TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 709 • Publication 709

Andrey Norkin

Multiple Description Coding of Visual Information

Tampere 2007

Tampereen teknillinen yliopisto. Julkaisu 709
Tampere University of Technology. Publication 709

Andrey Norkin

Multiple Description Coding of Visual Information

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 4th of December 2007, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2007

ISBN 978-952-15-1894-2 (printed)
ISBN 978-952-15-1919-2 (PDF)
ISSN 1459-2045

Abstract

Nowadays, image and video compression are well developed fields of signal processing. Modern state-of-the-art coders allow better compression with better quality. A new field in signal processing is representation of 3D scenes. 3D visual scenes may be captured by stereoscopic or multi-view camera settings. The captured multi-view video can be compressed directly or converted to more abstract 3D data representations such as 3D dynamic meshes (mesh sequences) and efficiently compressed. In any case, efficiently compressed visual data has to be transmitted over communication channels, such as wireless channels or best-effort networks. This raises the problem of error protection, since most of these channels are error-prone.

A common approach to error protection is to consider it as a pure channel problem, separate from the source compression problem. This approach is based on Shannon's work, which states that in principle source and channel coding tasks can be carried out independently with no loss of efficiency. However, this cannot be achieved in practice due to delay requirements and other problems. As an alternative, one can tolerate channel losses. Assuming that not all the data sent has reached the decoder, one can concentrate on ensuring efficient decoding of the correctly received data only. One way to achieve this is to use *multiple description coding* (MDC). The source is encoded into several descriptions, which are sent to the decoder independently over different channels. The decoder can reconstruct the source with lower yet acceptable quality from *any* description received. Better reconstruction quality is obtained from more descriptions.

This thesis investigates MDC of images, video, stereoscopic video, and 3D meshes thus validating MDC as an error resilience tool for various types of multimedia data. The thesis consists of four main chapters.

Chapter 2 deals with MDC of images. It introduces an MDC algorithm

based on a 2-stage compression scheme employing B-spline-based image resizing which is used to split the image into the coarse and residual parts. The coarse part is included in both descriptions while the residual part is split into two parts. A bit allocation algorithm optimizes the scheme for a given bit budget and probability of description loss.

Chapter 3 addresses MDC of video. It presents a 3D-transform-based MD video coder targeted for mobile devices. The encoder has low computational complexity and the compressed video is robust to transmission errors. The chapter estimates the scheme's encoding complexity and introduces an optimization procedure which minimizes the expected reconstruction distortion subject to the packet loss rate.

In Chapter 4, MDC of stereoscopic video is addressed. Two MDC schemes are introduced, one based on spatial scaling and another based on temporal subsampling. A switching criterion makes it possible to decide, which scheme is more advantageous for the sequence being encoded.

Chapter 5 discusses MDC of 3D meshes. It introduces two MDC approaches for coding highly detailed 3D meshes. The schemes are able to produce multiple descriptions and are easily adaptable to changing packet loss rate and bit budget. The proposed D-R curve modeling significantly decreases computational load at the preparatory stage.

Acknowledgements

The work presented in this thesis has been carried out at the Institute of Signal Processing, Tampere University of Technology, Tampere, Finland.

First and foremost I wish to express my gratitude to Professor Jaakko Astola, my thesis supervisor, for encouraging me to work on multiple description coding problem, for the possibility to work at the Institute of Signal Processing, and for his expertise and advice throughout these years. I am deeply thankful to Professor Karen Egiazarian for his valuable guidance, support and encouragement during my work. I owe special thanks to Dr. Atanas Gotchev for the numerous fruitful discussions and recommendations, for his advice and help on the way to my degree.

I am grateful to Professor Edward J. Delp from Purdue University, USA, and Dr. Marius Tico from Nokia Research Center, Finland, for reviewing this thesis and for their valuable comments and recommendations.

I would like to express my sincere appreciation to Associate Professor Gözde Bozdagi Akar from Middle East Technical University (METU), Turkey, who has given me a chance to work within the Multimedia Research Group, METU, during one month period in 2006.

My warmest thanks are to my co-authors, among them M. Oguz Bici, Anil Aksay, and Cagdas Bilen for sharing their time and expertise during our joint research work in the 3DTV project. A debt of thanks is also to Atanas Boev, Michail Georgiev, and other members of the 3DTV team in TUT. I also wish to thank the members of the video compression team in TUT, particularly Jari Koivusaari and Dmytro Rusanovskyy. Many thanks to Vinod Malamal Vadakital for the discussions on video compression.

I would like to thank Professor Jarmo Takala for reviewing my Licentiate of Technology thesis, which parts have been used in this monograph.

I also owe this thesis to Professor Valeri Labunets who encouraged me to

pursue a doctoral degree in signal processing and to apply to the Institute of Signal Processing.

I am glad to express my sincere gratitude to all my colleagues at the Institute of Signal Processing for providing such a nice atmosphere. I would also like to thank the whole personnel of the Institute of Signal Processing. A debt of thanks is owed to Ms. Pirkko Ruotsalainen, coordinator of TICSP, and to Ms. Virve Larmila for their assistance in practical arrangements.

The financial support provided by the Tampere Graduate School in Information Science and Engineering (TISE) and the Academy of Finland: Finnish Centre of Excellence Programme is gratefully acknowledged.

I am thankful to all my friends in Tampere: Andrei, Alexandra, Roman, Daniel, Alessandro, Pilar, and the others for the cheerful atmosphere that I was surrounded.

I owe deep love and gratitude to my parents, Tatiana and Vladimir, for their unconditional support, understanding, and love throughout my life and to my grandfather Alexander for always believing in me. I would like to thank my younger brother Alexander for his interest in everything that I do and for keeping the connection even when being far away.

Finally, my warmest thanks go to Ekaterina Jarova, who has been beside me many years that I spent here in Tampere, for her love and patience, for sharing happy and sad moments of my life.

Tampere, November 2007
Andrey Norkin

Contents

List of Figures

List of Tables

Notations and Abbreviations

λ	Lagrangian multiplier
$\mathbb{R}$	Real space, i.e. the set of all real numbers
$\mathbb{R}^n$	n-dimensional real space, i.e. the space of all n-dimensional real vectors
ρ	Redundancy
σ_x^2	Variance of x
$\mathbf{x}^t$	Transpose of vector $\mathbf{x}$
$\widehat{x}$	Estimated value of x
$D(R)$	Distortion-rate function
$d(x, \hat{x})$	Single-letter distortion measure
D, d	Distortion
$E[X]$	The expected value of X
$L(x, \lambda)$	Lagrangian function
P	Probability
$p_X(x)$	Probability mass function of random variable x
Q	Quantization step, quantization factor
R	Rate
$R(D)$	Rate-distortion function
T	Transform
x'	Updated value of x
$X^\perp$	The orthogonal complement of X
3D	Three dimensional
3D VR DCT	3D Vector-Radix Decimation-in-frequency DCT

