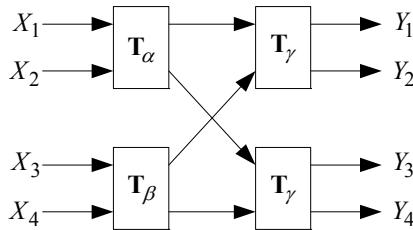


Figure 4.9: Cascaded structure for MDTC of four random variables.

4.4.2 MDC via the Polyphase Transform and Selective Quantization

Introducing a controlled amount of redundancy between descriptions is a main requirement in any practical MDC system. For MDSQ and MDTC, the redundancy is implicitly introduced during the description generation process. Once the descriptions are generated, the redundancy between them cannot be altered. For changing the redundancy, the only possibility that remains is the regeneration of descriptions with the intended amount of redundancy. In many cases, the transmission of descriptions is carried out over channels for which the description loss probability varies over time. For such scenarios, it is beneficial to have an MDC scheme that can explicitly add a controlled amount of redundancy without a complete regeneration of the descriptions.

In the polyphase transform method the redundancy is explicitly introduced using a polyphase transform and selective quantization [56]. The block diagram of the polyphase transform based MDC method is shown in Figure 4.10. An IID source is first split into two sub-sources, one consisting of the odd while the other consisting

of the even source samples. The two odd and even sampled sources are quantized using a single fine uniform quantizer Q_0 to form the first parts of the first and the second descriptions. The second part of the first (respectively second) description is formed by quantizing the even (respectively odd) samples using a coarser uniform quantizer Q_1 [56].

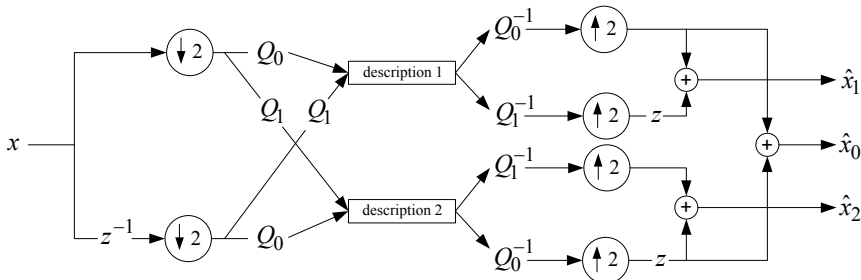


Figure 4.10: The MDC system based on polyphase transform and selective quantization.

For central decoding, the first parts of the two descriptions are used to reconstruct the source with a finer distortion, i.e., the distortion of the inverse quantizer Q_0^{-1} . Let R_0 denote the rate in bps of the quantizer Q_0 , the resulting central distortion for a unit-variance Gaussian source can be approximated, using the high-rate approximations of [45], as

$$D_0 \approx \frac{\pi e}{6} 2^{-2R_0}. \quad (4.62)$$

For side decoding, half of the source samples are reconstructed using Q_0^{-1} while the other half are reconstructed using Q_1^{-1} . In this case the side distortion $D_1 (= D_2)$ can be approximated, using the high-rate approximations of [45], as [56]

$$D_1 = D_2 \approx \frac{1}{2} \frac{\pi e}{6} (2^{-2R_0} + 2^{-2R_1}), \quad (4.63)$$

where R_1 is the rate for the quantizer Q_1 . Since a single description consist of an equal number of samples quantized using Q_0 and Q_1 , the total side rate R is given by

$$R = \frac{1}{2} (R_0 + R_1). \quad (4.64)$$

Since the central distortion D_0 correspond to the single description coding rate of $R^* = R_0$, the redundancy rate ρ in the polyphase method of Figure 4.10 is equal to R_1 . For probability of description loss p_L , the average distortion \bar{D} can be formulated as:

$$\bar{D} = (1 - p_L)^2 D_0 + 2p_L (1 - p_L) D_1 + p_L^2 \sigma^2. \quad (4.65)$$

Substituting the expressions for D_0 and D_1 from equations (4.62)-(4.63) in the

above equation, the optimal redundancy rate ρ^* can be computed by solving $\partial\bar{D}/\partial\rho = 0$, which is given by

$$\rho^* = \frac{1}{2}R + \frac{1}{4}\log_2(p_L). \quad (4.66)$$

From equation (4.66) it is evident that the higher the probability of loss on the channel the more the redundancy should be added in the generated descriptions. From the above equation, the valid positive range of the redundancy is $[0, R/2]$. For 100% channel loss rate, the redundancy can be coded at half the description rate R . Note that, in this sense, the polyphase transform based MDC is superior to MDSQs as it can optimally allocate redundancy for any packet loss rate. Recall that, for MDSQs, the redundancy can be optimally allocated only for $p_L \leq 0.11$ – see equation (4.48).

In practice, despite its simple coding structure, the polyphase transform based MDC [56] provides competitive performance when compared to MDSQs. In fact, in the low redundancy regime, it performs slightly better than MDSQs. Another advantage of the polyphase transform based method is that it can be easily extended to produce more than two source descriptions. In this context, we refer the interested reader to the work of Samarawickrama et al. [88, 89].

4.4.3 Forward Error Correction based MDC

A typical forward error correction (FEC) system that maps k input symbols to N output symbols, where $N > k$, is referred to as an (N, k) FEC code [61]. In systematic FEC codes, the first k symbols are actually the input *information* symbols, while the remaining $N - k$ are the *parity* symbols. Maximum distance separable codes [61], e.g., Reed Solomon codes, provide error-free decoding of the k input symbols for up to $N - k$ channel erasures. In practice, such a code can serve as the basis for a FEC based MDC system, where the reception of any k symbols (descriptions) can yield lossless retrieval of information symbols. However, since the reception of more than k symbols cannot further improve the decoding quality of such FEC codes, a careful formation of descriptions is needed to attain a strict improvement of decoded quality with the number of received descriptions.

An FEC based MDC system was proposed by Puri and Ramchandran in [84]. It was followed from the unequal error protection (UEP) based MDC frameworks [71, 72], wherein a progressive bit-stream is cascaded with Reed-Solomon codes in order to generate an encoding that is progressive in the number of received descriptions, regardless of their identity or the order of their arrival. Starting from a progressive bit-stream, Puri and Ramchandran [84] created N equally important descriptions using (N, k) Reed Solomon codes. For the FEC based MDC method of [84], the

description generation process is shown in Figure 4.11 and Figure 4.12. First, the progressive bit-stream is split into N sections S_1, S_2, \dots, S_N using $N-1$ rate points – see Figure 4.11. Since the bit-stream is progressive, each rate point represents a certain quality level. Each section is then split into equal size parts before FEC coding. In particular, any section S_i , $1 \leq i \leq N$ is split into i equal size parts (see Figure 4.11) and coded with an (N, i) Reed Solomon code. The columns of Figure 4.12 show the N code symbols resulting from the FEC coding of S_i 's, while the rows are the source descriptions sent over independent unreliable channels. Given such a description structure, the i th quality level of the progressive bit-stream is decoded if i descriptions are received at the decoder. Note that, the decoding quality will strictly improve with the reception of more and more descriptions and the complete bit-stream, yielding the highest quality of source reconstruction, can be decoded when all the descriptions are received at the decoder. In [84], an optimization algorithm is also provided to near-optimally select the rates R_j , $0 \leq j < N$ such that the average distortion at the decoder is minimized.

There are several important advantages of the FEC based method over the MDC schemes previously described. In particular, unlike MDSQs, FEC based MDC can be easily generalized to produce an arbitrary number of descriptions. The FEC based MDC method is independent of the employed source coding and directly works with progressively coded bit-streams. This allows for the use of state-of-the-art progressive compression systems, developed for image/video/3-D mesh data, without sacrificing the source compression efficiency. Similar to the polyphase method, the redundancy is explicitly added using FEC codes; in general, redundancy can be controlled by adjusting the redundancy of the FEC code or the number of generated descriptions. Unlike quantization based MDC methods, which directly map a source symbol to two or more channel symbols (i.e., descriptions), FEC based MDC method allows for a modular division of tasks, i.e., the source compression can be designed without considering channel losses and optimizations for the channel loss can be done independent of the source statistics.

The main disadvantage of FEC based MDC system is that it introduces a structural delay in the transmission chain due to iterative soft-decision belief propagation decoding [115] of FEC codes. For many real-time multimedia applications, such as video conferencing, peer-to-peer video streaming [81], and real-time delivery of video [13], [117], the structural delay in decoding can cause a prohibitive drop in QoE.

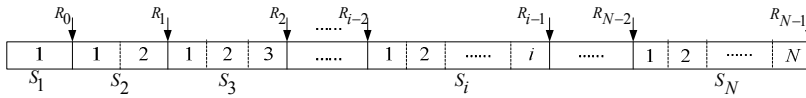


Figure 4.11: Irregular splitting of the progressive bit-stream. The irregular rate points R_i 's define the different quality levels. Any $S_i, i=1,2,..$ is split into i equal size parts, which are included in the first i descriptions in an ascending order, see Figure 4.12.

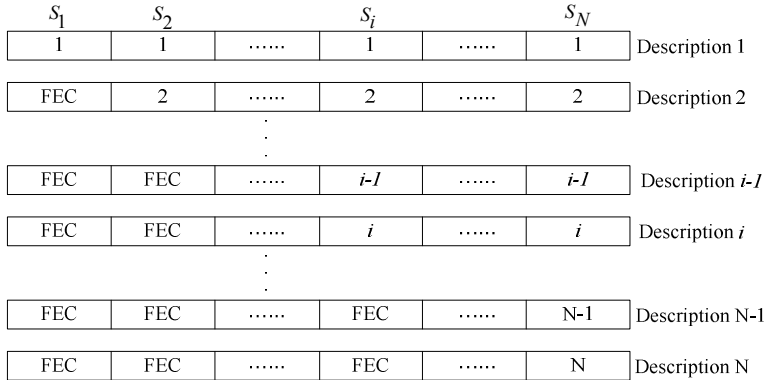


Figure 4.12: FEC based multiple description generation with N quality levels.

4.4.4 Discussion

The different MDC frameworks described above feature different pros and cons. MDSQ suffers from the fine redundancy control problem, since the redundancy can only be controlled by varying the number of diagonals in the IA matrix, which can only be integer. Moreover, due to the complexity of IA, extending MDSQs to generate more than two descriptions is quite a challenging task. However, since the two descriptions MDC case is the most common transmission scenario, this limitation of MDSQs is not particularly harmful from a practical point of view. MDTC seems an attractive approach for MDC, however, in [50] it was shown that, in the medium-to-high redundancy regime, MDSQs significantly outperform MDTC for generic sources. Similar to MDTC, the polyphase transform based MDC provides slightly better performance than MDSQs only for low redundancy values. Finally, the main drawback of FEC-based MDC is the introduced structural delay, due to the interleaver and iterative soft decoding of FEC codes, which can raise many practical concerns if FEC-based MDC is deployed in real-time multimedia transmission applications.

Besides these issues, one major aspect that MDSQ, MDTC and polyphase transform based methods lack is providing scalability of the source representation.

In many cases, it is desirable to have progressively coded descriptions. Using progressively coded descriptions, the transmission rate constraints can be fulfilled on-the-fly. For MDSQ, MDTC and polyphase transform based MDC, only a specified generation of descriptions can ensure the fulfillment of a certain rate constraint. FEC-based MDC enables progressive decoding of the source, however, in the strict sense, the generated descriptions are not progressive.

Progressive encoding and decoding of descriptions can be enabled using a scalable extension of MDSQ, known as *scalable* or *embedded MDSQ*. A scalable MDSQ framework was initially proposed in [51]. For low rates and high redundancy, the approach of [51] delivers superior D-R performance in comparison with a scalable variation of the polyphase MDC system. In the next chapter, we will focus on the design of scalable MDSQs.

4.5 SUMMARY

This chapter described the MDC problem from an information theoretic point of view. In this context, the achievable rate region for the Gaussian source, using the MSE as the distortion metric, is described in detail. Furthermore, MDSQs, which are the pioneering method to perform practical MDC, are explained and their performance bounds for Gaussian sources are presented. Moreover, few alternative frameworks, e.g., MDTC, polyphase transform based MDC and FEC-based MDC, are briefly explained to give a general overview of practical MDC approaches to a reader who is not/less familiar with the domain. In general, the notations and definitions introduced in this chapter will set the basis for the next chapter, which deals with the design of scalable extensions of MDSQs. The design of scalable MDSQ is motivated by the fact that such a framework can enable error-resilience and source scalability in a single coding framework, without introducing the structural delay observed for FEC-based MDC systems.

Chapter 5

SCALABLE MULTIPLE DESCRIPTION SCALAR QUANTIZATION

5.1 INTRODUCTION

Existing MDC schemes, e.g., [50, 56, 113, 114, 121], consider transmission system design for certain pre-known channel conditions, e.g., available bandwidth, packet loss rate, etc. However, in real networks, transmission conditions are often variable, meaning that, for low delay applications, adaptation to varying conditions needs to be performed in real-time. In addition, real-time applications are deployed over networks with a high degree of heterogeneity in terms of available bandwidth, user requests and loss characteristics, thus simultaneously demanding error-resilience and a scalable representation of the source. Scalable MDC (SMDC) combines both scalability and multiple description based error-resilience in a single coding framework. In SMDC each source description is made scalable by employing layered coding, hence facilitating its progressive encoding and decoding. The practical framework for SMDC are provided by the scalable extensions of MDSQs, also referred to as scalable MDSQs [51], [38, 39, 42, 43].

In packet switched networks, packet losses often occur due to over-saturated network links or channel degradations. In such packet loss transmission conditions, one can discriminate between two different situations. In the first situation, the loss statistics are unknown at the encoder. However, in the second situation, an average packet loss probability estimate can be determined at the encoder using a low-rate feedback channel. For the first situation, an SMDC transmission system would optimally suggest a balanced operation, that is, equal number of packets from all descriptions are transmitted. On the other hand, the estimated packet loss rate in the second situation can be exploited to accomplish a more efficient unbalanced transmission. In today's networks both situations can simultaneously occur. Hence, a single system design is needed to get aligned with the aforementioned situations.

In order to meet these requirements, in this chapter, we introduce a generic two-description symmetric scalable MDSQ (SSMDSQ) [93], which yields a perfectly symmetric D-R function, i.e., $D(R_1, R_2) = D(R_2, R_1)$, where the first and the second argument of $D(\bullet, \bullet)$ denote the rate of the 1st and the 2nd source description, respectively. This is in contrast to existing approaches [51], [38, 42] which are characterized by only an approximately symmetric or sometimes even highly asymmetric D-R function. Novel connected-cell and disconnected-cell constructions, which can realize both high and medium-to-low redundancy SSMDSQs, are presented in this chapter. For generalized Gaussian (GG) distributions [78], the proposed SSMDSQs are shown to deliver improved D-R performance versus the existing asymmetric quantizer designs.

In addition, this chapter presents a generalization of the Lloyd-Max algorithm [65, 70] to realize locally-optimal, level-constrained and entropy-constrained SSMDSQs. The proposed algorithm iteratively adjusts quantization thresholds in order to search for the local minimum of a cost function. It is worth pointing out that the proposed optimization methodology is not specific to SSMDSQs and can be utilized to optimize any scalable MDSQ. To the best of the author's knowledge, this is the first work in the related literature that deals with the optimization of scalable MDSQs.

What is more, based on the designed SSMDSQs, a packet-based SMDC framework is established which can match the available channel rate, while guaranteeing a fixed average distortion at the decoder. In this framework, a balanced transmission of descriptions is performed when no knowledge of the channel's packet loss rate is available at the encoder. Conversely, when the encoder is aware of the channel statistics, a greedy packet scheduling scheme is formulated so as to carry out an unbalanced transmission. In this context, the symmetry of the SSMDSQ-based SMDC's D-R function reduces the complexity of determining optimum packet scheduling compared to asymmetric contemporary SMDC systems. Experimental evaluations of the transmission of GG sources over packet loss channels reveal that, within the considered SMDC framework, the proposed quantizer constructions and the optimization strategy contribute significant average SNR gains compared to existing designs. Similar conclusions are also obtained when the proposed SSMDSQ are employed for wavelet-based scalable coding of photographic images.

5.2 RELATED WORK

A scalable MDSQ based SMDC framework was initially proposed in [51], also referred in short as multiple description uniform scalar quantizer (MDUSQ),

wherein the concept of embedded index assignment (EIA) was first introduced to achieve separate and combined scalability of two source descriptions. For low rates and high redundancy, the SMDC system of [51] demonstrates superior D-R performance in comparison to a scalable variation of the polyphase MDC system of [56]. Unfortunately, the EIA strategy of [51] permits limited control over the redundancy between descriptions. In order to produce an arbitrary number of scalable descriptions as well as to permit an enhanced control over the redundancy between the quantization levels, embedded MDSQs (EMDSQs) were introduced in [38, 39, 42, 43]. Preliminary EMDSQ instantiations [38, 42] were designed using connected partition cells to yield high redundancy descriptions. In [43], these instantiations were combined with a minimal-redundancy EIA to achieve low-to-medium overall redundancy. Notice that the main design objective of the EMDSQs of [38, 39, 42, 43] was the use of a double dead-zone uniform central quantizer at all quantization levels. As explained earlier, such a quantization strategy is also termed as SAQ [100] and is acknowledged to be near-optimal [108]. This attribute significantly improved the D-R performance of EMDSQs over competing designs.

Anchored in the designed EMDSQs [42], erasure-resilient scalable image coding frameworks were proposed in [40], [41]. The proposed frameworks outperform the equivalent MDUSQ [51] based image coding system. Additionally, for scalable video streaming, an EMDSQ based multicast error-resilient video coding architecture was proposed in [117]. Compared to a data-partitioning-based video codec [9], the former accounts for an improved compression versus error-resilience trade-off for video streaming applications over best-effort networks.