3D-DCT	Three-Dimensional DCT
AC	Transform coefficients different from DC, basis image is not flat
ATTEST	Advanced Three-dimensional Television System Technologies
AVC	Advanced Video Coding
B-spline	Biorthogonal spline
BFOS	Breiman, Friedman, Olshen, and Stone (algorithm)
bpp	bits per pixel
bpv	bits per vertex
CIF	Common Intermediate Format, 352×288 pixels
CPM	Compressed Progressive Mesh
CT	Correlating Transform
D-R	Distortion Rate
DC	First coefficient in most transforms, average value of the block
DCT	Discrete Cosine Transform
DIF	Decimation In Frequency
DPB	Decoded Picture Buffer
DPCM	Differential Pulse Code Modulation
DV	Disparity Vector
FEC	Forward Error Correction
GOB	Group Of Blocks
GOP	Group Of Pictures
H.26x	A family of video coding standards the domain of the ITU-T Video Coding Experts Group (VCEG)
HVS	Human Visual System
I	Intra-coded frame
i.i.d.	Independent Identically Distributed
JPEG	Joint Photographic coding Expert Group
kbps	kilobit per second
LOD	Level Of Details
LOT	Lapped Orthogonal Transform

MC	Motion Compensation / Motion Compensated
MD	Multiple Description
MD-FEC	MDC based on Forward Error Correction
MDC	Multiple Description Coding
MDLVQ	Multiple Description Lattice Vector Quantization
MDSQ	Multiple Description Scalar Quantization
MDTC	Multiple Description Transform Coding
MDVQ	Multiple Description Vector Quantization
ME	Motion Estimation
MPEG	Motion Pictures Expert Group, video compression standards
MSE	Mean Squared Error
MV	Motion Vector
MVC	Multi-view Video Coding
OBMC	Overlapped Block Motion Compensation
OSI	Open Systems Interconnection (model)
P	Predicted frame
PCT	Pairwise Correlating Transform
PGC	Progressive Geometry Compression
PLR	Packet Loss Rate
PM	Progressive Mesh
PSNR	Peak Signal to Noise Ratio
QCIF	Quater CIF, 176×144 pixels
QF	Quantized Frame
RD	Rate Distortion
RGB	color model: Red, Green, and Blue components
RLE	Run-length Encoding
RRD	Redundancy Rate Distortion
RS	Reed-Solomon (code)
SPIHT	Set Partitioning In Hierarchical Trees
TCP	Transmission Control Protocol

TG	Touma and Gotsman (coder)
ULP	Unequal Loss Protection
UMDC	Unbalanced Multiple Description Coding
VLC	Variable Length Coding
VRC	Video Redundancy Coding
WCT	Whitening Correlating Transform
YCbCr	color model: luminance (Y), blue chrominance (Cb), and red chrominance (Cr) components

Chapter 1

Introduction

1.1 Motivation for multiple description coding

Nowadays, image and video compression are well developed and widely used fields of signal processing. Modern state-of-the-art coders provide better compression with better quality. This is particularly true for video, as a video sequence possesses correlation in both spatial and temporal domains. The most efficient video compression schemes use motion estimation and compensation algorithms that exploit prediction to remove the correlation in temporal domain. The correlation in spatial domain is usually removed by methods similar to those of image compression.

A new field in signal processing is representation of 3D scenes. Recently, it has drawn significant attention of industry and research community. 3D visual scenes may be captured by stereoscopic or multi-view camera settings. The captured multi-view video can be compressed directly or converted to more abstract 3D data representation such as 3D dynamic meshes (mesh sequences) and efficiently compressed.

In any case, the efficiently compressed visual data has to be transmitted over communication channels, such as wireless channels or best-effort networks. This raises the problem of error protection, since most of these channels are error-prone. Compressed images and especially video sequences are vulnerable to transmission errors. If the error occurs in a video frame, it may propagate further into subsequent frames because of motion-compensated prediction. Contemporary multi-view video compression methods utilize comprehensive temporal and inter-view prediction structures and therefore channel

errors occurring in one view can propagate not only to the subsequent frames of the same view but also to the other views.

A common approach to error protection is to consider it as a pure channel problem, separate from the source compression problem. This approach is based on Shannon's work, which states that in principle the source and channel coding tasks can be carried out independently with no loss of efficiency. Following this approach, raw source sequences are processed in a way which reduces the data rate as much as possible. Reliable transmission of the bitstream to the receiver is provided then by a channel coder. The transport mechanism has to be perfect since a single error in the compressed bitstream might severely damage the reconstructed signal. According to the noisy channel coding theorem, a near-perfect transmission can be achieved if the data rate does not exceed the channel capacity. However, this can be hardly achieved in practice.

Widely used error-protection methods operate at the transport layer of the OSI model [68], e.g. the TCP protocol. There, error-free transmission is achieved by retransmitting packets that have been lost or corrupted. A problem with such a mechanism is that it causes delays and thus requires larger memory buffers. The delay is at least a packet round-trip time. A second problem arises when packet losses are caused by network congestions. Trying to retransmit lost packets generates extra data traffic and makes the network even more congested. Furthermore, retransmissions are virtually impossible in digital broadcasting. During broadcasting, loss of even a single packet may cause the transmitter to receive multiple retransmission requests, an effect called a *feedback implosion*.

Another approach for reliable transmission over lossy channels is so-called *forward error correction* (FEC). The compressed bitstream data is distributed between packets and protected by block channel codes. The data from the lost packets can be reconstructed from the received packets. The choice of the block code length is important. In terms of efficiency, long blocks are preferred since short blocks generate bitstreams with a relatively large number of additional symbols. The law of large numbers also dictates the choice of longer blocks since the number of errors is predicted more easily by long sequences. Just as with the previous approach based on retransmissions, a problem of delays and large memory buffers exists, this time caused by long blocks.

Two above-mentioned approaches tolerate no errors. They assume that all the data transmitted is correctly received. Consequently, they spend a considerable amount of resources to guarantee this. These expences grow with the amount of data to be transmitted. One may consider multi-view video as a case of such a growing amount of data compared to single-view video. Pure channel coding approaches might not be quite feasible in such cases. As an alternative, one can tolerate channel losses. Assuming that not all data sent has reached the decoder, one can concentrate on ensuring efficient decoding of the correctly received data. In this case, one needs to change the source coding accordingly and, more broadly, to consider the error protection problem as a *joint source-channel* problem.

Multiple description coding (MDC) is a coding approach for communicating a source over unreliable channel. The source is encoded into several descriptions, which are sent to the decoder independently over different channels. A certain amount of controlled redundancy is added to the compressed descriptions to make them useful even when received alone. The decoder can reconstruct the source from one description with low, yet acceptable quality. The more descriptions received, the higher is the reconstruction quality. Usually, the descriptions are *balanced*; that is, the descriptions are of equal rate and importance. In that case, the reconstruction quality depends on the number of received descriptions only and not on which particular descriptions are received.

The simplest MDC framework (Figure 1.1) considers communication over two independent channels. The encoder codes the source X into two redundant and mutually refining descriptions, which are sent separately over *Channel 1* and *Channel 2*. There are three decoders at the receiver side. The *central* decoder (*Decoder 0*) outputs an estimate $\hat{X}_0$ of the source X based on two received channels, while the *side* decoders (*Decoder 1* and *Decoder 2*) receive one channel each and output estimates $\hat{X}_1$ and $\hat{X}_2$ accordingly. The channels have two states: "on" (working) or "off" (failing). Then, the decoder receives information either from one or two channels, depending on their states.

When transmitted over a packet network, each description is placed in a separate packet protected with a block code. The packet in the receiver is considered either correctly received or lost. Increasing the number of descriptions increases the probability that at least one packet (i.e. one description) reaches the decoder. However, it also increases the coding redundancy and

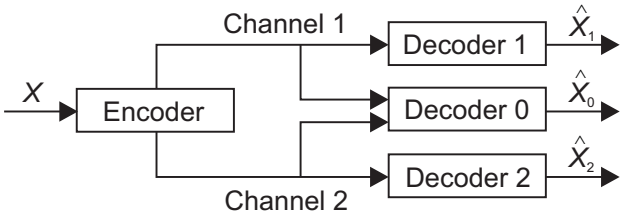


Figure 1.1: Scenario for MD coding with two channels and three receivers.

the decoder complexity.

In summary, MDC is an attractive coding approach as it provides reliable source reconstruction using only part of the data sent to the decoder and employs no priority network transmission mechanisms. MDC is especially advantageous in short-delay media streaming scenarios such as video conferencing and when broadcast over unreliable channels where it provides acceptable reconstruction quality and prevents feedback implosions in case of numerous packet losses.

1.2 Overview of the thesis

This thesis consists of four parts, which describe MDC of images, video, stereoscopic video and 3D meshes. In this way, the reader can follow the development of MDC and its applications, from less complex data (as two-dimensional images) to more complex (like 3D meshes).

Chapter 2 deals with MDC of images. First, it surveys MDC approaches to coding of random variables and images. Next, it presents an MDC algorithm, which is based on the 2-stage compression scheme employing B-spline-based image resizing in order to split the image into the coarse image and the residual. Then, we provide a scheme analysis and present a bit allocation algorithm, which optimizes the scheme for a given bit budget and probability of description loss.

Chapter 3 addresses MDC of video. Along with a survey of MD video coding methods, it presents a 3D-transform-based MD video coder targeted for mobile devices. The encoder has low computational complexity and high

robustness to transmission errors. The proposed scheme is an adaptation to video coding of the 2-stage method described in Chapter 2. The chapter also estimates the encoding complexity of the scheme and provides an algorithm for optimal bit allocation in case of varying description loss rate.

In Chapter 4, we address a new problem of MDC of stereoscopic video. To our knowledge, no previous research has been done on this topic. We propose two prediction schemes which are capable of creating two descriptions. Later, we state a switching criterion that makes it possible to decide, which scheme to apply for the sequence being encoded. It is possible to switch between the schemes "on the fly".

Chapter 5 discusses MDC of 3D meshes. First, we survey approaches to 3D mesh compression including MDC of 3D-meshes. Then, we describe in detail two MD approaches for coding highly detailed 3D meshes. The schemes are able to generate multiple descriptions and are easily adaptable to the changing packet loss rate and the total bit budget. The chapter also introduces D-R curve modeling, which significantly decreases the computational load at the preparatory stage.

1.3 Author's contributions

The author's contributions to multiple description coding are mostly reflected in the following publications, denoted as P1, P2, ..., P8. The author has accomplished the publication tasks with the co-authors and supervisor of the thesis. The co-authors have seen this description and agree with the author.

P1 A. Norkin, A. Gotchev, K. Egiazarian, and J. Astola, "Multiple description image coders using whitening transforms: analysis and comparative study", in *Proc. TICSP Workshop SMMSP'2004*, Vienna, Austria , pp. 225–232, Sept. 2004.

P2 A. Norkin, A. Gotchev, K. Egiazarian, and J. Astola, "Two-stage multiple description image coders: analysis and comparative study", *Signal Processing: Image Communication*, vol. 21/8, pp. 609–625, Sept, 2006.

P3 A. Norkin, A. Gotchev, K. Egiazarian, and J. Astola, "A low-complexity multiple description video coder based on 3D-transforms", in *Proc. EUSIPCO 2006*, Florence, Italy , Sept. 2006.

P4 A. Norkin, A. Gotchev, K. Egiazarian, and J. Astola, "Low-complexity multiple description coding of video based on 3D block transforms", *EURASIP Journal on Embedded Systems*, vol. 2007, Article ID 38631, 11 pages, 2007. doi:10.1155/2007/38631.

P5 A. Norkin, A. Aksay, C. Bilen, G. Bozdagi Akar, A. Gotchev and J. Astola, "Schemes for multiple description coding of stereoscopic video", in *Proc. MRCS 2006*, Istanbul, Turkey. *LNCS, Multimedia Content Representation, Classification and Security*, vol. 4105, pp. 730–737, Sept. 2006.

P6 A. Norkin, M. O. Bici, G. Bozdagi Akar, A. Gotchev, and J. Astola "Wavelet-based multiple description coding of 3-D geometry", *VCIP2007*, in *Proc. SPIE*, vol. 6508, pp. 65082I, 10 pages, Jan. 2007.

P7 M. O. Bici, A. Norkin, G. Bozdagi Akar, A. Gotchev and J. Astola, "Multiple description coding of 3D geometry with forward error correction codes", in *Proc. 3DTV-CON2007*, Kos, Greece, May 2007.

P8 A. Norkin, M. O. Bici, A. Aksay, C. Bilen, A. Gotchev, G. Bozdagi Akar, K. Egiazarian, and J. Astola, "Multiple description coding and its relevance to 3DTV", chapter in edited book "Three-dimensional television: capture, transmission, and display", Editors: H. M. Ozaktas and L. Onural, Springer, Heidelberg, 2007.

Publication P1 addresses MDC of images. The author has proposed and developed a modified MDC scheme based on whitening transform. The author wrote the necessary program code and performed all the simulations showing the efficiency of the proposed approach.

Publication P2 extends the results of P1. There, the author has done the performance analysis of the 2-stage scheme and derived a sub-optimal bit allocation formula based on the given probability of the description loss and the total bit budget. The author has studied the effect of using DCT and LOT for residual image coding. For DCT, the author has proposed a postprocessing algorithm for the side reconstruction images. The author has done the programming and simulations and written the publication.

The subject of publication P3 is MDC of video. The research has been done in the joint VCMode project between the Transforms and Spectral Techniques Group and the research group of Prof. Jarmo Takala from the Institute of

Digital and Computer Systems, TUT. The author of the thesis has adapted the method described in P1 and P2 to coding of video. A simple 3D-DCT coder has been obtained from VCMode project and modified by the author of the thesis to the 2-stage encoding scheme providing MDC capabilities. The author has performed the complexity estimation of the encoder and done all the experiments and writing.