5.3 SCALABLE MULTIPLE DESCRIPTION QUANTIZER

As described in Chapter 4, a fixed-rate MDSQ quantizes each source sample to two or more quantization indices which individually yield a coarse, and jointly produce a fine reconstruction of the source. A scalable MDSQ is a special type of MDSQ, which additionally allows for scalable decoding of produced quantization indices. An M -description scalable MDSQ can be fully described by a set of M embedded side quantizers $\{\mathbf{Q}_m\}_{m=1}^M$ each having $P+1$ embedded levels $\{\mathcal{Q}_p^m\}_{p=0}^{P-P}$, where $p=0$ denotes the finest and $p=P$ denotes the coarsest level, respectively. The quantizer partition of \mathcal{Q}_p^m is denoted as S_p^m . For any side quantizer \mathbf{Q}_m , the partition cells of level p are embedded in the ones of level $p+1$, i.e., $S_p^m \subseteq S_{p+1}^m$, where $0 \leq p < P$. We aim to quantize a zero-mean, IID ergodic source $\{X_n, n \in \mathbb{Z}\}$ with a symmetric PDF $f(x)$, $-x_0 \leq x \leq x_0$. At the encoder, M source descriptions are produced using the M side quantizers. In particular, for the m th side quantizer,

a given source sample x is first mapped to an index s_p using the coarsest level partition S_p^m . At any finer level $0 \leq p < P$, the source sample is refined to a smaller sub-cell as defined by the refinement symbol s_p^* . A sequence of quantization indices, denoted by s_p, s_{p-1}, \dots, s_1 , is progressively coded using a FLC, e.g., the NBC, or a VLC, e.g., arithmetic coding (Section 2.3.6), thereby producing the m th source description. At the decoder, assuming that all M descriptions up to a certain level p are received, a *central quantization cell* $C_{p, s_p^*} = \bigcap_{m=1}^M S_{p, s_p^*}^m$, $s_p^* = s_p, s_{p-1}, \dots, s_1$, is determined for each source sample. Then, the source sample reconstruction is given by the center-of-mass (centroid) of the derived cell. In general, all M descriptions need not be received at the same level, yet one can still determine a joint partition cell for possibly different quantization levels $\mathbf{p} = [p_1, p_2, \dots, p_M]$. In this case, the central quantizer $Q_{\mathbf{p}}$ is defined by the corresponding central partition $C_{\mathbf{p}}$, as follows:

$$Q_{\mathbf{p}} \equiv Q_{[p_1, p_2, \dots, p_M]} : C_{\mathbf{p}} = \bigcap_{m=1}^M S_{p_m}^m, p_m \in \{0, 1, \dots, P+1\}, \quad (5.1)$$

with the convention that Q_{p+1}^m consist of a single cell, which spans the entire source range $[-x_0, x_0]$. Notice that the central and the side quantizers at any level p are given by $Q_{[p_1=p, p_2=p, \dots, p_M=p]} \equiv Q_p$ and $Q_{[p_1=p+1, \dots, p_m=p, \dots, p_M=p+1]} \equiv Q_p^m$, correspondingly.

5.3.1 Embedded Index Assignment

For simplicity, we confine our discussion to the two-description scalable MDSQ case. Let us consider an IA matrix \mathbf{M} , which assigns quantization indices to a side quantizer's cell partitions. Starting from single-cell side quantizers, i.e., Q_{P+1}^m , $m \in \{1, 2\}$, \mathbf{M} is recursively split $(P+1)$ times along each dimension to create an EIA of $P+1$ levels. For instance, for a particular level combination $\mathbf{p} = [p_1, p_2]$, \mathbf{M} can be expressed as a block matrix of the form:

$$\mathbf{M} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,J_{p_1}} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \cdots & \mathbf{B}_{2,J_{p_1}} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{B}_{J_{p_2},1} & \mathbf{B}_{J_{p_2},2} & \cdots & \mathbf{B}_{J_{p_2},J_{p_1}} \end{bmatrix}, \quad (5.2)$$

where, a non-empty block of index entries, \mathbf{B}_{j_2, j_1} , defines a specific partition cell of the quantizer $Q_{[p_1, p_2]}$. Note that, depending on the organization of the index entries in \mathbf{M} , certain blocks may be completely or partially empty. We also notice that $\mathbf{B}_{j_1} = \bigcup_{j_2} \mathbf{B}_{j_2, j_1}$ and $\mathbf{B}_{j_2} = \bigcup_{j_1} \mathbf{B}_{j_2, j_1}$ define cells of side quantization levels $Q_{p_1}^1$, $Q_{p_2}^2$. In general, one observes that $\mathbf{B}_{j_1} = \mathbf{B}_{1, j_1}$ and $\mathbf{B}_{j_2} = \mathbf{B}_{j_2, 1}$ at levels $\mathbf{p}_1 = [p_1, P+1]$ and $\mathbf{p}_2 = [P+1, p_2]$, respectively. Hence, using the above notation, side cells \mathbf{B}_{j_2} , \mathbf{B}_{j_1} can be referred to as cells at levels $\mathbf{p}_2 = [P+1, p_2]$ and

$\mathbf{p} = [p_1, P+1]$, respectively. In order to realize the next level, i.e., $\mathbf{p} = [p_1 - 1, p_2 - 1]$, a further splitting of each $\mathbf{B}_{j_2 j_1}$ of equation (5.2), along each dimension is performed, and so on. Moreover, notice that at level $\mathbf{p} = [0, 0]$, each block \mathbf{B} contains at most a single index entry.

For $M > 2$, any \mathbf{B} is a hypercube with M dimensions. Similarly, for $M > 2$, cells for the m th side can be referenced as cells at levels $\mathbf{p}_m = [p_1 = P+1, \dots, p_m = p, \dots, p_M = P+1]$.

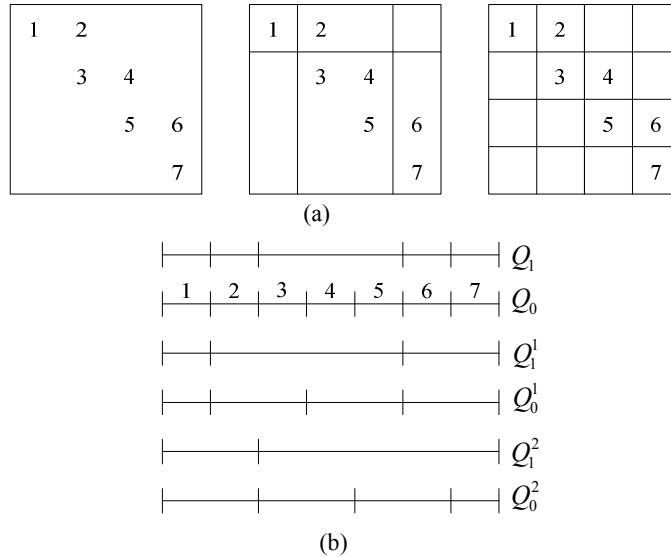


Figure 5.1: Example of a scalable MDSQ created using an EIA matrix. (a) IA splitting (b) The corresponding side and central quantizer partitions at different quantization levels.

Figure 5.1 depicts an example of a two-level EIA. The original matrix depicted in Figure 5.1(a)-left is split into six blocks to create the level $p = 1$ – see Figure 5.1(a)-middle. The resulting blocks are then further split to create the finest level $p = 0$. Assuming a uniform Q_0 , partitions of the side and the central quantizers at different levels are shown in Figure 5.1(b). The example given in Figure 5.1 belongs to the category of connected-cell scalable MDSQs, wherein the side and the central quantization cells at any level consist of a single continuous interval. Different categories of scalable MDSQs, namely connected- and disconnected-cell scalable MDSQs, will be discussed in detail in the later part of this chapter.

5.3.2 Description Packetization

The descriptions' bit-streams are generally transmitted using packets of fixed size. At first, the bit-planes resulting from the quantization process per description are coded with NBC in case of FLC or with adaptive arithmetic coding in case of VLC.

Per description, the coded bit-planes are put together in the order of their significance to create a single scalable bit-stream. Each such bit-stream per description is thereafter split into a number of fixed size packets. This way each packet per description yields a progressive increment in the description rate, say κ bps. In other words, if we receive N_m packets from the m th description, $m=1,2$, then we can decode the description at $R_m = \kappa N_m$ bps.

From here onwards, we assume that thresholds for Q_0 are always defined as an even function of x , allowing to interchangeably refer to a certain scalable MDSQ or its EIA.

5.3.3 Definitions

Let $\mathbf{p}=[p_1, p_2, \dots, p_M]$ denote the side quantization levels up to which M side descriptions are received at the decoder. If $\hat{X}_{\mathbf{p}}^0$ denotes the joint reconstruction random variable then the MSE, which is considered as the distortion metric, can be formulated as:

$$D(\mathbf{p}) = E\left[\left(X - \hat{X}_{\mathbf{p}}^0\right)^2\right] = \sum_{i=1}^{|C_{\mathbf{p}}|} \int_{C_{\mathbf{p},i}} (x - \hat{x}_i)^2 f(x) dx, \quad (5.3)$$

where $f(x)$ is the input PDF, $C_{\mathbf{p}}$ denotes the induced central partition, $C_{\mathbf{p},i}$ is the quantization cell corresponding to the i th central cell, $E[\cdot]$ is the expectation operator, and $|C_{\mathbf{p}}|$ denotes the cardinality, i.e., the number of cells in $C_{\mathbf{p}}$. For any level p , the central distortion $D_0(p)$ and the M sides distortions $D_1(p), D_2(p), \dots, D_M(p)$ are defined as:

$$\begin{aligned} D_0(p) &= D(\mathbf{p}=[p, p, \dots, p]) = E\left[\left(X - \hat{X}_p^0\right)^2\right], \\ D_m(p) &= D(\mathbf{p}_m) = E\left[\left(X - \hat{X}_p^m\right)^2\right], \quad \text{for } 1 \leq m \leq M, \end{aligned} \quad (5.4)$$

where \hat{X}_p^0, \hat{X}_p^m denote random variables defined by the central and the m th side quantizer's codebook, respectively. To compute the entropy of each side quantizer, we define a random variable $P_{S_p^m}$ using the probability mass function of the p th level partition of the m th side quantizer, i.e., $\Pr(P_{S_p^m} = i) = \int_{S_{p,i}^m} f(x) dx = q_{S_{p,i}^m}$, where $S_{p,i}^m$ is the i th cell of S_p^m . The entropy $H_m(p)$ for the m th side quantizer is then expressed as:

$$H_m(p) = - \sum_{i=1}^{|S_p^m|} q_{S_{p,i}^m} \log_2 \left(q_{S_{p,i}^m} \right). \quad (5.5)$$

Notice that scalable MDSQ is considered to be balanced if for any level p , the distortions and the entropies corresponding to all the M sides are equal, that is, $D_m(p) = D_{m'}(p)$ and $H_m(p) = H_{m'}(p)$, $\forall (m, m')$.

5.4 SYMMETRIC SCALABLE MDSQ

In scalable MDSQs, descriptions are progressively transmitted. Although the generated descriptions are balanced, an *unbalanced* (i.e. $R_1 \neq R_2$) decoding situation occurs when an unequal number of quantization levels between descriptions are received. This forces the D-R function to follow a surface rather than separate side and central curves as in the case of non-scalable MDSQ, expressed by equations (4.40)-(4.41). Figure 5.2 depicts a schematic representation of a scalable MDSQ's D-R surface $D(R_1, R_2)$ in which the solid line curves AB , AC and AD denote the two sides and the central distortion functions, respectively.

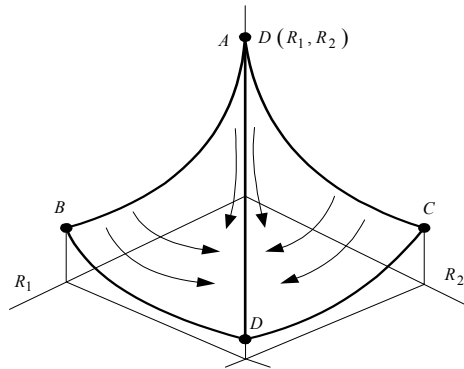


Figure 5.2: Illustration of the D-R surface of a two-description scalable MDSQ.

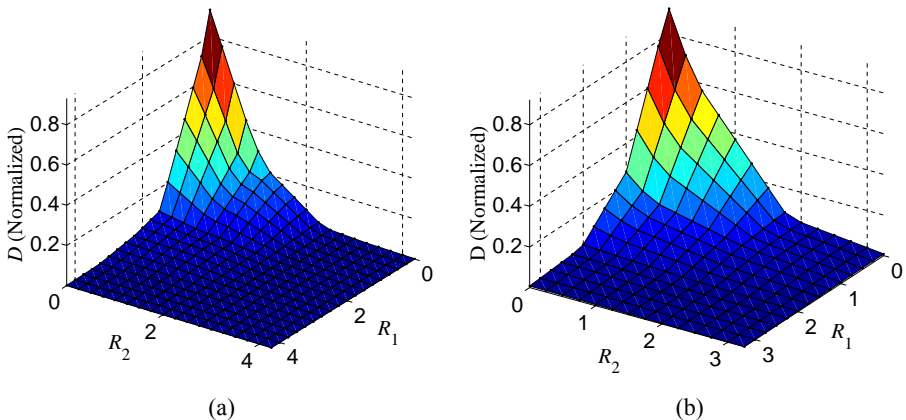


Figure 5.3: Experimental D-R surface of EMDSQ instantiations. (a) One and half diagonal EIA [38] (b) Two diagonal EIA [42].

Figure 5.3 plots the experimental D-R surface for two high-redundancy EMDSQs [38, 42]. We observe that the D-R surfaces are not perfectly symmetric, in the sense that $D(R_1, R_2)$ is generally not equal to $D(R_2, R_1)$. The main reason for such an asymmetric D-R behavior is due to the fact that different permutations of a certain

combination of side quantization levels produce unequal distortions, i.e., $D(p_1, p_2) \neq D(p_2, p_1)$. Such an asymmetry of the D-R function is observed in all proposed instantiations of scalable MDSQs [51], [38, 39, 42, 43].

In this section, we determine the sufficient conditions for realizing a symmetric scalable MDSQ (SSMDSQ), which, opposed to an ordinary balanced scalable MDSQ, always guarantees an equal distortion for both permutations of a given combination of quantization levels received from both the side descriptions, i.e., $D(p_1, p_2) = D(p_2, p_1)$.

Symmetric Cells: Let the index entries of \mathbf{M} be $\{1, 2, \dots, \Omega\}$. The partition cells \mathbf{B} and \mathbf{B}' are defined as being *symmetric* to each other if for any index ω in \mathbf{B} there is an index $\Omega - \omega + 1$ in \mathbf{B}' . The symmetry between \mathbf{B} and \mathbf{B}' is denoted as $\mathbf{B} \leftrightarrow \mathbf{B}'$.

Symmetric EIA: An EIA is defined as being *symmetric* if each partition cell for any combination $\mathbf{p} = [p_1, p_2]$ has a unique symmetric partition cell for $\mathbf{p}' = [p_2, p_1]$.

In Figure 5.4 (a), a symmetric EIA is formed by employing a uniform four-level splitting, of a certain arrangement, of the entries in \mathbf{M} . The symmetric cell pairs at any splitting level are given below:

$$\begin{aligned} \text{level 1: } & \{1, 2, 3, 4\} \leftrightarrow \{9, 8, 7, 6\}; \{5, 6, 7, 8, 9\} \leftrightarrow \{5, 4, 3, 2, 1\}, \\ \text{level 2: } & \{1, 2, 3, 4\} \leftrightarrow \{9, 8, 7, 6\}; \{5\} \leftrightarrow \{5\}, \\ \text{level 3: } & \{1, 3\} \leftrightarrow \{9, 7\}; \{2, 4\} \leftrightarrow \{8, 6\}; \{5\} \leftrightarrow \{5\}; \{6, 7\} \leftrightarrow \{4, 3\}; \{8, 9\} \leftrightarrow \{2, 1\}, \\ \text{level 4: } & \{1\} \leftrightarrow \{9\}; \{2\} \leftrightarrow \{8\}; \{3\} \leftrightarrow \{7\}; \{4\} \leftrightarrow \{6\}; \{5\} \leftrightarrow \{5\}. \end{aligned}$$

An example of a symmetric cell pair in Q_0 corresponding to the IA in Figure 5.4(a) is given in Figure 5.4(b).

Remark 1: In the example of Figure 5.4(a), the chosen IA matrix \mathbf{M} is actually the fixed-rate (non-scalable) modified linear (ML) IA of [113]. In general, all fixed-rate IAs proposed in the literature can be used to create scalable MDSQs by employing bit-slicing of quantization indices. However, one may notice that, in contrast to fixed-rate IAs, where the spread is only minimized along the columns and the rows of the IA matrix [113], an explicit design of EIA must consider the spread minimization along rectangles or squares cells, similar to the ones shown in Figure 5.4(a).

Theorem [93]: For a symmetric zero-mean source PDF $f(x)$, a balanced SSMDSQ is always realized if the EIA is symmetric and the thresholds of Q_0 are defined using an even¹⁰ function of x .

¹⁰ Any real function $g(x)$ is referred to as an even function if for any x , $g(x) = g(-x)$.

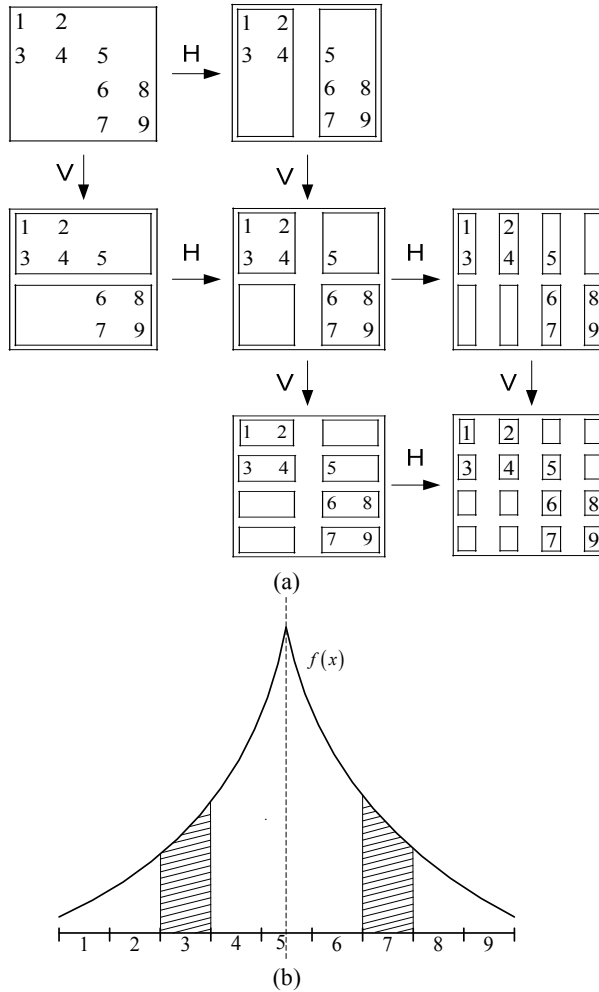


Figure 5.4: (a) An example of a symmetric EIA. “V” denotes vertical split, while “H” denotes the horizontal split of the IA matrix. (b) Finest central quantizer Q_0 , symmetric cell pair $\{3\} \leftrightarrow \{7\}$ is highlighted using dashed area under the PDF.

Proof: If the EIA is symmetric and the thresholds of Q_0 are defined using an even function of x , the particular structure of the index entries within a symmetric cell pair ensures that their thresholds have equal absolute values – see the example in Figure 5.4(b). For a symmetric source PDF, this results in equal distortions of symmetric cells. For a symmetric EIA, wherein each cell of a combination $\mathbf{p} = [p_1, p_2]$ has a unique symmetric cell in $\mathbf{p}' = [p_2, p_1]$, the total distortion $D(\mathbf{p})$, which is the sum of distortion contributions from each cell, is always equal to $D(\mathbf{p}')$. This confirms the symmetry of the distortion of SSMDSQs. Since the above argument of the same equal absolute value thresholds is also true for side symmetric cells, i.e., the cells of permutations $[p, P+1]$, $[P+1, p]$, $0 \leq p \leq P$, the cells in a

side symmetric pair contribute the same distortion and entropy, namely, $D([p, P+1]) = D([P+1, p])$ and $H([p, P+1]) = H([P+1, p])$. In other words, $D_1(p) = D_2(p)$ and $H_1(p) = H_2(p)$, are true for any level p , thereby proving that an SSMDSQ is always perfectly balanced. ■

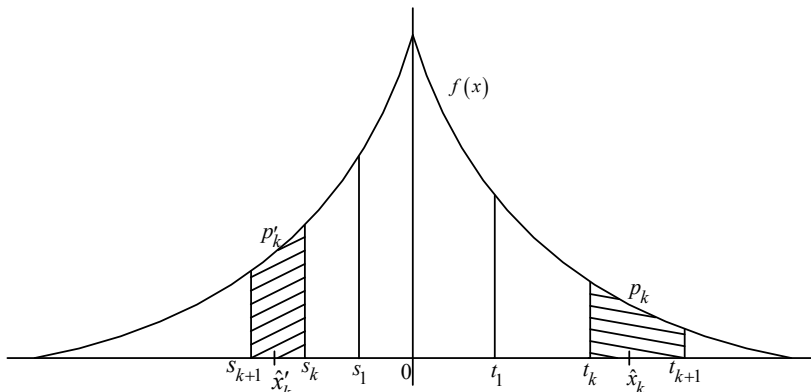


Figure 5.5: Illustration figure for SSMDSQ construction when thresholds of Q_0 are not an even function – see Remark 2.