Publication P4 continues the research on MDC of video published in P3. The author has done the necessary derivations and performed optimization of the scheme for varying channel conditions. The author has also performed simulations of video transmission over the network with packet losses and has written the publication.

Publication P5 is a result of the joint work between the Institute of Signal Processing of Tampere University of Technology (TUT), Finland and Multimedia Research Group (MMRG) of Middle East Technical University (METU), Turkey in the framework of 3DTV Network of Excellence. The author has proposed two MDC schemes for stereoscopic video. A multi-view video coder has been provided by MMRG group and necessary modifications of the software have been done by Anil Aksay and Cagdas Bilen from MMRG. The author has also proposed a criterion to switch between these schemes and done most experimental and writing work in this publication.

Publication P6 is a joint work between Institute of Signal Processing, TUT and MMRG, METU. The works is a result of collaboration between the author of the thesis and M. Oguz Bici, METU. The author has proposed to apply the scheme with duplication of wavelet coefficients to MDC of 3D meshes. The author has implemented a bit allocation algorithm for multiple descriptions, has proposed and realized D-R function modeling, done part of simulations and main part of the writing work. The realization of SPIHT algorithm for 3D meshes and some other parts of the software have been done by M. Oguz Bici.

Publication P7 is also a result of joint work between the author and M. Oguz Bici. It presents the MDC scheme for 3D meshes based on MD-FEC. In P7, the author has introduced D-R function modeling, has done part of experimental work, and has been heavily involved in preparing the publication.

Publication P8 surveys MDC algorithms and their application to images, video, stereoscopic video, and 3D meshes. The author has prepared and writ-

ten the sections on MDC rate-distortion region, MDC of images, MDC of video, introduction, and conclusions and has completed the final text assembling and editing.

Chapter 2

Multiple Description Coding of Images

A typical transform-based image coder consists of the following blocks: a *transform*, which maps the spatial domain image representation to the transform domain for better decorrelation; a *quantizer*, which scans and quantizes the transform coefficients to achieve lossy compression; and an *entropy coder*, which removes the remaining statistical redundancy between quantized coefficients in a lossless manner. When an MDC module is to be added to such a scheme to achieve channel error protection, a first and crucial problem is where to include it. MDC methods utilize various possibilities for adding controllable redundancy. Some employ subsampling in spatial or transform domain, while others employ transforms to create interleaved patterns to be distributed among descriptions or special scanning and quantization of transform coefficients [100].

This chapter is organized as follows. In Section 2.1, we review the most common MDC approaches and practical existing schemes. Rather than being very comprehensive, this survey introduces principles of MDC of images and describes state-of-the-art MDC schemes. In Section 2.2, we introduce a two-stage MD image coder. Next two sections present the details about each of two stages: Section 2.3 addresses the coarse approximation coding stage while Section 2.4 deals with the residual image coding stage. Section 2.5 provides the analysis of the proposed scheme and introduces a bit allocation algorithm. Section 2.6 presents the numerical results and comparisons with other MDC methods, and Section 2.7 concludes the chapter.

2.1 Survey of multiple description image coding approaches

2.1.1 Multiple description scalar quantization

A simple way to add error-protecting redundancy to the compressed image bitstream is to do it at the quantization stage, i.e. the stage where loss of insignificant information occurs. This idea has been extensively developed in the works of Vaishampayan, who has suggested a *theory of multiple description scalar quantizers* addressing cases of fixed-rate quantization [145] and entropy-constrained quantization [148].

Multiple description scalar quantization (MDSQ) works as follows. Two side (coarse) quantizers with overlapping cells operate in parallel at the quantization stage. The quantized source can be reconstructed from the output of either quantizer with lower quality. When the outputs of two quantizers are combined, they produce higher quality reconstruction due to the resulting smaller quantization cells. In a practical scheme, the encoder first applies a regular scalar quantizer, mapping the input variable x to a quantization index I [145]. Then, an *index assignment* is done, mapping each index I to a codeword index pair (i, j) in a codebook.

Figure 2.1 (a) [145] presents the index assignment matrix for the case of "staggered" index assignment. The cells of the quantizer corresponding to the index I are numbered in the matrix from 0 to 14. The row and column indices of the index assignment matrix form the index pair (i, j). Index i is included in *Description 1* whereas index j is included in *Description 2*. The central decoder reconstructs the exact value of index I and the corresponding value $\hat{X}_0$. The side decoders estimate X as the expected value when one of the indices is fixed. Thus, the quality of the side reconstruction is determined by the number of diagonals in the index assignment matrix. In Figure 2.1 (a), only 15 out of 64 cells in the index assignment matrix are occupied. Unoccupied cells constitute coding redundancy. Figure 2.1 (b) shows the index assignment with three diagonals filled and lower redundancy.

The high-rate analysis of MDSQ has been presented in [147]. The performance of MDSQ has been compared to Ozarow's rate-distortion bound [107] for squared error distortion measure and the memoryless Gaussian source and a 3.07 dB gap between the product of the average central and side distor-

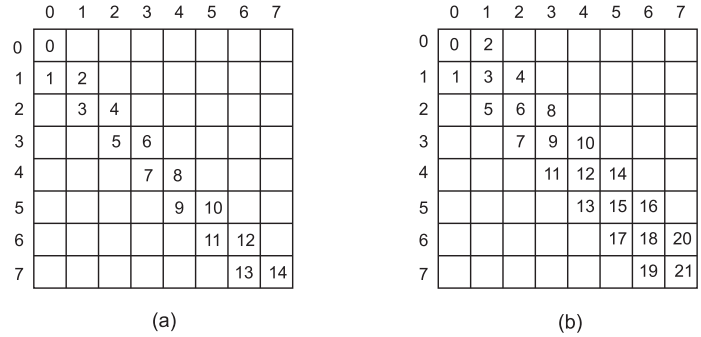


Figure 2.1: Index assignment: (a) Staggered quantization cells; (b) Higher spread quantization cells.

tions $d_0 d_1$ has been identified [147]. It has been conjectured that this gap is caused by the non-spherical form of quantization cells [130]. Therefore, the gap can be closed by constructing quantizers with more "spherical"-like cells; that is, cells with a smaller normalized second moment than a hypercube. Several solutions have been proposed, including trellis-coded quantization [76] and *multiple description lattice vector quantizer* (MDLVQ) [150]. It has been shown [150] that MD vector quantizers are capable of closing the 3.07 dB gap when the vector dimensions tend to infinity ($N \to \infty$). An improvement of 0.3 dB is achieved in the two-dimensional case using an MD hexagonal lattice quantizer [130].

Originally, MDLVQ was limited to the balanced case (equal rates $R_1 = R_2$ and equal distortions $D_1 = D_2$). Diggavi et al. [38] generalized this method to *asymmetric multiple description vector quantizers* that cover the entire spectrum of the distortion profile, ranging from balanced to successively refinable descriptions. The improvements for MDLVQ provide more operating points and more flexible rate-distortion trade-off by the slight increase in complexity [55,81]. A generalized MDVQ for more than two descriptions has also been designed [47].