Remark 2: Assuming EIA to be symmetric, one can construct symmetric quantizers for zero-mean symmetric PDFs for which the thresholds of Q_0 are *not* an even function of x . The proof is constructive. With the notations of Figure 5.5, one can impose that the probability of observing index p_k is equal to the probability of observing p'_k , i.e., the dashed areas are equal. With this condition, one proceeds recursively, starting from given thresholds s_k, t_k, t_{k+1} to identify s_{k+1} . The result is that the entropies on both sides of the PDF are equal. Subsequently, by imposing equal distortions in the dashed cells, one determines the reconstruction point \hat{x}'_k in cell p'_k for a given reconstruction point \hat{x}_k (e.g., the centroid) in cell p_k . Proceeding recursively for each k ensures that the distortions on both sides of the PDF are equal. Applying this procedure on both the central and side quantizers at all quantization levels leads to balanced SSMDSQs, with symmetric EIA, for which the thresholds of Q_0 are *not* necessarily an even function of x . However, in our work we do not follow the design methodology sketched above. First, nothing can be said about global optimality, and the thresholds depend on the input PDF. Furthermore, the thresholds and reconstruction points need to be determined recursively for each cell in order to ensure symmetry. This corresponds to a computationally expensive recursive placement of quantizers' thresholds which is unattractive from a practical point of view.

In Sections 5.4.1 and 5.4.2, we highlight the drawbacks of previously proposed *connected-cell* as well as *disconnected-cell* scalable MDSQs, respectively, and

propose new SSMSDQ constructions to overcome these limitations. From here onwards, we assume that the thresholds for Q_0 are always defined as an even function of x , allowing to interchangeably refer to a certain scalable MDSQ or its EIA.

5.4.1 Connected-cell Scalable MDSQ

The most attractive feature of connected-cell scalable MDSQs is that their central and side quantizers have connected partition cells at any level. Although largely reducing the implementation complexity of such quantizers, this characteristic may lead to a sub-optimal D-R performance – see Theorem 5 in [77]. An additional drawback of maintaining connected partition cells is that the resulting quantizers can only generate high-redundancy side descriptions.

The EMDSQs of [38, 42] are representative examples of connected-cell scalable MDSQs. In Figure 5.6(a), the two level EIA of [42] is depicted. Notice that, since connected-cell side and central partition cells are realized at each level, entries along a row (or a column) are consecutive. This implies that $\Omega \leq |S_0^1| + |S_0^2| - 1$, where $|S_0^1| = |S_0^2|$. Upon a close look at the EIA of [42] – see Figure 5.6(a), one observes that its index arrangement is highly asymmetric. Table 5.1 reports the distortion performance of the resulting EMDSQ [42]. We note that the side distortion values in Table 5.1 deviate by a large magnitude from their mean. In practice, the same behavior can be also observed for side entropies. This unbalanced behavior is due to the irregular splitting of \mathbf{M} , as shown in Figure 5.6(a).

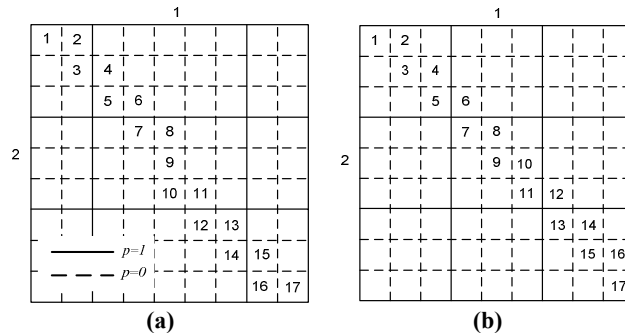


Figure 5.6: EIAs of connected-cell scalable MDSQs. (a) The asymmetric EIA of [42], (b) The equivalent symmetric EIA.

A straightforward way to produce a connected-cell symmetric scalable MDSQ is by sequentially populating the IA matrix, such that the sufficient requirements of the symmetric EIA definition are fulfilled. Namely, starting from the top-left corner, let an entry ω be placed at position (i, j) in the IA matrix. Then, symmetry demands that an entry $\Omega - \omega + 1$ is placed at position $(|S_0^1| - j + 1, |S_0^2| - i + 1)$. In this way a

symmetric IA matrix is assembled and uniform splitting is carried out to derive a symmetric EIA, as in Figure 5.6(b). The latter has the same total number of index entries with the one of Figure 5.6(a), therefore the two EIAs are equivalent. The experimental validation of the designed SSMDSQ – see Table 5.1, demonstrates that for a given number of received bit-planes the symmetric EIA exhibits lower central and mean side distortions compared to the asymmetric EIA. For both quantizers, the finest central quantizers Q_0 are the same, thus resulting in exactly the same central distortion when all levels from both sides are used for source reconstruction.

Table 5.1: Distortion performance of the scalable MDSQs of Figure 5.6 for a unit-variance Gaussian source. Mean denotes the mean of two side distortion readings, i.e., $0.5 * (D_1 + D_2)$. DDZ=double dead-zone. Div= $|D_{1,2} - Mean|/Mean$.

| | EIA OF [42] + DDZ UNIFORM QUANTIZER | | | | | SYMMETRIC EIA + DDZ UNIFORM QUANTIZER | | | |
|---------|--|----------|----------|--------|-----|--|----------|----------|---------------|
| | $D_0(p)$ | $D_1(p)$ | $D_2(p)$ | Mean | Div | $D_0(p)$ | $D_1(p)$ | $D_2(p)$ | Mean |
| $p = 2$ | 1.0028 | 1.0028 | 1.0028 | 1.0028 | 0 | 1.0028 | 1.0028 | 1.0028 | 1.0028 |
| $p = 1$ | 0.4005 | 0.9338 | 0.4118 | 0.6728 | 38% | 0.3931 | 0.5209 | 0.5209 | 0.5209 |
| $p = 0$ | 0.0328 | 0.1710 | 0.0620 | 0.1165 | 46% | 0.0328 | 0.0987 | 0.0987 | 0.0987 |

5.4.2 Disconnected-cell Scalable MDSQ

Quantizers belonging to this family are composed of disconnected partition cells. This trait increases their implementation complexity, however, contrary to connected-cell quantizers, it enables them to generate medium-to-low redundancy source descriptions [43]. Generally, their side quantizers do not employ a dead-zone cell and their reconstruction value is given by the centroid. The symmetric ML IA of [113] and the asymmetric EIAs of [51], [43] are the most representative examples of disconnected-cell scalable MDSQs.

Table 5.2 depicts a D-R comparison of the ML(8,2) IA of [113] against the equivalent EIA of [51] for a zero-mean, unit-variance Gaussian source. The granular distortion [10, 110], which is the product of the side and the central distortion (see Section 4.2.3), is reported for several side rates. For a fair comparison between symmetric and asymmetric EIAs, the overall side distortion is taken as the mean of distortions of the two sides. Thus the reported granular distortion is:

$$D_g(R) = D_0(R) \left(\frac{D_1(R) + D_2(R)}{2} \right). \quad (5.6)$$

From Table 5.2 we notice that, in contrast to the EIA of [51], the symmetric ML(8,2) IA always produces perfectly balanced side descriptions. The second row

of Table 5.2 includes the *percentage deviation from the mean (%Div)* of $D_1(R)$ and $D_2(R)$, for the EIA of [51]. Interestingly, a large deviation between the two side distortions is observed, demonstrating that the EIA of [51] realizes highly unbalanced descriptions. Furthermore, Table 5.2 clearly shows that the EIA of [51] outperforms the ML IA of [113] in terms of granular distortion. The obtained gains are attributed to the connected central partition EIA of [51] and to the fact that the side descriptions are largely unbalanced, yielding significantly lower central distortion values. However, recall that, in case the encoder is unaware of the channel conditions, an SMDC system has to employ a balanced transmission. This cannot be achieved using asymmetric EIAs due to their unbalanced nature.

Table 5.2: Granular distortion of three disconnected-cell scalable MDSQs for a unit-variance Gaussian source (exact variance 1.0028). Q_0 is considered uniform. Each quantizer consists of three embedded levels (i.e., $P = 2$). For the asymmetric EIA of [51], %Div of the side distortion from their mean value is given in parenthesis.

| Rate (bps) | 0.25 | 0.50 | 0.75 | 1.00 | 1.25 | 1.50 | 1.75 | 2.00 | 2.25 |
|-------------|---------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| ML IA [113] | 0.753 | 0.532 | 0.347 | 0.192 | 0.172 | 0.153 | 0.135 | 0.117 | 0.084 |
| EIA [51] | 0.710 (7%) | 0.456 (15%) | 0.247 (26%) | 0.078 (41%) | 0.065 (43%) | 0.054 (46%) | 0.044 (49%) | 0.034 (52%) | 0.026 (54%) |
| PROPOSED | 0.666 | 0.394 | 0.191 | 0.0799 | 0.0598 | 0.0420 | 0.0257 | 0.0180 | 0.0124 |

5.4.2.1 Proposed Symmetric EIA

To resolve this drawback, we propose a novel symmetric disconnected-cell EIA strategy, which combines the improved D-R advantage of [51] with the balanced side distortion characteristics of the ML IA of [113].

In particular, consider the design of an EIA for $|S_0^m| = 2^{p+1}$, $m \in \{1, 2\}$, with side cells using a certain number of diagonals $d = 2k + 1$, where $k = 0, 1, \dots, (2^{p+1} - 1)$. The proposed EIA starts by initializing the $p = 1$ level IA using the symmetric ML IA with parameters $(2^p, k - 1)$ – see [113]. Each index of the resulting IA is subsequently split into 2×2 sub-indices to realize the finest ($p = 0$) level. Depending on whether k is even or odd, predefined index scans, at level $p = 0$, are carefully performed to conform to the definition of a symmetric EIA.

Five index scans are employed which are shown in Figure 5.7. The complete algorithm is given in the form of pseudo code in the Appendix at the end of the chapter. In general, one can split each index entry of the ML IA at $p = 1$ into a block of $l \times l$ sub-entries, where $l = 2, 3, 4, \dots$. To yield a symmetric final IA, when deriving the indices of level $p = 0$, the index scanning of the i th and the

corresponding $(N_1 - i + 1)$ th block should be a *reverse mirror* of each other, where N_1 denotes the number of entries at level $p=1$. Note that, although any $l=2,3,4,\dots$ can be used to realize a symmetric EIA, in our realization we choose $l=2$, for simplicity.

In detail, for $l=2$, the reversed mirrored Z and V scanning patterns are applied to all entries belonging to diagonals $2,\dots,2k$ – when k is odd – and to all entries belonging to diagonals $1,\dots,2k+1$ – when k is even. In the former case, the lower diagonal (LD) and the upper diagonal (UD) scans are applied to all elements of the diagonals indexed by 1 and $2k+1$, respectively. Notice that for these elements, one can still simply use the Z and V scans instead of the LD and UD scans; however, this would notably increase the total number of entries of the proposed EIA compared to those of the ML IA [113] or the EIA of [51], for a given odd value of k . Finally, the L scan, which is the reverse mirror of itself, is only used to split the specific $(N_1 + 1)/2$ index of the ML IA at $p=1$ level.

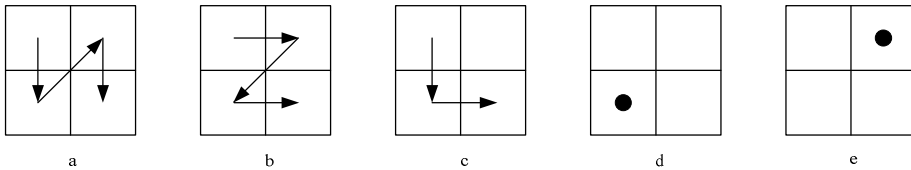


Figure 5.7: The index scans which are employed in order to construct a symmetric EIA. (a) V scan, (b) Z scan, (c) L scan, (d) UD scan, (e) LD scan.

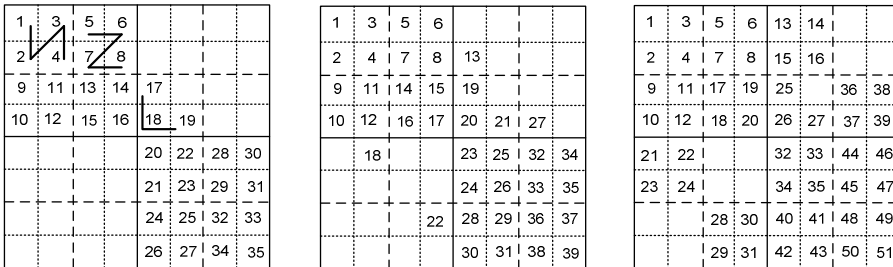
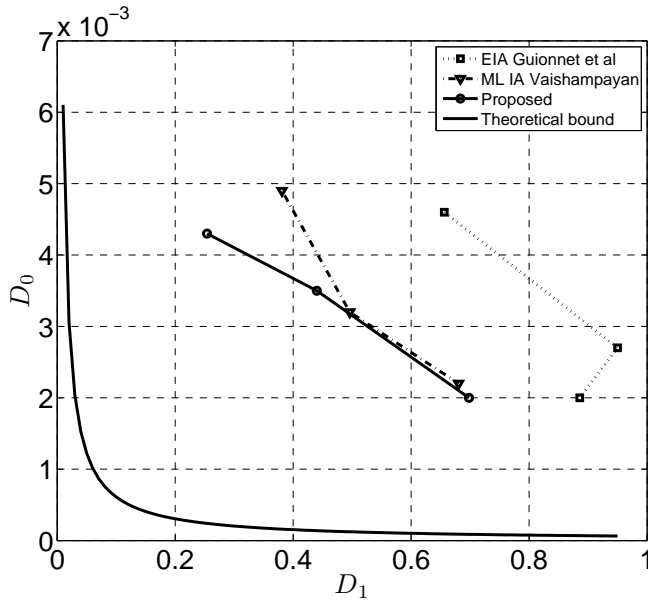
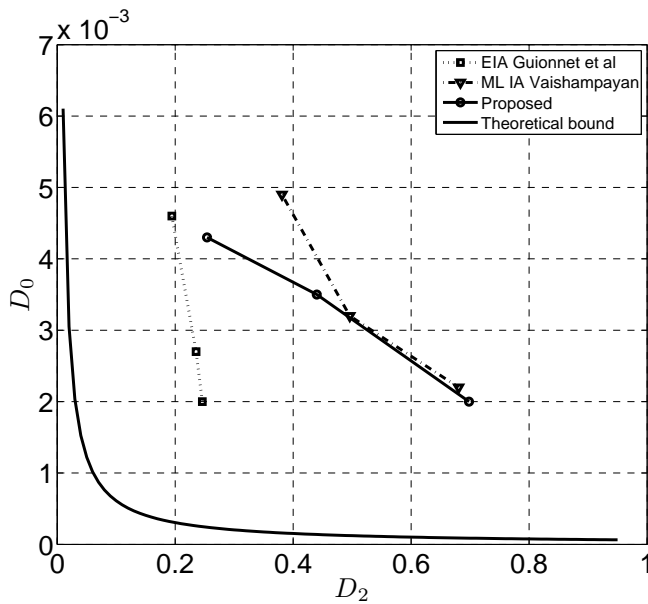


Figure 5.8: Constructive design of a disconnected-cell symmetric EIA with $P=2$: (left) $d=5$, (middle) $d=7$, (right) $d=9$. Embedded levels can be differentiated using different line styles (solid line: $p=2$), (dashed line: $p=1$) and (dotted line: $p=0$).

Figure 5.8 depicts the resulting symmetric EIAs for different number of diagonals. We notice that, unlike the ML IA of [113] and the EIA of [51], the proposed EIA strategy may include partially filled diagonals. However, for a given P and d , the total number of index entries of the proposed and the existing IAs would roughly be the same. The last row of Table 5.2 reports the granular distortion $D_g(R)$ of the proposed symmetric EIA based SSMSQ. Clearly, the proposed EIA



(a)



(b)

Figure 5.9: Central versus side distortion comparison for an IID Gaussian source with unit-variance at $R_1 = R_2 = 3.0$. Three points are plotted for a given EIA family, each corresponding to a certain number of diagonals in the IA matrix. The left-most points correspond to $d = 5$, the middle points to $d = 7$, while the right-most points correspond to $d = 9$.

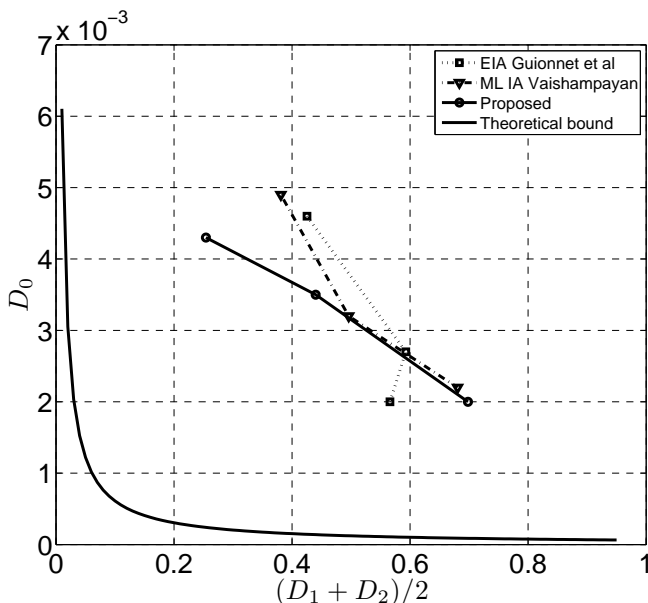


Figure 5.10: Central versus mean side distortion comparison for an IID Gaussian source with unit-variance at $R_1 = R_2 = 3.0$.

strategy yields the lowest $D_g(R)$ values, while still maintaining perfectly balanced side distortions.

Moreover, Figure 5.9, Figure 5.10 plot the side versus the central distortion for the three families of EIAs with respect to the theoretical bound given by $D_0 D_1 \geq (1/4)2^{-4R}$ [10]. Guionnet et al. denotes the EIA of [51]. Different points for a given family are obtained by varying the number of diagonals in the IA matrix.

As a general multiple description rule, at a given side rate, the side distortion should increase when lowering the redundancy (i.e., by increasing the number of diagonals) – see the curves in Figure 5.9 and Figure 5.10 corresponding to the ML IA [113] and the proposed symmetric EIA. For the asymmetric EIA of [51], although the second side description obeys this rule – see Figure 5.9(b), the same is not true for the first side description, as shown in Figure 5.9(a). In particular, for the first side of the EIA of [51], we notice that $D_1(d=7) > D_1(d=9)$. This causes the mean side distortion for $d=9$ to be lower than the mean side distortion for $d=7$ – see Figure 5.10. Such a behavior is specifically explained by the asymmetric nature of EIA of [51]. In contrast to [51], the proposed EIA strategy ensures a smooth decay of side distortion, similar to ML IA. Moreover, Figure 5.9 and Figure 5.10 corroborate that the proposed EIA strategy accounts for a better side versus central distortion trade-off for a certain achievable redundancy point.

5.5 QUANTIZER OPTIMIZATIONS

In this section, a novel algorithm to optimize an SSMDSQ, driven by the PDF of the source to be coded, is presented. Our approach is generic since it can optimize different types of SSMDSQs with different D-R characteristics. The proposed optimization algorithm is a novel extension of the Lloyd-Max algorithm [65, 70] to the multiple description embedded setting, wherein both level-constrained and entropy-constrained optimization can be performed using the same algorithm.

Similar to the Lloyd-Max algorithm, the proposed optimization builds on the fact that an encoder-decoder pair (\mathbf{f}, \mathbf{g}) can be iteratively optimized by keeping one component fixed while optimizing the other. By doing so, a non-increasing sequence of cost function values is obtained. Therefore, the derived encoder-decoder pair is locally optimal and a good initialization condition is necessary to realize a good final solution.

Considering a two-description SSMDSQ, we define the aggregate side and aggregate central distortions as a weighted sum of per-level quantities, given by (5.4), as below:

$$\tilde{D}_0 = \sum_{p=0}^P \theta_p D_0(p), \quad \tilde{D}_1 = \sum_{p=0}^P \phi_p D_1(p), \quad \tilde{D}_2 = \sum_{p=0}^P \phi_p D_2(p) \quad (5.7)$$

Similarly, the aggregate side entropies are defined as:

$$\tilde{H}_1 = \sum_{p=0}^P \phi_p H_1(p), \quad \tilde{H}_2 = \sum_{p=0}^P \phi_p H_2(p). \quad (5.8)$$

$\theta = \{\theta_p, \theta_{p-1}, \dots, \theta_0\}$ and $\phi = \{\phi_p, \phi_{p-1}, \dots, \phi_0\}$ in (5.7) and (5.8) denote central and side weighting functions, respectively. These weighting functions can be used to weight the central and the side quantities at each refinement level according to their relative importance in a given application. Additionally, we impose that the sum of weighting factors should be equal to 1, i.e., $\sum_{p=0}^P \theta_p = \sum_{p=0}^P \phi_p = 1$.

We aim to jointly optimize the side and central quantizers. The optimization is formulated as a minimization problem as follows:

$$\min \tilde{D}_0 \quad \text{subject to} \quad \tilde{D}_1 \leq D_1^*, \tilde{D}_2 \leq D_2^*, \tilde{H}_1 \leq R_1^*, \tilde{H}_2 \leq R_2^*. \quad (5.9)$$

D_1^* (and D_2^*) and R_1^* (and R_2^*) in the above equation denote the target value of aggregate distortion and aggregate rate of the first (and the second) side, respectively.

5.5.1 Lagrangian Formulation

The Lagrangian functional for the multi-constraint optimization problem of (5.9) can be formulated using the Lagrangian constants λ_1 , λ_2 , μ_1 and μ_2 , as follows:

$$J(\mathbf{f}, \mathbf{g}, \lambda_1, \lambda_2, \mu_1, \mu_2) = \tilde{D}_0 + \lambda_1(\tilde{D}_1 - D_1^*) + \lambda_2(\tilde{D}_2 - D_2^*) + \mu_1(\tilde{H}_1 - R_1^*) + \mu_2(\tilde{H}_2 - R_2^*) \quad (5.10)$$

Since an SSMSDQ is balanced, one can take $\lambda_1 = \lambda_2 = \lambda$ and $\mu_1 = \mu_2 = \mu$, as the two side quantities are equally important from an optimization point of view. Let \mathbf{f}^* be such that it minimizes $J(\mathbf{f}, \mathbf{g}^*, \lambda, \mu)$ over all \mathbf{f} for a fixed \mathbf{g}^* , while \mathbf{g}^* minimizes $J(\mathbf{f}^*, \mathbf{g}, \lambda, \mu)$ over all \mathbf{g} for a given \mathbf{f}^* , for any $\lambda, \mu \geq 0$. Then $(\mathbf{f}^*, \mathbf{g}^*)$ satisfies the first order necessary conditions [68] subject to the constraints: $\tilde{D}_1 \leq \tilde{D}_1(\lambda, \mu)$, $\tilde{D}_2 \leq \tilde{D}_2(\lambda, \mu)$, $\tilde{H}_1 \leq \tilde{H}_1(\lambda, \mu)$ and $\tilde{H}_2 \leq \tilde{H}_2(\lambda, \mu)$. The parameters λ and μ are varied until the constraints on side distortions \tilde{D}_1 , \tilde{D}_2 and side entropies \tilde{H}_1 , \tilde{H}_2 , given in (5.9) are fulfilled.

5.5.1.1 Optimal Decoder

For a given encoder \mathbf{f} the optimal decoder \mathbf{g}^* is the one that reconstructs all source samples within a given quantization cell using the centroid. Hence, at any level p , for the i th cell of the m th side, the optimum side decoder is defined by the following relation:

$$\hat{x}_{p,i_p}^m = E \left[X \mid X \in S_{p,i_p}^m \right] = \frac{\int_{S_{p,i_p}^m} x \cdot f(x) dx}{\int_{S_{p,i_p}^m} f(x) dx}, \quad m \in \{1, 2\} \quad (5.11)$$

Using a training set $x(t)$, $t=1, 2, \dots, T$, the reconstruction points can be experimentally approximated as:

$$\hat{x}_{p,i_p}^m = \frac{1}{|x(t) \in (S_{p,i_p}^m)|} \sum_{x(t) \in S_{p,i_p}^m} x(t), \quad m \in \{1, 2\}, \quad (5.12)$$

where $|x(t) \in S|$ denotes the number of training samples falling in the side partition cell S . Similarly, the reconstruction points at level p for the central cell C_{p,i_p,j_p} , which is formed by the cell i_p of the first side quantizer and by the cell j_p of the second side quantizer, can be determined as:

$$\hat{x}_{p,i_p,j_p}^0 = E \left[X \mid X \in C_{p,i_p,j_p} \right] = \frac{1}{|x(t) \in (C_{p,i_p,j_p})|} \sum_{x(t) \in C_{p,i_p,j_p}} x(t). \quad (5.13)$$

5.5.1.2 Optimum Encoder:

The optimum encoder \mathbf{f}^* for a given decoder \mathbf{g} is given by the minimum of the scalar cost function of (5.10), that is:

$$\begin{aligned}
 \mathbf{f}^* &= \arg \min_{\mathbf{f}} J(\mathbf{f}, \mathbf{g}, \lambda, \mu) \\
 &= \arg \min_{\mathbf{f}} \sum_{p=0}^P \left(\theta_p \sum_{i_p=1}^{|S_p^1|} \sum_{j_p=1}^{|S_p^2|} \int_{(i_p, j_p) \in C_{\{p, p\}}^t} \left(x - \hat{x}_{p, (i_p, j_p)}^0 \right)^2 f(x) dx + \right. \\
 &\quad \left. 2\lambda\phi_p \sum_{i_p=1}^{|S_p^1|} \int_{S_{p, i_p}^1} \left(x - \hat{x}_{p, i_p}^1 \right)^2 f(x) dx - 2\mu\phi_p \sum_{i_p=1}^{|S_p^1|} \int_{S_{p, i_p}^1} f(x) \log_2 q_{S_{p, i_p}^1} \right)
 \end{aligned}$$

or

$$\mathbf{f}^* = \arg \min_{\mathbf{f}} \sum_{p=0}^P \left(\theta_p E \left[\left(X - \hat{X}_p^0 \right)^2 \right] + 2\lambda\phi_p E \left[\left(X - \hat{X}_p^1 \right)^2 \right] - 2\mu\phi_p E \left[\log_2 P_{S_p^1} \right] \right) \quad (5.14)$$

where, the expectation in the central distortion is defined by the side quantization indices (i_p, j_p) , which correspond to non-empty entries in the EIA. The cells i_p , j_p denote the coarser p th level side cells corresponding to the finest level $i = i_0$ and $j = j_0$ side cells, respectively. Note that, since for SSMDSQ the two sides are perfectly balanced, i.e., $\tilde{D}_1 = \tilde{D}_2$ and $\tilde{H}_1 = \tilde{H}_2$, only one side distortion and side entropy need to be computed, leading to $\lambda\tilde{D}_1 + \lambda\tilde{D}_2 = 2\lambda\tilde{D}_1$ and $\mu\tilde{H}_1 + \mu\tilde{H}_2 = 2\mu\tilde{H}_1$ in equation (5.14). However, for an asymmetric scalable MDSQ case \tilde{D}_1 , \tilde{D}_2 , \tilde{H}_1 and \tilde{H}_2 need to be computed even when $\lambda_1 = \lambda_2 = \lambda$ and $\mu_1 = \mu_2 = \mu$.

Bearing in mind that the side cells are composed of one or more central cells, one can take the expectation operator out of the summation in equation (5.14), leading to:

$$\mathbf{f}^* = \arg \min_{\mathbf{f}} E \left[\sum_{p=0}^P \left(\theta_p \left(x - \hat{x}_{p, (i_p, j_p)}^0 \right)^2 + 2\lambda\phi_p \left(x - \hat{x}_{p, i_p}^1 \right)^2 - 2\mu\phi_p \log_2 q_{S_{p, i_p}^1} \right) \right] \quad (5.15)$$

The minimum in equation (5.15) is achieved when the expression inside the expectation operator is individually minimized for each sample of the training set $x(t)$, i.e., by optimally mapping each sample x to a side cell pair (i_p, j_p) that results in the smallest value of the aforementioned expression. In fact, this is equivalent to creating an optimal finest central partition; since the finest central partition together with the EIA defines side cells at any level p . Using the training set $x(t)$, the expression inside the expectation operator can be minimized by defining the finest central partition $C_{0, (i, j)}$ as below:

$$C_{0,(i,j)} = \left\{ \begin{array}{l} x : \sum_{p=0}^P \theta_p \left(x - \hat{x}_{p,(i_p,j_p)}^0 \right)^2 + 2\lambda\phi_p \left(x - \hat{x}_{p,i_p}^1 \right)^2 - 2\mu\phi_p \log \left(N_{S_{p,i_p}^1} / T \right) \\ \leq \\ \sum_{p=0}^P \theta_p \left(x - \hat{x}_{p,(i'_p,j'_p)}^0 \right)^2 + 2\lambda\phi_p \left(x - \hat{x}_{p,i'_p}^1 \right)^2 - 2\mu\phi_p \log \left(N_{S_{p,i'_p}^1} / T \right), \\ \forall (i, j) \neq (i', j') \end{array} \right\}, \quad (5.16)$$

where $N_{S_{p,i_p}^1} = \left| \{x(t) \in S_{p,i_p}^1\} \right|$. Defining:

$$\alpha_{ij} = \sum_{p=0}^P \left(\theta_p \hat{x}_{p,(i_p,j_p)}^0 + 2\lambda\phi_p \hat{x}_{p,i_p}^1 \right), \quad (5.17)$$

and

$$\beta_{ij} = \sum_{p=0}^P \left(\theta_p \left(\hat{x}_{p,(i_p,j_p)}^0 \right)^2 + 2\lambda\phi_p \left(\hat{x}_{p,i_p}^1 \right)^2 - 2\mu\phi_p \log \left(N_{S_{p,i_p}^1} / T \right) \right), \quad (5.18)$$

the expression (5.16), which refers to the finest central partition cells $C_{0,(i,j)}$, reduces to:

$$C_{0,(i,j)} = \left\{ x : 2\alpha_{ij}x - \beta_{ij} \geq 2\alpha_{i'j'}x - \beta_{i'j'}, \forall (i, j) \neq (i', j') \right\}. \quad (5.19)$$

The finest central partition $C_{0,(i,j)}$, as given by (5.19), is solved using the extreme-point algorithm of [113].

5.5.2 Optimization Algorithm

For a given value of λ and μ , the optimization algorithm is as follows:

- 1) SET iteration counter $n = 1$. SET $J_0 = \infty$.
INITIALIZE the encoder-decoder pair $(\mathbf{f}_0, \mathbf{g}_0)$ using a two-description symmetric SSMDSQ (i.e., symmetric EIA + uniform finest central quantizer).
SET the stopping threshold ε to an appropriate small value.
- 2) Determine the optimum decoder $\mathbf{g}_n = \left\{ \left\{ \mathbf{g}_p^0 \right\}_{p=0}^{p=P}, \left\{ \mathbf{g}_p^m \right\}_{p=0}^{p=P} \right\}$ for $m \in \{1, 2\}$, using the per-level decoder for the center, \mathbf{g}_p^0 , and the sides, \mathbf{g}_p^m , as explained in Section 5.5.1.1.
- 3) Determine the optimum encoder \mathbf{f}_n for the decoder \mathbf{g}_n found in the last step by solving (5.19) using the extreme-point algorithm of [113].
- 4) Compute the cost function J_n using the encoder-decoder pair $(\mathbf{f}_n, \mathbf{g}_n)$.
- 5) IF $[J_{n-1} - J_n] / J_n < \varepsilon$ THEN STOP. ELSE $n = n + 1$ and GO TO STEP 2.

5.5.3 Discussion

Since both step 2 and step 3 in the optimization algorithm reduce the cost function value with respect to the previously computed value, a non-increasing sequence of

cost function values is obtained over all iterations. As the cost function cannot be negative, the produced sequence is lower bounded by zero and convergence to some positive value is thus guaranteed.

The complexity of the optimization increases exponentially with P due to the exponential increase in the total number of quantization cells. The appropriate value of λ and μ are any set of values that fulfill the constraints on aggregate rate and distortions, as given in equation (5.9). When the quantization indices are encoded using FLC, the entropy constraints in equation (5.9) become meaningless, i.e., one can set $\mu = 0$. However, for VLC of quantization indices, both λ and μ need to be carefully searched so as to find the appropriate local minima which fulfills all the constraints of equation (5.9). In general, it is quite difficult to determine optimum values of λ and μ . One way to choose proper λ and μ is by iteratively narrowing the possible range of values between two extreme points using the bisection method [108].

Another important aspect is the appropriate selection of the weighting functions θ , ϕ^m . In our work, the selected side and central weighting factors for any level p are taken as:

$$\phi_p^m = \frac{\log_2 |S_p^m|}{\sum_{p=0}^P \log_2 |S_p^m|}, \quad \text{and} \quad \theta_p = \frac{\log_2 |C_p|}{\sum_{p=0}^P \log_2 |C_p|}, \quad \text{where, } 0 \leq p \leq P. \quad (5.20)$$

Figure 5.11 plots the central SNR versus the side rate behavior of the proposed SSMSDQ optimized for a zero-mean unit-variance Gaussian source PDF. The *consi-*

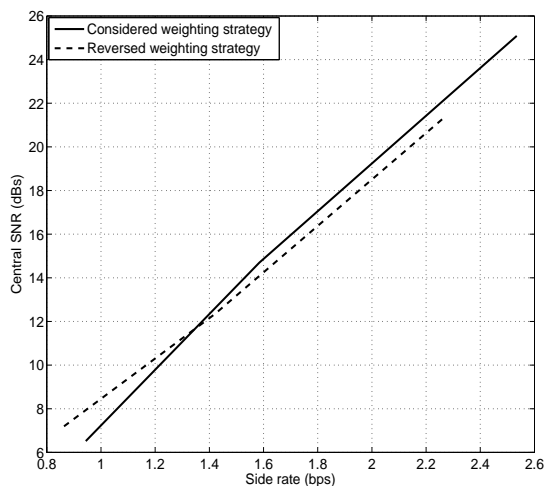


Figure 5.11: Central SNR versus side rate plot for optimized SSMSDQ. The optimization is carried out using the considered and the reverse weighting strategy for a zero-mean, unit-variance Gaussian source.

dered weighting strategy is the one given in equation (5.20), where the MSB is given the lowest while the LSB is given the highest weight. In the *reverse weighting strategy* the weights of the considered strategy are applied in the reverse order, i.e., the MSB is given the highest while the LSB is given the lowest weight. One may observe that, for rates below 1.3 bps the reverse weighting strategy yields an SNR gain compared to the *considered weighting strategy*. However, for rates above 1.3 bps, the considered weighting strategy provides superior SNR performance compared to the reverse weighting strategy.

Intuitively, we can say that, the source reconstruction at high rates should be given more importance than the reconstruction at low rates, thus θ and ϕ^m should be strictly increasing functions of the rate – or otherwise, strictly decreasing functions of level p . However, at low rates, it is better to use the reverse weighting strategy, wherein, θ and ϕ^m are strictly increasing functions of level p .

Although the proposed optimization algorithm is presented for two-side SSMDQs, it can be straightforwardly extended to any scalable MDSQ with more than two (say M) sides, by defining the following cost function:

$$J = \tilde{D}_0 + \sum_{m=1}^M (\lambda_m \tilde{D}_m + \mu_m \tilde{H}_m), \quad (5.21)$$

where, the aggregate central and side ($m = 1, 2, \dots, M$) quantities are given as:

$$\tilde{D}_0 = \sum_{p=0}^P \theta_p D_0(p), \quad \tilde{D}_m = \sum_{p=0}^P \phi_p^m D_m(p) \quad \text{and} \quad \tilde{H}_m = \sum_{p=0}^P \phi_p^m H_m(p).$$

Unlike the SSMDQ, for an asymmetric scalable MDSQ we have $\lambda_m \neq \lambda_{m'}$ and $\mu_m \neq \mu_{m'}$, where $m \neq m' \forall (m, m')$. Moreover, in contrast to SSMDQs where one weighting function $\phi = \{\phi_p, \phi_{p-1}, \dots, \phi_0\}$ for all side quantizers is defined, an asymmetric scalable MDSQ optimization needs a separate weighting function $\phi^m = \{\phi_p^m, \phi_{p-1}^m, \dots, \phi_0^m\}$ for each side m .

Since the number of terms in equation (5.21) grow linearly with M , we can say that for a fixed P , the complexity of the proposed optimization algorithm would increase linearly with M . However, it is worth pointing out that designing an efficient multi-dimensional IA is a significantly complex problem by itself. A solution for $M = 3$ is given in [111]. Alternatively, a less complex way of generating 2^k , $k \geq 1$, number of descriptions is described in [15].