One of the first multiple description image coders was proposed by Vaishampayan [146]. Multiple description scalar quantization is applied to DCT coefficients of JPEG coder. The indices obtained are entropy-coded separately in both descriptions. The descriptions are sent to the destination in different packets. An MD image coder based on MDSQ has been proposed by Servetto et al. [131, 132]. In this coder, MDSQ is applied to each coefficient of the

wavelet transform. Thus, two descriptions of the wavelet coefficients are created. Each description is then coded independently with a single description coder (e.g. SPIHT). Different MD scalar quantizers are applied to different subbands to achieve better redundancy allocation.

2.1.2 Multiple description transform coding

Another MD approach considers adding redundancy immediately after the stage of transform coding by means of so-called *pairwise correlating transform* (PCT) [154]. The general framework is the following. First, the input signal is decorrelated using a proper transform (e.g. DCT). The resulting coefficients are ordered according to their variances and coupled into pairs. These pairs undergo the correlating transform, i.e. two uncorrelated coefficients at the PCT input give rise to two correlated coefficients at the PCT output. One transform coefficient is sent to *Description 1* and the other is sent to *Description 2*. By the explicitly added redundancy within the pairs of coefficients, a lost coefficient from the pair can be estimated from the received one. When both descriptions are received, the exact values of variables can be determined by taking the inverse transform. This method has been extended to more general orthogonal [106] and non-orthogonal transforms [152, 155].

Consider at the input two independent Gaussian random variables A and B with variances σ_a^2 and σ_b^2, respectively, $\sigma_a^2 > \sigma_b^2$ [106]. The output random variables C and D with variances σ_c^2 and σ_d^2, respectively, are related to A and B by a unitary matrix T

$$[C\ D]^t = T[A\ B]^t. \tag{2.1}$$

The transform T controls the redundancy by varying the correlation between C and D. For example, T can be parametrized by an angular parameter. The example of the orthogonal transform is [106]

$$T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{2.2}$$

and the optimal non-orthogonal transform is [155]

$$T = \begin{bmatrix} \sqrt{\frac{\cot\theta}{2}} & \sqrt{\frac{\tan\theta}{2}} \\ -\sqrt{\frac{\cot\theta}{2}} & \sqrt{\frac{\tan\theta}{2}} \end{bmatrix} \tag{2.3}$$

where the parameter θ relates to the amount of correlation introduced by T. After being correlated, the outputs C and D are quantized and entropy-coded.

The non-orthogonal transform is generally more efficient than the orthogonal transform as it can work on a larger redundancy interval. Further improvement of non-orthogonal transforms has been introduced in [155,156]. The basis vectors of the optimal non-orthogonal transform have the same length and are rotated with the same angle in opposite directions from the axis corresponding to the variable with higher variance. The suggested non-orthogonal transform outperforms the orthogonal transform. The orthogonal and non-orthogonal transforms yield the same results only in two points: with zero redundancy and maximum redundancy for orthogonal pairing [106].

A similar correlating transform has been obtained independently by Goyal et al. [56–58]. Generalizing the work of Orchard et al. [106], a transform-based approach was developed for producing M descriptions of an N-tuple source. Several transform optimization results were presented for memoryless Gaussian sources, including a complete solution for the $N = 2$, $M = 2$ case with arbitrary weighting of the descriptions.

An important tool for evaluating performance of MD codes is the *redundancy rate-distortion* (RRD) function [106]. The redundancy is determined as $\rho = R - R^*$. Here, R is the resulting rate of the MD coder with the central channel distortion D_0, and R^* is the rate of the best single description coder for the given distortion D_0. Thus, ρ is the additional bitrate needed for one-channel reconstruction. The RRD function is then defined as $\rho(D_1; D_0)$, where D_1 is the averaged one-channel distortion. Usually, ρ depends weakly on D_0. Hence, the RRD function could be written as $\rho(D_1)$ [106].

The performance of the JPEG-like MD coder using correlating transforms was compared with the performance of the coder using MDSQ [156]. It was shown that PCT exhibits very good performance in the small redundancy region but fails to achieve good reconstruction quality for higher redundancies. Poor performance for high redundancies is due to the fact that the RRD curve of correlating transform converges to a nonzero value $\sigma_B^2/2$ [156] (in a region with redundancy close to zero it shows super exponential decay). The performance of multiple description transform coding (MDTC) for higher redundancies has been improved in Generalized MDTC (GMDTC) proposed by Wang et al. [155, 158]. This scheme uses the explicit redundancy to correct

the error resulting from reconstruction from a single description. For higher redundancies, GMDTC includes in the description carrying the coefficient C some information $C^{\perp}$, an orthogonal complement of C in the Hilbert space spanned by (C, D) [155]. This orthogonal complement information is the pure redundancy and is used only in case of reconstruction from a single description. If total redundancy is below a critical point ρ^*, all the redundancy is allocated to the transform. Thus, this hybrid scheme combines super-exponential decay in the low redundancy region and near-exponential decay in the higher redundancy region.

2.1.3 MD-FEC

A general approach is to use FEC codes to create multiple descriptions [3, 95]. Its basic idea is to assign unequal numbers of FEC symbols to different parts of the compressed bitstream, depending on the importance of these parts and their contribution to the overall reconstruction quality. This idea is best applied to so-called "progressive" bitstreams, where the bytes of the compressed source are ordered according to their importance. The wavelet-based SPIHT encoder is an example of compression algorithm generating an *embedded* bitstream [123]. Its first bytes are the more important ones and any subsequent byte refines the decoded image. Thus, the bitstream can be truncated based on the given bit budget leaving the reconstruction still possible. In connection with FEC, the first bytes should be better protected than the later bytes.

We illustrate how MD-FEC works by means of an example. Seventeen data symbols are coded using eight FEC symbols; thus, a total of 25 symbols is to be transmitted, with these broken into five codes, as shown in Table 2.1. FEC is implemented by means of Reed-Solomon (RS) codes. Stronger RS codes are applied to the data located at the beginning of the bitstream, i.e. to the more important data. Namely, (5,2)-codes are applied to symbols 1 and 2, (5,3)-codes are applied to symbols 3 to 8, (5,4)-codes are applied to symbols 9 to 12, and symbols 13 to 17 are left unprotected. Then, the symbols including the FEC ones are grouped vertically into multiple descriptions (packets). Each packet is protected with a parity code enabling error detection and sent to the receiver.

At the receiver side, the decoder detects erroneous descriptions and uses RS codes to reconstruct the lost data. As (5,2) RS code can sustain a loss

	D_1	D_2	D_3	D_4	D_5
Code 1	1	2	FEC	FEC	FEC
Code 2	3	4	5	FEC	FEC
Code 3	6	7	8	FEC	FEC
Code 4	9	10	11	12	FEC
Code 5	13	14	15	16	17

Table 2.1: Example of MD-FEC. In this example, D_j means *Description* j, and Code i means RS code (n, k_i) with $n = 5$ and k_i equal to the number of non-FEC symbols in the i-th row.

of three symbols. Receiving any two descriptions makes it possible to decode symbols 1 and 2. Similarly, receiving any three descriptions makes it possible to decode symbols 1 to 8. When no descriptions are lost, all 17 symbols are reconstructed. For reconstruction with RS codes, it does not matter which descriptions are lost, only the number of lost descriptions matters. Thus, MD-FEC generates inherently balanced descriptions (having the same size and resulting in the same distortion when lost).

MD-FEC is an attractive technique as it can be applied to any coder which generates the embedded bitstream. The method can also be applied to the non-embedded bitstream, after parts of the bitstream are rearranged in order of importance.

2.1.4 Other MD coding approaches

An MDC approach developed by Goyal et al. exploits quantized overcomplete frame expansions for generalized multiple description coding [59, 60]. The authors propose to use linear transforms from $\mathbb{R}^N$ to $\mathbb{R}^M$ with $M > N$ followed by scalar quantization, in the *quantized frames* (QF) system. Each quantized coefficient may be considered as a description. This approach is similar to block channel codes with swapped order of the transform and quantization operations [60]. Due to this interchanged order, the MDC system behaves differently from the conventional block channel code in the presence of erasures. A channel-code system with (M, N)-code would not show any degradation of reconstruction quality until the number of erasures exceeded $N - M$. If the number of erasures exceeds $M - N$, a so-called "cliff-effect" characterized by fast loss of reconstruction quality appears. In contrast, the QF system shows mild degradation of quality when the number of erasures is higher than $M-N$.

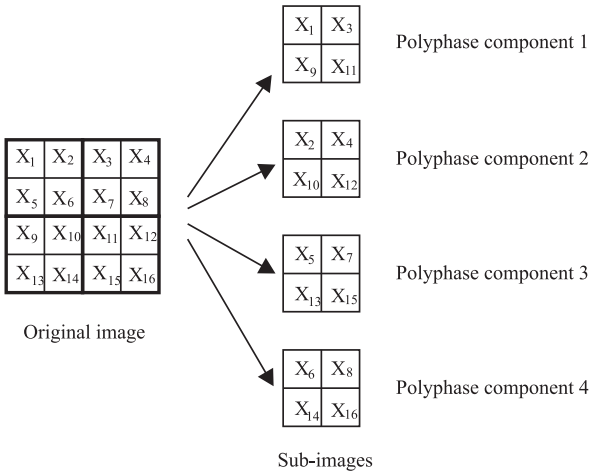


Figure 2.2: Example of polyphase transform when original image of size 4×4 is partitioned into 4 blocks [122].

Each transform coefficient brings some independent information, even those in excess of N coefficients [54]. Moreover, frame expansions have "quantization noise reduction" property [61], yielding better quality than conventional block channel coding when all the descriptions are received.

Several MDC schemes are based on spatial image subsampling, exploiting the fact that natural images possess a high degree of spatial correlation [49, 122]. Subsampling in 2D space may lead to different polyphase components, depending on the geometry of subsampling lattice applied. Figure 2.2 shows a subsampling scheme generating four polyphase components.

Polyphase subsampling is an attractive choice since it can be adjusted for an arbitrary number of descriptions making the coder easily tunable for changing channel conditions. At the receiver, pixels corresponding to the lost descriptions can be interpolated from the received neighbouring pixels. The redundancy is due to the spatial subsampling which decreases the interdependencies between the neighboring pixels thus making the compression less efficient. Additional redundancy can be added in the form of redundant copies of polyphase components [122]. In that case, each description contains a fully-coded copy of one polyphase component and several redundant copies of other polyphase components coded at lower rates. If the main copy of the polyphase component is lost, the decoder uses the available highest rate redundant copy of the same component included in the other description. An optimized bit al-

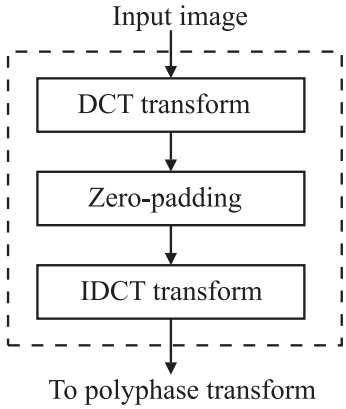


Figure 2.3: Preprocessing block [49].

location algorithm chooses the number of redundant copies and their bitrates for each polyphase component [122].

The redundancy can also be added at the preprocessing stage before the polyphase transform [49]. Such pre-processing procedure is shown in Figure 2.3. The input image is DCT-transformed to the array of size $D \times D$, which is padded with zeros to the size of $(D \times M)(D \times M)$. The obtained $(D \times M)(D \times M)$ representation is transformed back to the spatial domain, where it is split into multiple descriptions by polyphase subsampling. Clearly, the redundancy can be adjusted by changing the padding parameter M.

Polyphase subsampling can be done in the transform domain as well. In wavelet domain, such schemes are based on inter and intra-scale dependencies between wavelet coefficients [10, 78]. For *lapped orthogonal transforms* (LOT), such schemes have been generated by interleaving blocks of transform coefficients [30–32]. LOT blocks of transform coefficients are split between two descriptions in such a way that neighboring blocks are included in different descriptions. If a description is lost, missing coefficients are replaced with zeros, resulting in spatial interpolation due to overlapping reconstruction functions. To control the amount of redundancy, different LOT bases have been developed [32]. These bases trade off the coding gain against the reconstruction gain.

Miguel et al. have proposed a SPIHT-based MD image coder [94]. The coder can produce a flexible number of descriptions N. Spatially disperse wavelet coefficient trees are grouped into N sets. Each tree set is independently

coded with SPIHT algorithm. Each description consists of one tree set coded at a higher rate and $M-1$, $M \leq N$ redundant tree set copies coded at lower rates. Thus, the redundancy is formed by duplicating the coefficient trees. If the description is lost, the missing trees are obtained from the received redundant copy that has the highest rate [94]. Bit allocation is optimized for the target bitrate and the probability of description loss.

An MD coder producing two balanced descriptions compatible with JPEG 2000 [75] is proposed by Tillo et al. [142]. The algorithm generates two JPEG 2000 streams coded at rates R_1 and R_2 such that $R_1 > R_2$. Rate allocation in JPEG 2000 is based on code-block (CB) truncation, so that *Stream 2* has CB's truncated at a lower rate than CB's from *Stream 1*. To generate balanced descriptions, CB's from *Stream 1* are mixed with CB's from *Stream 2*, yielding two descriptions with the rate approximately equal to $(R_1 + R_2)/2$.

2.2 Two-stage multiple description coding

Multiple description transform coding [156] has been described in Section 2.1.2. After a decorrelating block transform, variances of the transform coefficients are estimated along blocks. Coefficients with higher variances are paired with coefficients with lower variances and undergo a pair-wise correlating transform (PCT). This yields two cross-correlated descriptions. If one description is lost, the correlation introduced in the known manner helps to estimate the lost coefficient from its counterpart in the pair. This requires knowledge, i.e. transmission, of the coefficient variances to the decoder. To ensure more accurate variance estimation, transform blocks can be separated into classes (e.g. predominantly horizontal details, predominantly vertical details, smooth, etc.) and variances are calculated separately within each class [156]. However, this increases the amount of side information.