In real multimedia sources such as image/video/3-D mesh, the input source is transformed to achieve the decorrelation of spatial-domain components, e.g., pixels, vertex positions. The transformed signal consists of a number of frequency/resolution subbands. In practice, the proposed optimization methodology can be employed to carry out an optimized multiple description quantization of transformed multimedia sources. In general, a separate optimization of the quantizer

can be performed for each subband. This is logical as each subband has its own statistical properties, e.g., mean, variance, and the histogram of each subband can be modeled using known distribution functions, e.g., GG distribution [69], LM distribution (see Section 3.3.2). The training data required to train the optimization algorithm can be acquired in two ways: 1) the model parameters for each subband are first estimated and the training sequence is generated using the corresponding source generation function, 2) training data can be accumulated from a number of real sources of similar perceptual properties. Note that, in the first method, quantization performance directly relates to the accuracy of source modeling. However, this is not the case for the second method as no modeling is performed. The main difficulty in the second method is the availability of enough source examples to generate sufficient amount of training data. In the following section, performance evaluation of the proposed optimization algorithm is given for the GG distributions. Furthermore, using the optimized quantizers, compression of wavelet decomposed image sources is assessed.

5.5.4 Performance Comparison

This section evaluates the performance of the proposed optimization algorithm for the family of GG distributions [78]. GG distributions are commonly used to model histograms of wavelet coefficients in images [69]. We consider three instances of the GG distribution of equation (3.18), defined by $\alpha = 1$ (Laplacian PDF), $\alpha = 1.5$ and $\alpha = 2$ (Gaussian PDF). First, a training sequence of 10^6 floating point samples, drawn from each of the considered source distributions, is used to carry out the quantizer optimization, as explained in the start of Section 5.5. The convergence rate of the optimization algorithm depends on the selected value of the termination threshold ε . It has been experimentally observed that for $\varepsilon = 10^{-4}$ the algorithm converges in between 45 to 60 iterations; on average, the optimum is attained after 50 iterations for all three source PDFs. The optimized quantizers are thereafter used to quantize the actual source samples. The experiments are carried out using sources of 25344 samples¹¹ and each D-R point is evaluated as a mean over 100 trials. To compare the D-R performance of different scalable MDSQs, the SNR metric is employed, that is:

$$SNR = 10 \log_{10} \left(\frac{\sigma^2}{D} \right) (\text{dBs}). \quad (5.22)$$

Table 5.3 reports the SNR improvements brought by the proposed optimization

¹¹ The size of HH wavelet transform subband of the luma component of a Common Intermediate Format (CIF) video frame.

algorithm for the connected-cell scalable MDSQs of Figure 5.6 at various side rates and for a Gaussian source PDF. Both level-constrained (using FLC) and entropy-constrained (using VLC) cases are considered. Since the optimization is done for the average quantities, over $P+1$ embedded levels, it cannot be generally ensured that the distortion performance at each rate point is also improved. For the level-constrained case the optimization always results in SNR improvements at all rate points with respect to the un-optimized quantizers. However, the same is not true for the entropy-constrained case. Negative sign values in Table 5.3 indicate instances when the optimized distortion is higher than the un-optimized distortion. However, for the majority of rate points, the optimized distortion is far below the un-optimized one, leading to an average improvement of central and side SNRs, – see Table 5.3. To demonstrate that the proposed algorithm improves the overall D-R, Table 5.3 also reports the *accumulated gain* G_{acc} :

$$G_{acc} = \frac{1}{K^2} \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} \left(SNR_{opt}(R_{k_1}, R_{k_2}) - SNR_{unopt}(R_{k_1}, R_{k_2}) \right), \quad (5.23)$$

where $R_0 = 0$, i.e., the source SNR is evaluated using a total of K^2 rate points. Table 5.3 clearly points out that the overall D-R improvement in case of VLC is approximately 0.5 dB, while in FLC it grows larger than 2.5dB, for both the symmetric and asymmetric connected-cell quantizers.

Similar to the connected-cell case, the optimization algorithm provides an important average gain for disconnected-cell quantizers as well – see Table 5.4. In this experimental setup, three source PDFs are quantized using both level-constrained and entropy-constrained optimized quantizers. We notice that the optimization algorithm contributes significant average gains for both the ML IA and asymmetric EIA [51] based scalable MDSQs. Again, the obtained SNR gains in the level-constrained case are higher compared to the entropy-constrained optimization case. Moreover, one observes that larger optimization gains are obtained for ML IA based quantizers while lesser but still significant gains are achieved for the asymmetric EIA of [51]. The smallest optimization gains are obtained for the proposed symmetric EIA based SSMDQS. This is to be expected, since the proposed symmetric EIA serves as an initialization and the optimization algorithm converges to a local minimum.

Table 5.3: SNR difference (in dB) between the optimized and the un-optimized scalable MDSQs of [42], for a unit-variance Gaussian source.
 $\Delta_{SNR} = SNR_{opt} - SNR_{unopt}$

| SIDE RATE(BPS) | VARIABLE-LENGTH CODING | | | | | | FIXED-LENGTH CODING | | | | | |
|-------------------|------------------------|---------------|-----------------|---------------|---------------|---------------|---------------------|---------------|---------------|---------------|-----------------|---------------|
| | EMDSQ [42] | | Proposed SSMDSQ | | EMDSQ [42] | | Proposed SSMDSQ | | EMDSQ [42] | | Proposed SSMDSQ | |
| | Central | Side | Central | Side | Central | Side | Central | Side | Central | Side | Central | Side |
| 0.25 | 0.2801 | 0.1485 | -0.1897 | -0.1338 | 0.1388 | 0.1189 | 0.0848 | 0.0861 | 0.1388 | 0.1189 | 0.0848 | 0.0861 |
| 0.50 | 0.9289 | 0.3524 | -0.4830 | -0.3239 | 0.2963 | 0.2493 | 0.1792 | 0.1795 | 0.2963 | 0.2493 | 0.1792 | 0.1795 |
| 0.75 | 0.4208 | 0.1809 | -0.5139 | -0.3574 | 0.4878 | 0.3938 | 0.2893 | 0.2881 | 0.4878 | 0.3938 | 0.2893 | 0.2881 |
| 1.00 | 0.8799 | 0.2699 | 0.3037 | 0.2731 | 0.7111 | 0.5495 | 0.4244 | 0.4186 | 0.7111 | 0.5495 | 0.4244 | 0.4186 |
| 1.50 | 3.0661 | 0.5364 | 2.6992 | 1.9407 | 1.3252 | 0.9182 | 0.7894 | 0.7583 | 1.3252 | 0.9182 | 0.7894 | 0.7583 |
| 2.00 | 1.2813 | -0.8226 | 1.4177 | 0.7778 | 2.3652 | 1.3752 | 1.3028 | 1.2151 | 2.3652 | 1.3752 | 1.3028 | 1.2151 |
| 2.50 | 0.9368 | -1.2337 | 0.4683 | -0.2686 | 3.2823 | 1.6343 | 2.4110 | 2.0533 | 3.2823 | 1.6343 | 2.4110 | 2.0533 |
| 3.00 | 1.5205 | 0.1177 | 0.8107 | 0.0147 | 5.0545 | 2.1108 | 3.7721 | 2.9230 | 5.0545 | 2.1108 | 3.7721 | 2.9230 |
| 3.50 | 1.5191 | 0.6929 | 0.4279 | -0.2314 | 4.9853 | 2.0007 | 3.5043 | 2.6103 | 4.9853 | 2.0007 | 3.5043 | 2.6103 |
| 4.00 | -0.4100 | 0.3080 | 2.3900 | 0.6094 | 4.4758 | 1.7408 | 3.0692 | 2.1351 | 4.4758 | 1.7408 | 3.0692 | 2.1351 |
| AVERAGE | 1.0423 | 0.0550 | 0.7331 | 0.2301 | 2.3122 | 1.1091 | 1.5827 | 1.2667 | 2.3122 | 1.1091 | 1.5827 | 1.2667 |
| G_{rec} | 0.5279 | | 0.5533 | | 2.9121 | | 2.5317 | | 2.9121 | | 2.5317 | |

Table 5.4: SNR difference (in dB) between the optimized and the un-optimized scalable MDSQs, for three unit-variance source PDFs.

| | VARIABLE-LENGTH CODING | | | FIXED-LENGTH CODING | | |
|-------------------|------------------------|--------------------|-----------|---------------------|--------------------|-----------|
| | Gaussian | GG, $\alpha = 1.5$ | Laplacian | Gaussian | GG, $\alpha = 1.5$ | Laplacian |
| MLIA [113] | Central | 3.5797 | 3.5058 | 4.3025 | 3.3470 | 4.3163 |
| | Side | 1.9491 | 1.7941 | 2.1494 | 1.9345 | 2.1105 |
| | G_{acc} | 3.1228 | 3.3387 | 3.6178 | 3.2143 | 3.7737 |
| EIA [51] | Central | 1.0404 | 1.0780 | 2.3850 | 1.2618 | 2.8331 |
| | Side | 1.0013 | 0.8520 | 0.6282 | 0.9929 | 0.6811 |
| | G_{acc} | 0.9625 | 1.0659 | 1.8652 | 0.9214 | 1.8730 |
| PROPOSED | Central | 1.1183 | 0.9795 | 0.5215 | 1.3276 | 2.3906 |
| | Side | 0.5497 | 0.5973 | 0.5983 | 0.5841 | 1.0306 |
| | G_{acc} | 0.7592 | 0.6004 | 0.3433 | 0.8755 | 1.5755 |

To verify their optimized performance for real multimedia data, the proposed SSMDSQs are utilized for SMDC of wavelet-decomposed images. Table 5.5 reports the reconstructed image quality versus rate when subbands resulting from the 6-level CDF 9/7 [20] wavelet transform are quantized using the five diagonal SSMDSQ and the equivalent asymmetric scalable MDSQ of [51]. The LL subband is losslessly coded and the ensuing information is duplicated in both descriptions. This is due to the fact that side quantizer cells are disjoint and quantization of LL subband would result in high distortion of side descriptions. This problem can be avoided by losslessly coding the LL subband in both the descriptions. The bit-planes of the other frequency subbands are coded using the adaptive arithmetic coding in quality scalable manner.

For the optimized case, the subband histograms are first modeled using GG distributions. The estimation of model parameters is done using the *method of moments*, explained in [69]. For each wavelet subband, the estimated shape-parameter α_{est} is mapped to the closest value in the set $\{1.0, 1.5, 2.0\}$ and the SSMDSQ optimized for the source PDF corresponding to the selected value of $\alpha \in \{1.0, 1.5, 2.0\}$ is then employed to quantize the subband¹². Since the original optimization is done for zero-mean unit-variance sources, the quantizer's thresholds and reconstruction alphabets are adjusted based on the estimated mean m_{est} and the standard deviation σ_{est} of each subband's coefficients.

The results in Table 5.5 show that the proposed SSMDSQ (with and without optimization) provide improved performance over the scalable MDSQ of [51]. We highlight that for the case of joint decoding of the descriptions, the improvement in the PSNR of the reconstructed image can be of up to 4-5dB. Furthermore, visual comparisons of joint descriptions decoding at level $[p_2, p_1] = [1, 1]$, shown in Figure 5.12, demonstrate that the reconstructed image quality provided by the proposed scheme is significantly improved compared to that of [51]. Additionally, one may notice that, unlike the asymmetric scalable MDSQ of [51], the side PSNRs for the SSMDSQ are nearly equal at different side rates, thereby confirming its fairly balanced description structure.

The coding system used for computing the results of Table 5.5 is a basic one, wherein, the generated bit-planes were progressively coded using adaptive arithmetic entropy coding. In general, scalable image codecs, e.g., [108], [94], also employ bit-plane coding schemes such as run-length coding, quadtree coding, to

¹² The same experiment is also performed using $\alpha \in \{1.0, 1.25, 1.5, 1.75, 2.0\}$, however, the obtained results are just marginally different from the one obtained with $\alpha \in \{1.0, 1.5, 2.0\}$.

further exploit the spatial correlation within a given bit-plane. For the results of Table 5.5, we do not claim the best performance of the image coding system itself; only the performance with respect to other quantization strategies, when employed in the above-presented basic image coding system, is highlighted. We point out that such a comparison is fair and at the same time indicative, since the addition of bit-plane coding would bring a roughly constant performance gain for systems employing the proposed and existing quantizers. Due to the lack of bit-plane coding in our basic coding system, the rate values are relatively high with respect to JPEG-2000 [108].



Figure 5.12: Visual comparison of images reconstructed using the central quantizer at level $\mathbf{p} = [1, 1]$. Images (a) and (b) are the reconstructions using the asymmetric EIA [51]. Reconstructions using the proposed optimized SSMDQ are shown by the images (c) and (d). The reported rate is the sum rate of two sides, i.e., $R = R_1 + R_2$.

Table 5.5: Performance evaluation of the proposed SSMDSQ for wavelet-based SMDC of Lena (512x512) and Barbara (512x512) images. $PSNR_0$, $PSNR_{1,2}$ (resp. $R_{1,2}$) denote the central and the sides peak signal-to-noise-ratio in dB (resp. mean side rate in bits per pixel (bpp)).

| $R_{1,2}$ (bpp) | EIA [51] | | PROPOSED UN-OPTIMIZED | | PROPOSED OPTIMIZED | | | | | |
|--------------------|------------------|----------------------|-----------------------|----------------------|--------------------|----------------------|-------|-------|-------|-------|
| | $PSNR_0$ (dB) | $PSNR_{1,2}$ (dB) | $PSNR_0$ (dB) | $PSNR_{1,2}$ (dB) | $PSNR_0$ (dB) | $PSNR_{1,2}$ (dB) | | | | |
| LENA | 0.50 | 18.58 | 16.91 | 18.03 | 20.54 | 18.76 | 18.69 | 23.03 | 19.89 | 19.81 |
| | 0.75 | 20.30 | 16.97 | 18.82 | 23.06 | 20.22 | 20.03 | 26.65 | 20.35 | 21.06 |
| | 1.00 | 23.80 | 17.04 | 19.79 | 29.24 | 22.30 | 21.39 | 28.61 | 20.56 | 21.36 |
| | 1.25 | 27.21 | 17.39 | 20.96 | 31.03 | 22.46 | 21.53 | 31.94 | 20.78 | 21.69 |
| BARBARA | 0.60 | 19.26 | 17.11 | 18.39 | 22.35 | 19.66 | 19.87 | 25.72 | 20.20 | 20.39 |
| | 0.80 | 20.84 | 17.15 | 18.98 | 25.40 | 20.84 | 21.10 | 27.28 | 20.45 | 20.55 |
| | 1.00 | 23.82 | 17.19 | 19.67 | 29.80 | 21.69 | 21.97 | 29.49 | 20.72 | 20.71 |
| | 1.20 | 26.03 | 17.33 | 20.74 | 31.82 | 21.83 | 22.10 | 33.53 | 21.03 | 20.88 |

Table 5.6: Un-optimized SSMDSQ based quadtree coding system versus the equivalent single and multiple (duplicated) description image coding systems. For SSMDSQ based system, the central reconstruction quality ($PSNR_c$) is reported.

| | RATE(BPP) | JPEG-2000 (DUPLICATED) | SSMDSQ | JPEG-2000 (SDC) | QT-L (SDC) |
|---------|-----------|---------------------------|--------|--------------------|---------------|
| LENA | 0.14 | 28.50 | 29.71 | 31.39 | 31.60 |
| | 0.18 | 29.53 | 31.52 | 32.57 | 32.69 |
| | 0.46 | 33.65 | 34.94 | 36.82 | 36.89 |
| BARBARA | 0.26 | 25.89 | 27.14 | 29.02 | 28.81 |
| | 0.90 | 32.06 | 33.51 | 37.11 | 37.18 |
| | 1.60 | 36.16 | 38.79 | 42.01 | 41.75 |

Table 5.6 gives performance results of the proposed SSMDSQ when incorporated in a full-blown scalable image coding system. For bit-plane coding, we have used the wavelet-based quadtree coding system of [76, 94], called QT-L – for more information we refer to Section 2.5.5.2. The SSMDSQ together with QT-L coding system is compared against the duplicated description JPEG-2000 system [108]. It can be easily seen that for the same rate, the duplicated JPEG-2000 performs worse than the SSMDSQ based system for both Lena and Barbara images. It is seen that the proposed multiple description image codec gives a performance advantage of 1-2 dB with respect to JPEG-2000 duplicated.

For comparison purposes, we also include the single description coding results of the JPEG-2000 and the QT-L codecs, showing that they perform similarly. The proposed multiple description image codec underperforms the single description JPEG-2000 and QT-L. Since no channel losses are considered, the performance deficit of our codec with respect to JPEG-2000 (or QT-L) is the price of performing error-resilience. However, this performance deficit will pay off when transmission is carried out for lossy channel, as we will see in the next section.

5.6 TRANSMISSION VIA PACKET LOSS CHANNEL

In order to evaluate the error-resilience properties of an SMDC system employing the proposed SSMDSQs, this section considers packetized transmission of source descriptions via packet erasure channels. We consider transmission via a single link which is divided into multiple logical channels, namely, packets representing different source descriptions are transmitted using time-division multiplexing. Let $p_L \geq 0$ denote the average probability of packet loss on the transmission link. For a two-description SMDC, the average expected distortion at the decoder, when sending N_m packets for the m th description, $m \in \{1, 2\}$, is given by:

$$\bar{D}(N_1, N_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \Pr(n_1, n_2) D(R_{n_1}, R_{n_2}) \quad (5.24)$$

where

$$\Pr(n_1, n_2) = \prod_{m=1}^2 (1 - p_L)^{n_m} p_L^{u[N_m - 1 - n_m]} \quad (5.25)$$

In equation (5.25), $u(n)$ represents the discrete time unit-step function, which is defined as follows:

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

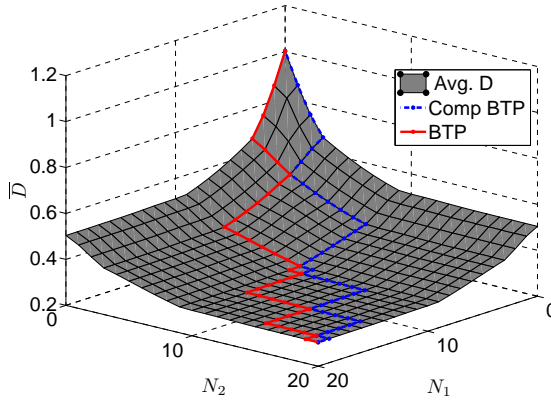


Figure 5.13: Average experimental D-R at $p_L = 0.15$ as a function of transmitted packets (N_1, N_2) , for a unit-variance Gaussian source. The best and the complementary best TPs are illustrated using solid (red) and dashed (blue) lines, respectively.

For a symmetric D-R surface, $D(R_{n_1}, R_{n_2})$, of which a schematic representation is illustrated in Figure 5.2, and for a given p_L , equation (5.24) yields a symmetric average distortion surface – see Figure 5.13.

Notice that in this work we consider IID packet erasures over the transmission link. In general, transmission channels may well encounter burst packet losses. In such scenarios, the simplification resulting from the assumption of IID losses, i.e., equation (5.24) can still be employed, provided that bursts of losses can be spread across independent packets by interleaving [117].

5.6.1 Balanced Transmission of Descriptions

In today's packet switched networks, nodes randomly drop packets to match link rates. In point-to-point communication employing layered coding, packets arrive at a network node's queue in the order of their significance. Therefore, it is more beneficial to drop packets that arrived later rather than packets that arrived earlier.

Assuming such a packet drop strategy, we first consider scalable multiple description transmission when the encoder is unaware of the packet loss probability on the channel. For a balanced transmission, packets from the two descriptions are alternatively placed over the transmission link and are assumed to arrive in the same order at the network node. If the network node is forced to drop L packets, the packet dropping strategy will always be one of the following:

$$L \text{ odd} : \left\{ \left(\frac{L+1}{2}, \frac{L-1}{2} \right), \left(\frac{L-1}{2}, \frac{L+1}{2} \right) \right\},$$

$$L \text{ even} : \left\{ \frac{L}{2}, \frac{L}{2} \right\},$$

where the first and the second term in $\{\bullet, \bullet\}$ denote the number of packets dropped from the first and the second description, respectively. Thus, the transmission is still approximately balanced for any number of packets dropped by the network node. This is not true for unbalanced transmission, for which packets belonging to the two descriptions are not alternatively placed on the transmission link. In this case, the network node may drop all packets belonging to a single description, thus enabling only source reconstruction with side distortion, which is usually much higher than the central distortion. As a consequence, for an unbalanced transmission, a certain loss pattern may lead to a much higher decoding distortion compared to other loss patterns for the same total number of lost packets. Hence, one concludes that a balanced transmission of descriptions is always a better choice if the amount of packet loss is unknown at the encoder.

The performance of a scalable MDSQ in the considered packet-based SMDC system is reported using the average signal-to-noise-ratio, \overline{SNR} , which specifies the average source reconstruction quality over all packet loss patterns. To compute the \overline{SNR} , equation (5.22) is used, in which D is replaced by the average distortion \overline{D} . One notices that, for a given rate, \overline{SNR} decreases when the average packet loss rate increases, and vice versa.

5.6.1.1 Connected-cell Quantizers

The \overline{SNR} profile of the connected-cell EMDSQ of [42] and the proposed SSMDSQ with three levels ($P=2$), for a unit-variance Gaussian source over a wide range of packet loss probability p_L , is depicted in Figure 5.14. Both FLC and VLC results are reported. As expected, due to rate savings, VLC outperforms FLC for a given packet loss probability p_L . The results demonstrate that the proposed un-optimized SSMDSQ outperforms the un-optimized EMDSQ of [42] for all p_L values greater than 1% and 5% when considering FLC and VLC, respectively. In

addition, when optimization is applied – see Section 5.5, the proposed optimized SSMSDQ outperforms the optimized EMDSQ of [42] for p_L values greater than 3% for FLC. Alternatively, in case of VLC, both optimized quantizers operate on par for p_L values above 7%. The obtained performance improvements for high packet loss probabilities, show the benefit of balanced transmission, as achieved using the proposed SSMSDQ compared to the unbalanced EMDSQ of [42]. For very low packet loss probability, the proposed SSMSDQ falls behind the EMDSQ of [42], which reveals the penalty for providing a perfectly symmetric D-R function.

Furthermore, in SMDC-based transmission, the proposed optimization algorithm brings notable \overline{SNR} gains for both the proposed (SSMSDQ) and the contemporary (EMDSQ [42]) connected-cell quantizers, as shown in Figure 5.14. We also note

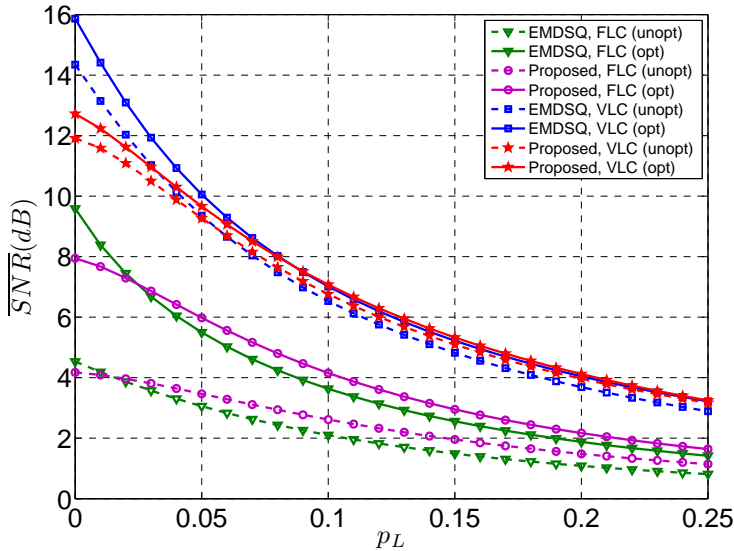
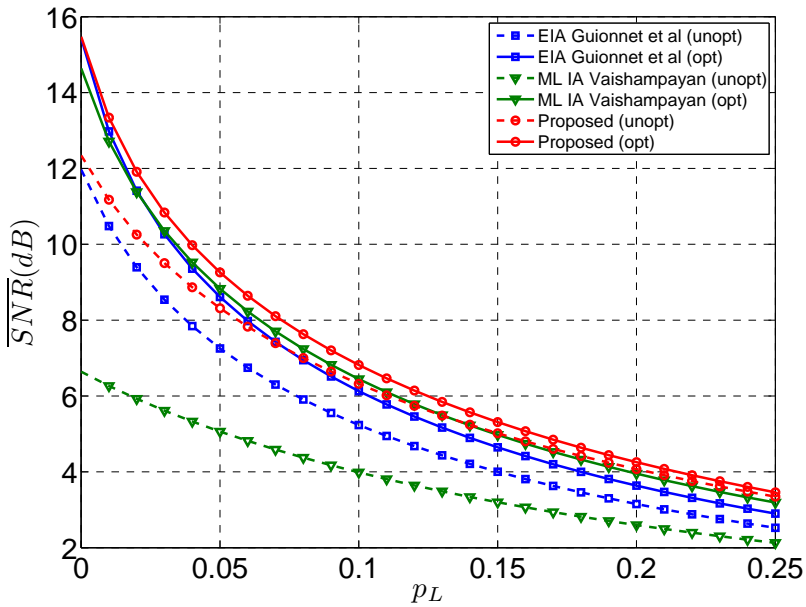


Figure 5.14: \overline{SNR} profiles for symmetric and asymmetric connected-cell scalable MDSQs at (a) $R_1 = R_2 = 3.0$ bps, for a unit-variance Gaussian source. Dashed lines correspond to un-optimized (*unopt*), while solid lines are used for optimized (*opt*) quantizers.

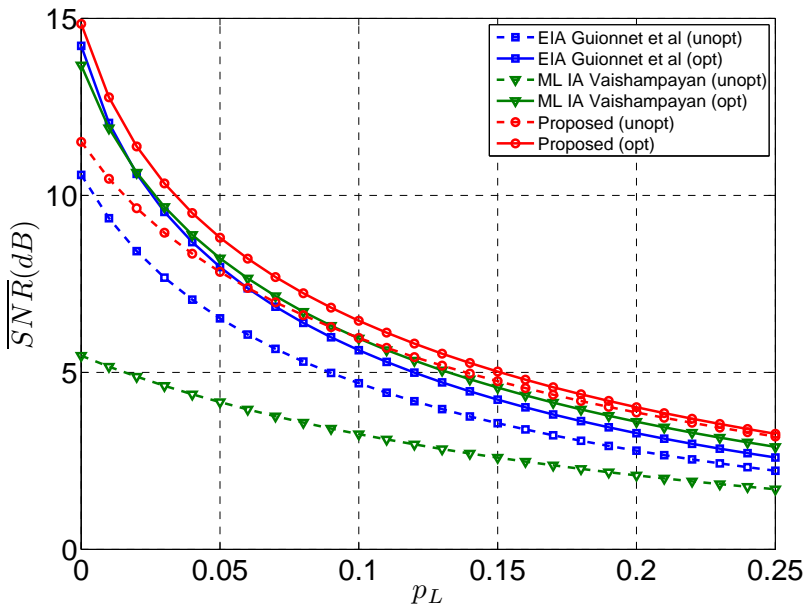
that the benefit of the proposed optimization algorithm reduces as the probability of packet loss, p_L , increases. This is anticipated since the optimization is carried out based on ideal transmission, that is, $p_L = 0$. As explained in Section 5.5, this is done in order to ensure that the optimization algorithm is performed independently of the packet loss statistics, which are often varying.

5.6.1.2 Disconnected-cell Quantizers

For disconnected-cell quantizers, the \overline{SNR} profiles for the three considered source PDFs are depicted in Figure 5.15, Figure 5.16. Guionnet et al. denotes the



(a)



(b)

Figure 5.15: \overline{SNR} profiles for symmetric and asymmetric disconnected-cell scalable MDSQs, at $R_1 = R_2 = 2.0$, for (a) Gaussian, (b) GG with $\alpha = 1.5$, source PDFs.

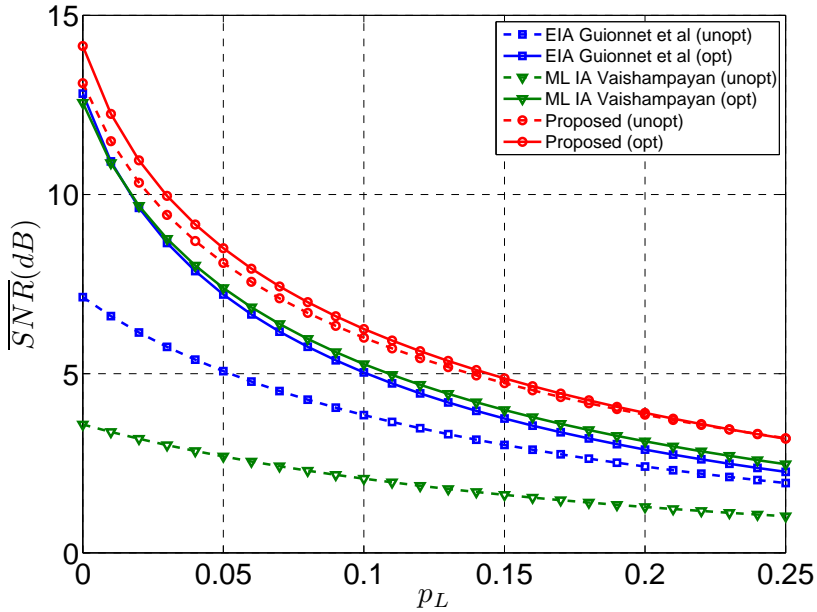


Figure 5.16: \overline{SNR} profile for symmetric and asymmetric disconnected-cell scalable MDSQs, at $R_1 = R_2 = 2.0$, for a unit-variance Laplacian PDF.

EIA of [51]. We notice that the optimized quantizers significantly outperform their un-optimized equivalents for all three PDFs. One observes that the improvements attained by the proposed optimization algorithm are higher for the ML IA [113] based quantizer, which comprises disconnected side and central cells, than for quantizers that feature connected central and disconnected side cells, e.g., the asymmetric EIA [51] and the proposed symmetric EIA based quantizers. This is because in the former, a reduction in the cost for a central cell will always result in a decrease of the cost for the corresponding side cells, and vice versa. However, this is not always true for quantizers belonging to the second category, as the optimization of the cost for the central and the side cells may have conflicting requirements.

In essence, the proposed symmetric EIA based quantizer significantly outperforms the existing ML IA [113] and EIA [51] based quantizers, with and without optimization. The obtained gains are systematic for all the considered source PDFs. Interestingly, we notice that for the Laplacian source PDF (Figure 5.16), the proposed un-optimized quantizer's performance surpasses the one obtained with the optimized quantizers of [113] and [51]. The latter highlights the vastly increased coding performance of the proposed SSMDQS with respect to existing quantization methods.

Table 5.7: Average decoded PSNR (in dB) for the Lena image at $R_1 = R_2 = 1.25$ bpp using balanced transmission of coded descriptions. The experimental settings of Table 5.5 are reused to compute the result in this table.

| | \overline{PSNR} $p_L = 0.0$ | \overline{PSNR} $p_L = 0.05$ | \overline{PSNR} $p_L = 0.10$ | \overline{PSNR} $p_L = 0.20$ |
|-----------------------|----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| EIA [51] | 27.21 | 23.21 | 21.38 | 19.47 |
| SSMDSQ (UNOPT) | 31.03 | 26.22 | 24.00 | 21.50 |
| SSMDSQ (OPT) | 31.94 | 26.81 | 24.56 | 22.06 |

Table 5.7 reports the average decoded PSNR, for the SSMDSQ and the equivalent scalable MDSQ of [51], when multiple description transmission of the Lena image is carried out over a packet loss channel. The experimental setup of Table 5.5 is reused to compute the average PSNR for the decoded image for four packet loss rates. In this context, the proposed SSMDSQ without optimization improves the quality of the reconstructed image by 2-3dB. On top of this, the proposed optimization approach yields an additional PSNR gain of up to 0.5 dB on average.

5.6.2 Unbalanced Transmission of Descriptions

In case the packet loss rate can be known at the encoder, an efficient transmission strategy is established by an unbalanced transmission of descriptions. Depending on p_L , we formulate a *greedy* packet scheduling strategy which minimizes \bar{D} – given by (5.24), for a certain total N_{tot} of packets sent over the transmission channel. For a given N_{tot} , an appropriate combination of packets (N_1^*, N_2^*) can be found as:

$$(N_1^*, N_2^*) = \min_{N_1, N_2} \bar{D}(N_1, N_2), \text{ subject to, } N_1 + N_2 = N_{tot}. \quad (5.26)$$

By sequentially increasing the total number of transmitted packets, exhaustive search solutions of (5.26) define a so-called *best* transmission path (TP). Specifically, the encoder regulates packet contributions from both sides, as suggested by the best TP, to minimize \bar{D} at the decoder. We notice that when $p_L = 0$, then $\Pr(n_1, n_2) = 0 \quad \forall n_1 < N_1, n_2 < N_2$, and $\Pr(N_1, N_2) = 1$. Therefore, for $p_L = 0$, the best TP corresponds to traversing the source D-R at the minimum source distortion.

Notice that in the aforesaid packet scheduling, for a given N_{tot} , only packet combinations which agree with the scheduling of $N_{tot} - 1$ packets are searched. In general, an optimum combination for $N_{tot} - 1$ may not lead to an optimum combination for N_{tot} . Therefore, since the packet scheduling strategy is considered *greedy*, it may lead to reduced performance of an SMDC system for large N_{tot} . A better approach would be to determine an *overall best* TP by considering all possible combinations for the highest N_{tot} and accordingly scheduling the transmission for

smaller rates. However, such an approach requires all packets of all descriptions to be available beforehand, which brings an unreasonable structural delay in many real-time applications, e.g., multimedia transmission.

For an SSMSQ-based SMDC system, which guarantees a symmetric D-R surface, a permutation of the solution of (5.26) is also a valid solution. Therefore, this leads to the existence of two complementary best TPs as demonstrated in Figure 5.13. Since both the best and its complementary TPs result in the same average distortion \bar{D} , the determination of the best TP in a SSMSQ based SMDC scenario requires searching only half of the \bar{D} space, as defined by $N_1 \geq N_2$ (or $N_2 \geq N_1$). This is in contrast to SMDC systems based on contemporary asymmetric scalable MDSQs, e.g., [51], in which one needs to perform an exhaustive search for any (N_1, N_2) pair, for which $N_{tot} = N_1 + N_2$.

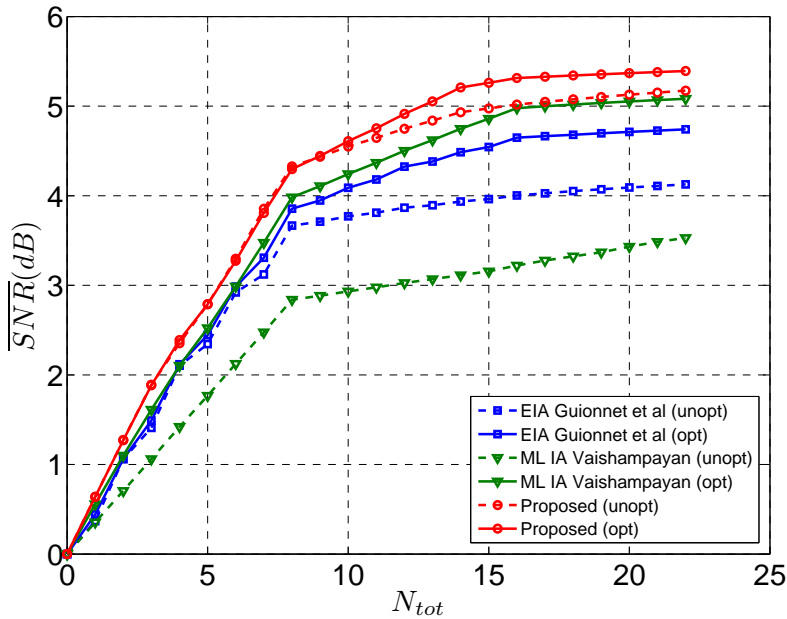
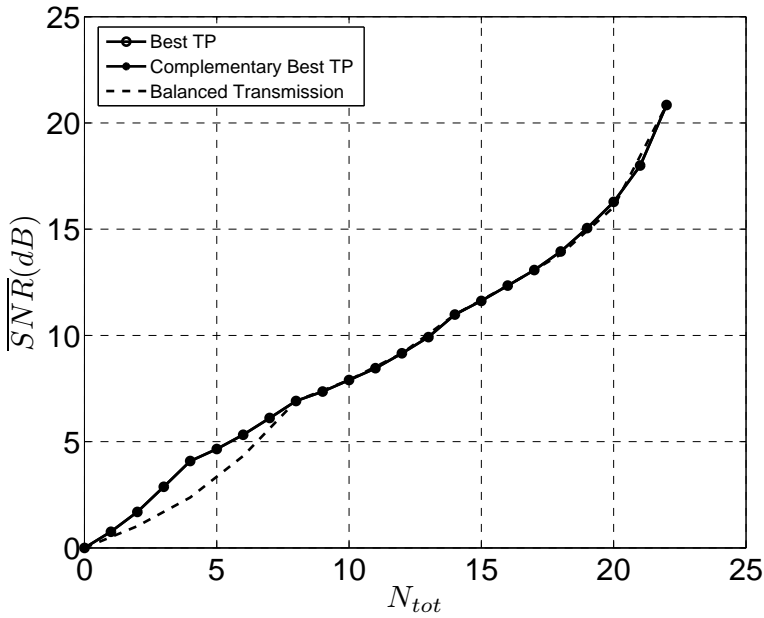
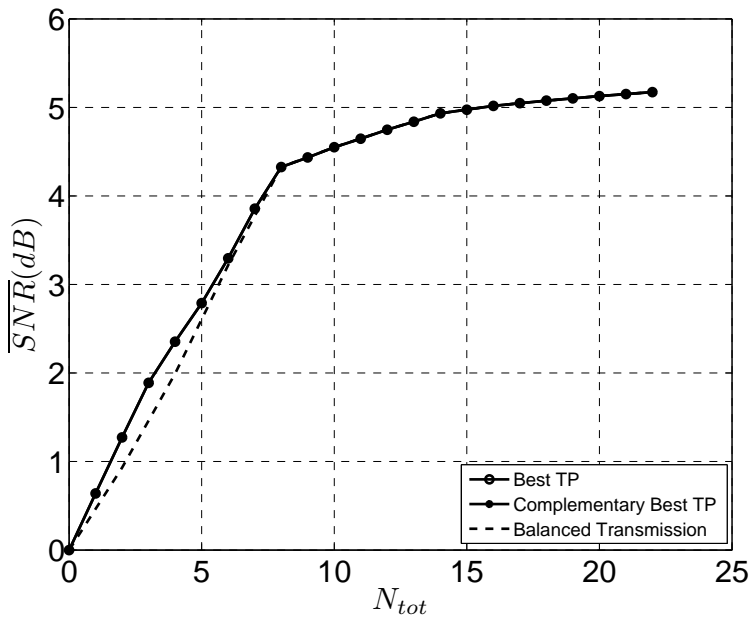


Figure 5.17: Performance comparison of the proposed disconnected-cell SSMSQ in an SMDC system, for an IID unit-variance Gaussian. Packet transmission is simulated for a packet loss channel with $p_L = 0.15$.