The above-mentioned problems have been addressed in [29] in the light of whitening the transform coefficients prior to PCT. Whitened coefficients have close variances that can be considered as equal. Hence, there is no need to estimate and transmit those variances as they cancel in the estimator formula. The whitening transform is approximated by a subtraction of a downsampled and coarsely coded image from the original image [29]. Then, PCT as in [106] is applied to the coefficients of the residual (whitened) image. The resulting two descriptions of the residual image are combined with the duplicated version of

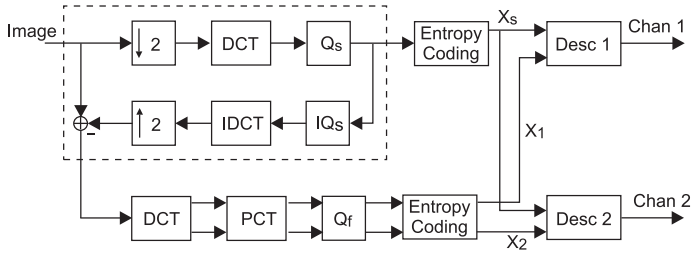


Figure 2.4: General scheme of method from [29].

the coarse image.

However, for the variables with equal variances, PCT does not introduce any redundancy. Thus, side reconstruction cannot have the distortion lower than $\sigma^2/2$, where σ^2 is the coefficient variance. We adopt the idea of 2-stage image coding. We refuse using PCT in the second coding stage as it does not introduce the redundancy in representation of the residual signal. We also suggest modifications in the coarse approximation stage and in the residual image stage aimed at improving the quality for a given bit budget by a better redundancy management. We present an algorithm, which optimizes the expected distortion based on channel conditions.

The general scheme of the method suggested in [29] is shown in Figure 2.4. The initial image is downsampled by two and then JPEG coded. Its decoded and interpolated version is subtracted from the initial image to approximate a whitening transform. DCT is applied to the residual image to get uncorrelated coefficients with the approximately equal variance. They undergo pairwise correlating transform (PCT) outputting two bitstreams. The JPEG coded coarse approximation is called *shaper* and is included into both descriptions. The redundancy in this method is mostly determined by duplicating the shaper as PCT introduces little redundancy when applied to the variables with similar variance. The authors [29] claim their method to produce better results than the method in [156].

We modify the above-mentioned method as shown in Figure 2.5. In the proposed scheme [101, 103], the shaper (blocks bordered by the dashed line) is generated by decimation with the arbitrary down-scaling factor of M followed by the JPEG coder (see Figure 2.5 (a)). We pay special attention to the way how the image is decimated and interpolated. We favor a B-spline-based least

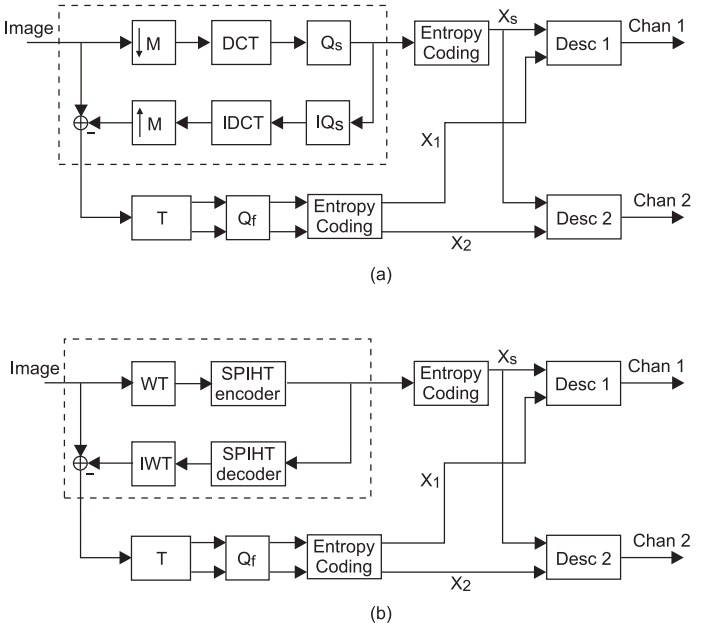


Figure 2.5: Varieties of proposed scheme: (a) shaper is obtained by spline resizing and JPEG coding; (b) shaper is obtained by SPIHT coding.

squares image resizing (biorthogonal projection) as it ensures the minimum loss of information [97]. Thus, the most important information is concentrated in the decimated image and included in both descriptions. For the decimated image, a DCT-based coder is a reasonable choice. Alternatively, the shaper can be generated by a wavelet-based coder, e.g. SPIHT (Figure 2.5 (b)). In this case, the biorthogonal projection is inherently included in the scheme.

The proposed scheme (Figure 2.5) is similar to one-level Laplacian pyramid [24]. The major difference is that in our coder the quantization of the base layer is performed before obtaining the residual image. We have found that the proposed scheme achieves better compression efficiency compared to Laplacian pyramid with the employed transform, quantization, and entropy coding. However, compression artifacts from the shaper are present in the residual image and affect its compression. Hence, the interpolation algorithm requires special attention.

In our modification, the residual image is coded by a JPEG-like coder using a block transform (denoted by T). It can be either DCT or lapped orthogonal transform (LOT). The transform coefficients are finely quantized

with the uniform quantization step (Q_r). Then, transform blocks are directly split into two parts in the *checkerboard* manner and entropy-coded. One part together with the shaper form *Description 1* while the second part combined again with the shaper form *Description 2*. Thus, each description consists of the coarse image approximation and *half* of the transform blocks of the residual image. Therefore, no extra redundancy is added in the residual image coding while generating two descriptions instead of one.

The proposed coder provides balanced descriptions both in terms of PSNR and bitrate. The amount of redundancy is also easily adjustable. The algorithm of optimal bit allocation subject to probability of the description loss is also provided. The following two sections explain in detail each stage of the coder. Also we give reasoning for using one or another method for each particular stage.

2.3 Coarse image approximation

The idea of this stage is to concentrate as much information as possible into the shaper within strict bitrate constraints. We would also like to reduce the artifacts which appear in the reconstructed coarse approximation and consequently in the residual image. To realize this idea we explore two alternatives: 1) Least squares image resizing prior to JPEG coding; 2) Wavelet-domain SPIHT coding.