Figure 5.17 shows the \overline{SNR} versus the total number, N_{tot} , of packets transmitted from two descriptions for the three evaluated quantizers. Guionnet et al. denotes the EIA of [51]. The greedy packet scheduling, explained above, is employed to carry-out an unbalanced transmission of descriptions for a unit-variance Gaussian source.



(a)



(b)

Figure 5.18: Balanced versus unbalanced packet transmission using an entropy-coded SMDC system equipped with the proposed disconnected-cell SSMSQ: (a) zero loss channel, (b) loss channel with $p_L = 0.15$.

Notice that for a fair comparison, the mean performance between the two best TPs, corresponding to $N_1 \geq N_2$ and $N_2 \geq N_1$, is reported for the asymmetric EIA based quantizer of [51]. The results demonstrate that the proposed SSMDSQ quantizer performs significantly better compared to the other two assessed quantizers, that is the ML IA [113] and EIA [51] based quantizers. Similar results have been observed for a GG ($\alpha = 1.5$) and for a Laplacian source PDF. Also, the results confirm that the proposed optimization algorithm contributes notable gains in all three quantizers.

Figure 5.18 plots the \overline{SNR} with respect to the number of transmitted packets for the best TP (as defined for unbalanced transmission) and for the TP corresponding to the balanced transmission, for the un-optimized SSMDSQ-based SMDC system. As anticipated, due to the symmetry of the D-R function incurred by the proposed SSMDSQ, the two complementary best TPs, for unbalanced transmission, lie on top of each other. We notice that, for the zero packet loss channel state, a certain improvement in \overline{SNR} is provided by unbalanced versus balanced transmission. However, this performance difference diminishes as p_L increases – see Figure 5.18(b). This implies that, for large p_L , the best unbalanced TP is approximated by the TP determined by balanced transmission. This in turn motivates the employment of the proposed SSMDSQs, which can produce a scalable and perfectly balanced transmission of descriptions.

5.7 CONCLUSIONS

In contrast to contemporary asymmetric scalable MDSQ designs, this chapter introduced a novel SSMDSQ which can be employed to attain perfectly balanced packetized transmission of descriptions. In particular, we determined the sufficient conditions and we proposed novel EIA constructions to realize practical SSMDSQs. For generalized sources, the proposed SSMDSQs outperform existing asymmetric scalable MDSQ designs in terms of D-R performance.

In addition, we propose a novel generalization of the Lloyd-Max algorithm to carry out PDF specific iterative optimization of scalable MDSQs. Based on the optimized quantizers, a packet-based SMDC framework is established for transmission via packet erasure channels. In this context, it is shown that the symmetry of the D-R surface, incurred by the proposed SSMDSQs, can facilitate balanced transmission when the packet loss rate is unknown at the transmitter. Additionally, when an unbalanced transmission is considered, the proposed quantizers simplify the problem of defining the best transmission path with respect to asymmetric quantizers. For SMDC-based communication, simulation results for packetized transmission of GG sources over erasure channels clearly show the

advantage of the proposed SSMDSQs and verify the improvements brought by the proposed optimization strategy.

In order to highlight the application aspects of the proposed techniques, we employed the proposed SSMDSQ as the quantization module in a basic as well as in a full-blown image coding system. In this way, we evaluated the performance gains brought by the proposed quantization and optimization algorithms. The experimental evaluations confirmed that the presented symmetric quantization (together with the proposed optimization methodology) accounts for potential performance gains with respect to the contemporary schemes in the basic compression system. Moreover, in a full-blown image coding framework, SSMDSQs provide significant performance improvements against duplicated JPEG-2000.

5.8 APPENDIX

Pseudo code to realize a disconnected-cell symmetric EIA [93] is given as under:

| | |
|--|--|
| INITIALIZE IA matrix at level $p=1$ using $ML(m/2, k-1)$. Set TOG to FALSE | |
| <p>k even</p> <p>FOR $n = 1$ to N_p</p> <p> Split the n^{th} index into 2x2 sub-indices.</p> <p> IF $n = (N_p + 1)/2$:</p> <p> Perform L scan.</p> <p> ELSE-IF TOG=TRUE:</p> <p> Perform Z scan, TOG=FALSE.</p> <p> ELSE-IF TOG=FALSE:</p> <p> Perform \mathcal{V} scan, TOG=TRUE.</p> | <p>k odd</p> <p>FOR $n = 1$ to N_p</p> <p> Split the n^{th} index into 2x2 sub-indices.</p> <p> IF $n = (N_p + 1)/2$:</p> <p> Perform L scan.</p> <p> ELSE-IF $n \in \{d = 1\}$</p> <p> Perform LD scan.</p> <p> ELSE-IF $n \in \{d = 2k - 1\}$</p> <p> Perform UD scan.</p> <p> ELSE-IF TOG=TRUE:</p> <p> Perform Z scan, TOG=FALSE.</p> <p> ELSE-IF TOG=FALSE:</p> <p> Perform \mathcal{V} scan, TOG=TRUE.</p> |

Note: $d = 1$ denotes the lowest and $d = 2k - 1$ denotes the highest diagonal in $ML(m/2, k-1)$ IA matrix.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

This dissertation dealt with lossy source coding and lossy joint source-channel coding paradigms. In this context, we made an extensive investigation of quantization based methods for scalable single and scalable multiple description coding of real multimedia sources, such as 3-D mesh and photographic image data. In the following, we briefly describe the major contributions of this dissertation along with a comprehensive overview of the open issues and prospective future work.

Contribution 1

The first contribution of this dissertation lied in the design of scalable semi-regular 3-D mesh coding frameworks through a constructive design methodology – see Chapter 3. In this context, state-of-the-art wavelet-based mesh coding systems directly employ coding techniques, which were originally designed for wavelet-based scalable coding of images. However, in the literature, no theoretical justification is available for the utilization of image coding techniques for mesh compression. Neither was it formally established which coding techniques are superior to others, when considered for mesh coding. In our method, we followed a different approach to the design of scalable mesh coding systems. In particular, different modules of the mesh coding pipeline, e.g., embedded quantization, bit-plane coding, etc., were analyzed from a theoretical point of view in order to single-out the appropriate design choices, which can potentially provide the best compression performance. Based on the selected options, we proposed novel intraband and composite scalable mesh coding frameworks. For both non-normal and normal meshes, the proposed frameworks were shown to deliver better compression performance in comparison with the current state-of-the-art zero-tree based coding methods, which can be categorized as an interband systems. The

proposed codecs not only contributed higher compression performance, but also provided an increased scalability of the compressed semi-regular mesh. In this context, the particular nature of the bit-plane coding method, employed in the proposed systems, allowed for resolution scalable decoding of the encoded bit-stream. In particular, due to independent bit-plane coding of different resolution levels, decoding can be performed so as to reconstruct a particular resolution of the compressed mesh. This cannot be done using the current state-of-the-art interband systems due to joint coding of bit-planes belonging to different resolution subbands. We point out that the feature of resolution scalability is particularly important when the compressed mesh needs to be decoded on terminals with varying resolution requirements. The other type of scalability, namely, quality scalability, is provided by both the proposed and existing systems. All in all, we can say that the proposed frameworks extended over the existing schemes in terms of compression performance and scalability, thus, providing a new state-of-the-art in the area of wavelet-based semi-regular mesh compression.

The scripts of this contribution are made available to the scientific community as one journal publication [27], a book chapter [91] and three conference publications [25, 28, 90].

Contribution 2

The second contribution of this dissertation was related to the design of symmetric scalable multiple description scalar quantizers – see Chapter 5. We point out that contemporary scalable multiple description quantization designs are characterized by asymmetric D-R surfaces, which generally lead to roughly balanced or sometimes highly unbalanced transmission of descriptions. In contrast, the proposed quantizers ensure the symmetry of the operational D-R surface and hence, the derived descriptions are always perfectly balanced. We derived the sufficient conditions for a generic symmetric scalable multiple description quantizer and proposed both connected-cell high-redundancy and disconnected-cell medium-to-low redundancy quantizer instantiations. When evaluated against the equivalent contemporary asymmetric designs, the proposed instantiations exhibited superior D-R performance for the family of GG sources. Similar behavior was also observed when the proposed and contemporary quantizers were employed for scalable multiple description coding of wavelet decomposed images.

The scripts of contribution 2 are made available to the scientific community as a journal publication [93]. Parts of [93] with a more detailed experimental evaluations are also submitted to a conference – see publication 5 in the list of publications.

Contribution 3

The third contribution of this dissertation was the introduction of a novel source specific optimization framework to realize locally optimal scalable MDSQs. In practice, source specific optimizations for fixed-rate single description scalar quantizers are performed using the Lloyd-Max algorithm. The proposed optimization methodology is a generalized extension of the Lloyd-Max algorithm to the scalable multiple description setting, wherein any scalable multiple description quantization (symmetric or asymmetric) and with any number of descriptions, can be optimized. For a concise yet illustrative presentation, in this dissertation, we considered the simplest and the most preferred case of two-description quantization. Both the level-constrained and the entropy-constrained optimizations cases were considered. For SSMDSQ, the optimization did not disturb the symmetry property, i.e., the D-R surface of the quantizer remains symmetric after the optimization as well. When evaluated for the family of GG distributions, it was experimentally demonstrated that the optimization algorithm brings large optimization gains in the level-constrained case and relatively small but still notable gains in the entropy-constrained case. Furthermore, the introduced optimization gains translated into an enhanced average SNR when descriptions were transmitted via a packet loss channel with a certain packet loss probability. Similar conclusions were also retained when the proposed quantizer optimizations were integrated as a part of wavelet-based image coding system.

The scripts of this contribution are made available to the scientific community as a journal publication [93]. Parts of [93] are also submitted to a conference – see publication 6 in the list of publications.

6.2 OPEN ISSUES AND PROSPECTIVE WORK

Various open issues and new research avenues emerge from this dissertation, as outlined in the following.

In Chapter 3, we proposed an LM model to approximate the histogram of mesh wavelet coefficients. Additionally, we derived the D-R function of the Laplacian mixture PDF, when quantized using a general embedded dead-zone uniform quantizer. The proposed model was used to investigate the optimality of an embedded dead-zone uniform quantizer for scalable mesh coding. It was concluded that no single embedded quantizer is generally optimal for all subbands, and the optimal quantizer is specific to a particular choice of the model parameters. SAQ is generally an acceptable choice since it lies not far from the optimal and it can be easily implemented by bit-slicing of the produced finest level quantization indices.

In Chapter 3, mesh compression results were given by quantizing all wavelet subbands using SAQ. However, using the proposed LM model one can also find a specific optimal embedded uniform quantizer for each resolution subband, which may or may not be SAQ. Note that quantizing different resolution subbands with different embedded quantizers will not harm the quality or resolution scalable decoding property of the proposed mesh coding frameworks. On the other hand, ensuring optimality of embedded quantization for each resolution subband may improve the overall performance of the coding system in comparison with the one utilizing a near-optimal SAQ for all subbands. We would like to consider this aspect as a prospective extension of our work.

In the context of scalable MDSQs, one main issue is the non-availability of accurate D-R models for such quantizers. Such a model can allow for determining the optimal redundancy between source descriptions for a given packet loss rate on the channel. In general, quantizer's D-R models are based on a high-rate analysis, which makes use of high-rate assumptions [44]. Since a scalable quantizer operates over the rate range $[0, R_{\text{max}}]$, finding a single high-rate D-R model which can accurately predict the quantizer's D-R behavior over the complete rate range may be difficult. An alternative method of finding a good D-R model is the accurate parameterization of the operational D-R surface of the quantizer. In case of SSMDSQs the D-R surface is symmetric and it is sufficient to model $D(R_1, R_2)$, for $R_1 \geq R_2$ or $R_1 \leq R_2$. In [92], we proposed a parametric D-R model for high-redundancy EMDSQs, wherein, the experimental D-R surface was modeled as a symmetric decaying function. In future work, we will consider the design of parametric models for SSMDSQs which can closely model their D-R behavior over a complete rate/redundancy range.

In Chapter 3, the octree based intraband and composite mesh coding frameworks were shown to provide a significant gain in the compression performance with respect to existing schemes. In Chapter 5, SSMDSQs were proposed which, on the average, also yielded improved D-R performance with respect to the contemporary scalable MDSQ designs. A natural extension of this work is the scalable multiple description coding of semi-regular 3-D meshes, which arises from the combination of techniques developed in this dissertation. In particular, the mesh coding techniques developed in Chapter 3 can be combined with the efficient scalable MDSQ techniques developed in Chapter 5 to realize a SMDC framework for semi-regular 3-D meshes. We conjecture that the resulting SMDC would be highly performant simply due to the fact that the individual components demonstrate significantly improved performance with respect to the competing alternative choices. To the best of our knowledge, SMDC of meshes have not been treated

comprehensively in the literature and the only mentionable work in this context is the one of Bici et al [12]. In future, we aim to utilize the quantization based methods developed in Chapter 5 in order to perform SMDC coding of complex geometry 3-D meshes.

LIST OF PUBLICATIONS

ISI Journal Publications

1. Shahid M. Satti, Nikos Deligiannis, Adrian Munteanu, Peter Schelkens, and Jan Cornelis, "Symmetric Scalable Multiple Description Scalar Quantization", *IEEE Transactions on Signal Processing*, issue 60, vol.7, pp.3628-3643, July 2012.
2. Leon Denis, Shahid M. Satti, Adrian Munteanu, Jan Cornelis, and Peter Schelkens, "Scalable Intraband and Composite Coding of Semi-regular Meshes", *IEEE Transactions on Multimedia*, issue 8, vol.12, pp.773-789, December 2010.

Chapters in Books

3. Shahid M. Satti, Leon Denis, Ruxandra Florea, Jan Cornelis, Peter Schelkens, and Adrian Munteanu, "Optimized Scalable Wavelet-based Codec Designs for Semi-regular 3-D Meshes", *InTech - Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology*, Edited by: Dumitru Baleanu, pp. 567-592, ISBN 978-953-51-0494-0, 2012.
4. Shahid M. Satti, Leon Denis, Adrian Munteanu, Jan Cornelis, and Peter Schelkens, "Modeling Wavelet Coefficients for Wavelet Subdivision Transforms of 3-D Meshes", *Advance Concepts for Intelligent Vision Systems*, from Lecture Notes in Computer Science, issue 2010, vol. 6474, pp. 267-278, published by *SpringerLink*, December 2010.

Conference Publications with Peer Review

5. Shahid M. Satti, Nikos Deligiannis, Adrian Munteanu, Peter Schelkens, and Jan Cornelis, "An Optimization Algorithm for Scalable Multiple Description Scalar Quantizers", *International Symposium on Information Theory and its Applications (ISITA)*, October 2012, (Submitted).
6. Shahid M. Satti, Nikos Deligiannis, Adrian Munteanu, Peter Schelkens, and Jan Cornelis, "A Model-Based Analysis of Scalable Multiple Description Coding", *International Conference on Digital Signal Processing (DSP)*, pp.1-5, published by *IEEE*, July 2011.
7. Leon Denis, Shahid M. Satti, Adrian Munteanu, Jan Cornelis, and Peter Schelkens, "Context-Conditioned Composite Coding of 3-D Meshes based on Wavelets on Surfaces", *IEEE International Conference on Image Processing (ICIP)*, pp.3509 - 3512, published by *IEEE*, 2009.
8. Dan Cernea, Leon Denis, Shahid M. Satti, Adrian Munteanu, and Peter

- Schelkens, "Scalable Coding of Meshes", *Bell Labs Open Days*, Antwerp, Belgium, 2009.
9. Leon Denis, Shahid M. Satti, Adrian Munteanu, Jan Cornelis, and Peter Schelkens, "Fully Scalable Intraband Coding of Wavelet-decomposed 3-D Meshes ", *IEEE International Conference on Digital Signal Processing (DSP)*, Santorini, Greece, pp.573 - 577, published by *IEEE*, 2009.
10. Shahid M. Satti, Leon Denis, Adrian Munteanu, Jan Cornelis, and Peter Schelkens, "Estimation of Interband and Intraband Statistical Dependencies in Wavelet-based Decomposition of Meshes", *Proceedings of SPIE, Electronic Imaging*, vol. 7248, pp. 7248A1-7248A10, 2009.

REFERENCES

- [1] MPEG: ISO/IEC 14496(MPEG-4) Part 16: Animation Framework eXtension (AFX)", April 2003 (International Standard status).
- [2] "Advanced video coding for generic audiovisual services," pp. 1-564, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005, Version 4: Sept. 2005, Version 5 and Version 6: June 2006, Version 7: Apr. 2007, Version 8 (including SVC extension): approved in Nov. 2007.
- [3] "Digital compression and coding of continuous-tone still images," CCITT Study Group VII and JPEG of ISO/IEC JTC 1/SC 29/WG 10, document ISO/IEC 10918-1, 1993.
- [4] R. Ahlswede, "The rate-distortion region for multiple descriptions without excess rate," *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 721-726, November 1985.
- [5] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90-93, January 1974.
- [6] M. Alasti, K. Sayrafian-Pour, A. Ephremides, and N. Farvardin, "Multiple description coding in networks with congestion problem," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 891-902, March 2001.
- [7] R. Aarean, J. Kovacevic, and V.K. Goyal, "Multiple description perceptual audio coding with correlating transforms," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 2, pp. 140-145, March 2000.
- [8] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "MESH: Measuring Errors between Surfaces using the Hausdorff Distance," in *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 705-708, 2002.
- [9] I. V. Bajic and J. W. Woods, "Domain-based multiple description coding of images and video," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1211-1225, October 2003.
- [10] J. C. Batllo and V. A. Vaishampayan, "Asymptotic performance of multiple description transform codes," *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 703-707, March 1997.

- [11] T. Y. Berger-Wolf and E. M. Reingold, "Index assignment for multichannel communication under failure," *IEEE Transactions on Information Theory*, vol. 48, no. 10, pp. 2656-2668, October 2002.
- [12] M. O. Bici and G. Bozdagi Akar, "3D mesh coding using multiple description scalar quantizers," in *Proceedings of International Conference on Image Processing*. Atlanta, GA, USA, pp. 1-4, 2006.
- [13] N. V. Boulgouris, K. E. Zachariadis, A. Kanlis, and M. G. Strintzis, "Multiple description wavelet coding of layered video using optimal redundancy allocation," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1-19, 2006.
- [14] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532-540, April 1983.
- [15] J. Cardinal, "Multistage index assignments for M-description coding," in *Proceedings of IEEE International Conference on Image Processing*, vol. 3. Barcelona, Spain, pp. 249-252, September 2003.
- [16] E. Catmull and J. Clark, "Recursively generated B-spline surfaces on arbitrary topological surfaces," *Computer-Aided Design*, vol. 10, no. 6, pp. 350-355, November 1978.
- [17] G. M. Chaikin, "An algorithm for high-speed curve generation," *Computer Graphics and Image Processing*, vol. 3, no. 4, December 1974.
- [18] C.-L. Chang and B. Girod, "Direction-adaptive discrete wavelet transform via directional lifting and bandletization," in *Proceedings of IEEE International Conference on Image Processing*, Atlanta, GA, pp. 1149-1152, 2006.
- [19] D.-M. Chung and Y. Wang, "Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 6, pp. 895-908, September 1999.
- [20] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on Pure Applied Mathematics*, vol. 45, pp. 485-560, 1992.
- [21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications. New York, USA: Wiley-Interscience, 1991.
- [22] G. A. Darbellay and I. Vajda, "Estimation of the information by an adaptive partitioning of the observation space," *IEEE Transactions on Information Theory*, vol. 45, pp. 1315-1321, May 1999.
- [23] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, pp. 909-996, 1988.
- [24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from

- incomplete data via the EM algorithm," *The Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1-38, 1977.
- [25] L. Denis, S. M. Satti, A. Munteanu, J. Cornelis, and P. Schelkens, "Context-conditioned composite coding of 3D meshes based on wavelets on surfaces," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Cairo, Egypt, pp. 3509-3512, November 2009.
- [26] L. Denis, R. Florea, A. Munteanu, and P. Schelkens, "Spatially-adaptive bases in wavelet-based coding of semi-regular meshes," in *Proceedings of SPIE Photonics Europe. Optics, Photonics, and Digital Technologies for Multimedia Applications*, vol. 7723, pp. 772310-772310-8, 2010.
- [27] L. Denis, S. M. Satti, A. Munteanu, J. Cornelis, and P. Schelkens, "Scalable intraband and composite wavelet-based coding of semiregular meshes," *IEEE Transactions on Multimedia*, vol. 12, no. 8, pp. 773-789, December 2010.
- [28] L. Denis, S. M. Satti, A. Munteanu, J. Cornelis, and P. Schelkens, "Fully scalable intraband coding of wavelet-decomposed 3D meshes," in *Proceedings of 16th international conference on Digital Signal Processing*, Santorini, Greece, pp. 1-5, July 2009.
- [29] D. Doo, "A subdivision algorithm for smoothing down irregularly shaped polyhedrons," in *Proceedings of Interactive Techniques in Computer Aided Design*, pp. 157-165, 1978.
- [30] D. Doo and M. Sabin, "Behavior of recursive division surfaces near extraordinary points," *Computer-Aided Design*, vol. 10, no. 6, pp. 356-360, 1978.
- [31] N. Dyn, D. Levin, and J. A. Gregory, "A Butterfly subdivision scheme for surface interpolation with tension control," *ACM Transactions on Graphics*, vol. 9, no. 2, pp. 160-169, April 1990.
- [32] M. Eck, T. Derose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes," in *Proceedings of ACM SIGGRAPH, 22nd Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, California, USA, pp. 173-182, August 6-11, 1995.
- [33] M. Eck and H. Hoppe, "Automatic reconstruction of B-spline surfaces of arbitrary topological type," in *Proceedings of ACM SIGGRAPH, Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 325-334, 1996.
- [34] A. A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Transactions on Information Theory*, vol. 28, no. 6, pp. 851-857, November 1982.
- [35] W. Equitz and T. Cover, "Successive refinement of information," *IEEE*

-
- Transactions on Information Theory*, vol. 37, no. 2, pp. 269-275, March 1991.
- [36] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta Informatica*, vol. 4, no. 1, pp. 1-9, 1974.
- [37] A. Fraysse, B. Pesquet-Popescu, and J.-C. Pesquet, "Rate-distortion results for generalized gaussian distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Las Vegas, USA, pp. 3753-3756, 2008.
- [38] A. I. Gavrilescu, A. Munteanu, P. Schelkens, and J. Cornelis, "Embedded multiple description scalar quantizers," *IEE Electronics Letters*, vol. 39, no. 13, pp. 979-980, June 2003.
- [39] A. I. Gavrilescu, A. Munteanu, P. Schelkens, and J. Cornelis, "Generalization of embedded multiple description scalar quantisers," *IEE Electronics Letters*, vol. 41, no. 2, pp. 63-65, January 2005.
- [40] A. I. Gavrilescu, F. Verdicchio, A. Munteanu, I. Moerman, J. Cornelis, and P. Schelkens, "Scalable multiple-description image coding based on embedded quantization," *EURASIP Journal on Image and Video Processing*, vol. 2007, no. Article ID 81813, pp. 1-11, January 2007.
- [41] A. I. Gavrilescu, A. Munteanu, P. Schelkens, and J. Cornelis, "Embedded multiple description scalar quantizers for progressive image transmission," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* vol. 5. Hong Kong, pp. 736-739, April 2003.
- [42] A. I. Gavrilescu, A. Munteanu, P. Schelkens, and J. Cornelis, "High-redundancy embedded multiple-description scalar quantizers for robust communication over unreliable channels," in *Proceedings of International Workshop on Image Analysis for Multimedia Interactive Services*, vol. CD-version. Lisboa, Portugal, April 2004.
- [43] A. I. Gavrilescu, A. Munteanu, P. Schelkens, and J. Cornelis, "A new family of embedded multiple description scalar quantizers," in *Proceedings of IEEE International Conference on Image Processing*, vol. 1. Singapore, pp. 159-162, October 2004.
- [44] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.
- [45] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing " *IEEE Transactions on Information Theory*, vol. 14, no. 5, pp. 676-683, September 1968.
- [46] S. W. Golomb, "Run length encodings," *IEEE Transactions on Information Theory*, vol. 12, pp. 399-401, July 1966.
- [47] V. Goyal, J. Kelner, and J. Kovacevic, "Multiple description vector quantization with a coarse lattice," *IEEE Transactions on Information Theory*, vol. 48, no. 3, pp. 781-788, March 2002.

-
- [48] V. K. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74-93, September 2001.
- [49] V. K. Goyal, "Single and multiple description transform coding with bases and frames," *Philadelphia, PA: Soc. Industr. Appl. Math.*, 2001.
- [50] V. K. Goyal and J. Kovacevic, "Generalized multiple description coding with correlating transforms," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2199-2224, September 2001.
- [51] T. Guionnet, C. Guillemot, and S. Pateux, "Embedded multiple description coding for progressive image transmission over unreliable channels," in *Proceedings of IEEE International Conference on Image Processing*, vol. 1. Thessaloniki, Greece, pp. 94-97, October 2001.
- [52] I. Guskov, K. Vidimčec, W. Sweldens, and P. Schröder, "Normal meshes," in *ACM SIGGRAPH, International Conference on Computer Graphics and Interactive Techniques*. New Orleans, Louisiana, USA, pp. 95-102, 2000.
- [53] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, "Piecewise smooth surface reconstruction," in *Proceedings of SIGGRAPH'94*. New York, pp. 295-302, 1994.
- [54] S.-T. Hsiang and J. W. Woods, "Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, Geneva, CH, Vol. 3, pp. 662-665, May 28-31, 2000.
- [55] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the I.R.E.*, vol. 40, no. 9, pp. 1098-1102, September 1952.
- [56] W. Jiang and A. Ortega, "Multiple description coding via polyphase transform and selective quantization," in *Proceedings of SPIE International Conference on Visual Communication and Image Processing*. San Jose, USA, pp. 998-1008, 1999.
- [57] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *Proceedings of ACM SIGGRAPH, International Conference on Computer Graphics and Interactive Techniques*, New Orleans, Louisiana, USA, pp. 271-278, July 23-28, 2000.
- [58] A. Khodakovsky and I. Guskov, "Compression of Normal meshes," *Geometric Modeling for Scientific Visualization*, Springer-Verlag, Ed., 2002, pp. 189-206, 2002.
- [59] J. Li and C. C. J. Kuo, "Progressive coding of 3-D graphic models," in *Proceedings of the IEEE* vol. 86, no. 6, pp. 1052-1063, August 1998.
- [60] C. Lin, Y. Zhao, and C. Zhu, "Two-stage diversity-based multiple description image coding," *IEEE Signal Processing Letters*, vol. 15, pp. 837-840, 2008.

- [61] S. Lin and D. J. Costello, *Error control coding, fundamentals and applications*, 2nd ed: Pearson, Prentice Hall, 2004.
- [62] J. Liu and P. Moulin, "Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients," *IEEE Transaction on Image Processing*, vol. 10, no. 11, pp. 1647-1658, 2001.
- [63] J. Liu and P. Moulin, "Analysis of interscale and intrascale dependencies between image wavelet coefficients," in *Proceedings of International Conference on Image Processing*, Vancouver, BC , Canada pp. 669-672, September 2000.
- [64] M. Liu and C. Zhu, "Enhancing two-stage multiple description scalar quantization," *IEEE Signal Processing Letters*, vol. 16, no. 4, pp. 253-256, 2009.
- [65] S. P. Lloyd, "Least square quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129-137, March 1982. Also, unpublished memorandum, Bell Laboratories, 1957.
- [66] C. T. Loop, "Smooth subdivision surfaces based on triangles," Department of Mathematics, University of Utah, Master's Thesis, 1987.
- [67] M. Lounsbery, T. D. Deroose, and J. Warren, "Multiresolution analysis for surfaces of arbitrary topological type," *ACM Transactions on Graphics*, vol. 16, no. 1, pp. 34-73, 1997.
- [68] D. G. Luenberger, *Linear and non-linear programming*, second ed: Reading, MA: Addison Wesley, 1984.
- [69] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.
- [70] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7-12, March 1960.
- [71] A. E. Mohr, E. A. Riskin, and R. Ladner, "Generalized multiple description coding through unequal forward error correction" in *Proceedings of IEEE International Conference on Image Processing*, pp. 411-415, 1999.
- [72] A. E. Mohr, E. Riskin, and R. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction" *IEEE Journal on Selected Areas in Communications*, vol. 18, no. no. 6, pp. 819-828, June 2000.
- [73] A. Munteanu, J. Cornelis, and P. Cristea, "Wavelet-based lossless compression of coronary angiographic images," *IEEE Transactions on Medical Imaging*, vol. 18, no. no. 3, pp. 272-281, March 1999.
- [74] A. Munteanu, J. Cornelis, G. Van der Auwera, and P. Cristea, "Wavelet-based lossless compression scheme with progressive transmission capability,"

-
- International Journal of Imaging Systems and Technology, Special Issue on Image and Video Coding*, J. Robinson and R. D. Dony, Eds., vol. 10, no. 1, pp. 76-85, January 1999.
- [75] A. Munteanu, J. Cornelis, G. Van der Auwera, and P. Cristea, "Wavelet image compression - The Quadtree coding approach," *IEEE Transactions on Information Technology in Biomedicine*, vol. 3, no. 3, pp. 176-185, September 1999.
- [76] A. Munteanu, "Wavelet image coding and multiscale edge detection - algorithms and applications," Vrije Universiteit Brussel, Belgium, PhD Thesis, <http://www.etro.vub.ac.be/Personal/acmuntea/>, 2003.
- [77] D. Muresan and M. Effros, "Quantization as histogram segmentation: optimal scalar quantizer design in network systems," *IEEE Transactions on Information Theory*, vol. 54, no. 1, pp. 344-366, January 2008.
- [78] S. Nadarajah, "A generalized normal distribution," *Journal of Applied Statistics*, vol. 3, no. 7, pp. 685-694, September 2005.
- [79] B. M. Oliver, J. R. Pierce, and C. E. Shannon, "The Philosophy of PCM," *Proceeding of the IRE*, vol. 36, no. 11, pp. 1324-1331, November 1948.
- [80] L. Ozarow, "On a source coding problem with two channels and three receivers," *Bell Systems Technical Journal*, vol. 59, pp. 1909-1921, December 1980.
- [81] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *Proceedings of IEEE International Conference Network Protocols*. Redmond, WA, USA, pp. 16-27, November 2003.
- [82] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 79-93, January-March 2000.
- [83] F. Payan and M. Antonini, "Mean square error approximation for wavelet-based semiregular mesh compression," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 649-657, July 2006.
- [84] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," in *Proceedings of 33rd Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Vol. 2, pp. 342 - 346, October 1999.
- [85] R. Ronfard and J. Rossignac, "Full-range approximation of triangulated polyhedra," *Proceeding of Eurographics, Computer graphics forum*, vol. 15, no. 3, pp. C67-C76, August 1996.
- [86] J. Rossignac and P. Borrel, "Multi-resolution 3D approximation for rendering complex scenes," *Geometric Modeling in Computer Graphics*, B. Falcidieno and T. L. Kunii, Ed: Springer-Verlag, 1993, pp. 455-465, 1993.
- [87] A. Said and W. Pearlman, "A new fast and efficient image codec based on set-

- partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, June 1996.
- [88] U. Samarawickrama and J. Liang, "Low complexity M-channel multiple description coding with two-rate predictive coding and staggered quantization," in *Proceedings of IEEE International Conference on Image Processing*, San Diego, CA, pp. 2044 - 2047, 2008.
- [89] U. Samarawickrama, J. Liang, and C. Tian, "A three-layer scheme for M-channel multiple description image coding," *Elsevier Signal Processing*, vol. 91, no. 10, pp. 2277-2289, October 2011.
- [90] S. M. Satti, L. Denis, A. Munteanu, P. Schelkens, and J. Cornelis, "Estimation of interband and intraband statistical dependencies in wavelet-based decomposition of meshes," in *Proceedings of IS&T/SPIE*, San Jose, California pp. 72480A-72480A-10, 2009.
- [91] S. M. Satti, L. Denis, R. Florea, J. Cornelis, P. Schelkens, and A. Munteanu, "Optimized scalable wavelet-based codec designs for semi-regular 3D meshes," *Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology*, D. Baleanu, Ed. 567-592: InTech - 978-953-51-0494-0, 2012.
- [92] S. M. Satti, N. Deligiannis, A. Munteanu, P. Schelkens, and J. Cornelis, "A model-based analysis of scalable multiple description coding," in *Proceedings of IEEE Digital Signal Processing (DSP)*, pp. 1-5, July 2011.
- [93] S. M. Satti, N. Deligiannis, A. Munteanu, P. Schelkens, and J. Cornelis, "Symmetric scalable multiple description scalar quantization," *IEEE Transactions on Signal Processing* vol. 60, no. 7, pp. 3628-3643, July 2012.
- [94] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Transactions on Medical Imaging, Special issue on "Wavelets in Medical Imaging*, vol. 22, pp. 441-458, 2003.
- [95] P. Schröder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere," in *Proceedings of ACM SIGGRAPH, 22nd Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, California, USA, pp. 161-172, August 6-11, 1995.
- [96] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane, "Multiple description lattice vector quantization," in *Proceedings of Data Compression Conference*, pp. 13-22, 1999.
- [97] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt, "Multiple description wavelet based image coding," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 813-826, May 2000.

-
- [98] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, July and October 1948.
- [99] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *Institute of Radio Engineers, International Convention Record*, vol. 7, pp. 142-163, 1959.
- [100] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, December 1993.
- [101] W. F. Sheppard, "On the application of the theory of error to cases of normal distribution and normal correlation," *Philosophical Transactions of the Royal Society*, vol. A-192, pp. 101-167, 1899.
- [102] L. Shu, D. Costello, and M. Miller, "Automatic-repeat-request error-control schemes," *IEEE Communications Magazine*, vol. 22, no. 12, pp. 5-17, December 1984.
- [103] M. Soucy and D. Laurendeau, "Multiresolution surface modeling based on hierarchical triangulation," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 1-14, January 1996.
- [104] G. Strang and T. Nguyen: *Wavelets and filter banks*, Wellesley, MA: Wellesley-Cambridge Press, 1996.
- [105] G. Taubin, A. Guézic, W. Horn, A. William, and F. Lazarus, "Progressive forest split compression," in *Proceedings of ACM SIGGRAPH, International Conference on Computer Graphics and Interactive Techniques*, Orlando, Florida, USA, pp. 123 - 132, July 1998.
- [106] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Transactions on Graphics*, vol. 17, no. 2, pp. 84-115, April 1998.
- [107] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158-1170, July 2000.
- [108] D. S. Taubman and M. W. Marcellin: *JPEG2000: Image compression fundamental, standards and practice*, Dordrecht, Kluwer Academic Publishers, 2001.
- [109] C. Tian and S. S. Hemami, "A new class of multiple description scalar quantizer and Its application to image coding," *IEEE Signal Processing Letters*, vol. 12, no. 4, pp. 329-332, April 2005.
- [110] C. Tian and S. S. Hemami, "Universal multiple description scalar quantization: analysis and design," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 2089-2102, September 2004.
- [111] C. Tian and S. S. Hemami, "Sequential design of multiple description scalar quantizers," in *Proceedings of Data Compression Conference (DCC)*, August 2004,

- pp. 32 - 41.
- [112] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proceedings of Graphics Interface*, BC, Canada, pp. 26-34, June 18-20, 1998.
- [113] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 821-834, May 1993.
- [114] V. A. Vaishampayan and J. Domaszewicz, "Design of entropy-constrained multiple description scalar quantizers," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 245-250, January 1994.
- [115] V. A. Vaishampayan and S. John, "Balanced interframe multiple description video compression," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 3, pp. 812-816, 1999.
- [116] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto, "Multiple description vector quantization with lattice codebooks: design and analysis," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1718-1734, July 2001.
- [117] F. Verdicchio, A. Munteanu, A. Gavrilescu, J. Cornelis, and P. Schelkens, "Embedded multiple description coding of video," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3114-3130, September 2006.
- [118] M. Vetterli and C. Herley, "Wavelets and filter banks: theory and design," *Transactions on Signal Processing*, vol. 40, no. 9, pp. 2207-2232, September 1992.
- [119] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260 - 269, April 1967.
- [120] J. S. Vitter, "Design and analysis of dynamic Huffman codes," *Journal of the ACM*, vol. 34, no. 4, pp. 825-845, October 1987.
- [121] Y. Wang, M. T. Orchard, V. Vaishampayan, and A. R. Reibman, "Multiple description coding using pairwise correlating transforms," *IEEE Transactions on Image Processing*, vol. 10, no. 3, pp. 351-366, March 2001.
- [122] T. A. Welch, "A technique for high-performance data compression," *Computer-Aided Design*, vol. 17, no. 6, pp. 8-19, June 1984.
- [123] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 3, no. 6, pp. 520-540, June 1987.
- [124] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," in *Proceedings of Thirty-First Asilomar Conference on Signals, Systems & Computers*, Pacific Grove, CA, Vol. 2, pp. 1378-1382, November 2-5, 1997.
- [125] Z. Zhang and T. Berger, "New results in binary multiple descriptions," *IEEE Transactions on Information Theory*, vol. 33, no. 4, pp. 502-521, July 1987.

- [126] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337-343, May 1977.
- [127] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530-536, September 1978.
- [128] D. Zorin and P. Schroder, "Interactive multiresolution mesh editing," in *Proceedings of SIGGRAPH 97*, pp. 259-26, 1997.