2.3.1 Least squares spline-based resizing and JPEG coding

A JPEG coder with a limited bit budget would use coarse quantization applied to the coefficients of the original image thus causing unacceptable blocking artifacts. A better alternative, especially for low bitrate coding, is to decimate the image and to apply JPEG with moderate quantization factor. The original image resolution is reconstructed by interpolation as a post-processing step. It has been proven by an analytical model and numerical analysis that by this approach the bit budget is kept the same while the visual quality and PSNR are higher [22]. The method in [29] also makes use of this approach as follows. The decimation by the factor of two in each direction is achieved by averaging over two pixels along columns followed by the same operation along rows. The original resolution is reconstructed by the nearest neighbor interpolation. This

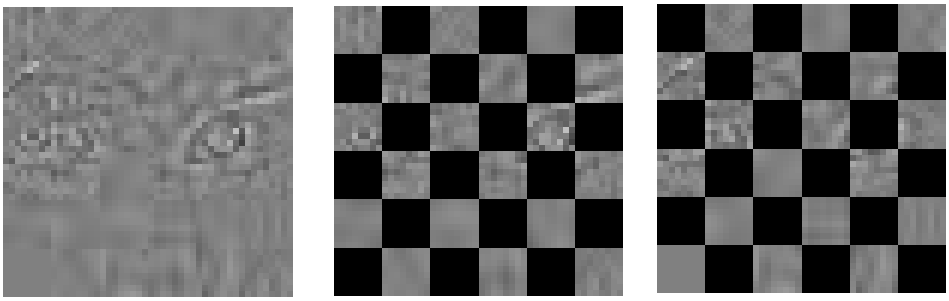
interpolation introduces blocking artifacts in the coarse approximation and as a result the residual image gets blocking artifacts as well.

In an attempt to concentrate more information in the coarse approximation and to make the residual signal closer to white noise, we identify the need of a better interpolation and decimation method. Spline-based interpolation methods have shown their superiority in terms of quality and computational complexity [52, 141]. In the spline formalism, a continuous image model is fit over the discrete pixels, involving B-spline or other optimized piecewise-polynomial basis functions. It allows resampling the initial image at any arbitrary finer grid. As far as the image decimation is concerned, it has to be performed using functions being biorthogonal to the chosen interpolation function. This is the biorthogonal projection or least squares paradigm, which ensures image decimation with a minimum loss of information [52, 97]. Our practical implementation uses a near least squares method for image decimation proved to be effective for a wide range of decimation ratios [53].

The redundancy in our coder is only determined by the size and quality of the shaper. Generally, there are two factors controlling the size of the shaper (and hence the redundancy). The first one is scaling (or interpolation) factor and the second one is the JPEG quantization factor. Using larger downsampling and quantization factors one can get lower level of redundancy, hence, lower quality of side reconstruction (reconstruction from only one description). Alternatively, using smaller downsampling and quantization factors, one can obtain higher quality side reconstruction. The quality of the two-channel reconstruction is determined mostly by quantization step used for quantization of LOT coefficients in the residual image.

2.3.2 Wavelet-based coding

An alternative to JPEG coding in obtaining good low bitrate image approximation is some wavelet-based coding scheme. In general, wavelets provide smooth reconstruction of compressed images even for low bitrates. As they are functions for multiresolution analysis, there is no need of a preliminary decimation step. In fact, the wavelet decomposition is precisely an orthogonal or biorthogonal projection into the space of synthesizing (reconstruction) wavelet functions. Moreover, the best wavelets for compression have been generated via splines, e.g. the famous 9/7 synthesis/analysis wavelet pair. In our



(a) Residual image (b) Description 1 (c) Description 2

Figure 2.6: Checkerboard splitting of residual image (in case of DCT).

scheme we have involved the SPIHT coding and quantization algorithm [123].

2.4 Residual image coding

Our approach relies on the quality versus bit budget compromise achieved at the coarse approximation stage. We speculate that our coarse approximation is as good as possible for the given bit budget. Thus, the residual signal is less informative, and there is no need to introduce redundancy to this signal. Respectively, the total redundancy is added by only duplicating the base layer (shaper). We essentially aim at avoiding redundancy in the residual image coding.

The residual image coding in our method is done by a block transform, e.g. blocks of 8×8 coefficients are considered. In coding the residual image, no redundancy is added provided that the proper block transform has been chosen. To generate MDC, the blocks are simply split into two descriptions in the *checkerboard* manner (see Figure 2.6).

We explore LOT and DCT as block transforms well suited for the residual image coding.

2.4.1 Coding the residual signal with block DCT

The residual image is transformed using 8×8 DCT. Then, all transformed blocks are finely quantized with a scalar quantizer using a constant quantization step Q_f. The transform blocks are split between two descriptions in the checkerboard manner and entropy coded separately.

2.4.2 Coding the residual signal with LOT

LOT is the alternative to DCT when the quality of the shaper is not good enough. In such cases some blocking artifacts can be encountered if the image reconstruction is based on only one description. LOT can efficiently smooth block borders due to the overlapping blocks it uses.

In LOT, each signal block of size N is mapped into a set of N basis functions, each of them is longer than N samples, i.e. overlapping over adjacent blocks [92]. For the 2D case the LOT's are implemented in a separable manner.

In our coder we use Malvar's LOT [92]. The overlapped blocks of the size 16×16 in a spatial domain correspond to 8×8 blocks in the transform domain. Next steps, i.e. quantization by a uniform quantization step Q_r and checkerboard-like splitting into two parts are essentially the same like in the case of DCT block coding.

2.4.3 Reconstruction when one description is lost

When the decoder receives both descriptions, the reconstruction is straightforward. In case of one-channel reconstruction, the lost coefficients are just filled with zeros. Then, the inverse quantization and inverse transform are applied. The shaper can be obtained from the received description and added to the reconstructed residual image.

It is obvious that this kind of reconstruction is appropriate when using DCT for coding of the residual image. It has been also found that it is the most appropriate way of reconstruction when using LOT for the residual image coding.

In [31, 66], it was shown that when reconstructing the original image from only one description, setting the lost coefficients equal to zero produces severe artifacts. Thus, [31, 66] present methods for estimation of the lost coefficients. In [66], the lost LOT coefficients are estimated as the mean of corresponding coefficients in the neighboring blocks. In [31], it was proposed to use an iterative procedure using maximally smooth recovery method. Moreover, a family of LOT transforms with advanced reconstruction capabilities was presented in [66]. However, it was found that for coding the residual zero-mean signal these methods work worse than just filling the lost coefficients with zeros. We suggest that this fact is connected with the high frequency nature of the

residual signal that does not allow estimation of the lost LOT block from the neighboring blocks.

2.4.4 Postprocessing for one-channel reconstruction

When DCT is used in the residual image coding, the image reconstructed from one description consists of the blocks, which have different quality. Thus, low bitrate of the shaper causes visible blocking artifacts. There are basically two types of blocking artifacts: artifacts in flat areas of the image and artifacts in the areas with high-frequency content. Artifacts in the flat areas of the image look similar to artifacts caused by coarse quantization. These artifacts are formed by the offset corresponding to the lost DC coefficient in the residual image and can be reduced by linear filtering across the block borders. In the areas with high-frequency content, high-quality blocks usually have texture while the blocks obtained solely from the shaper are flat. In these parts of the image, visual quality can be improved by smoothing the border between the flat and textured blocks.

Postprocessing in our coder is based on the deblocking filter of MPEG-4 [71, 87], which we modify according to our needs. The deblocking filter operates in two separate modes depending on the pixel behavior around the block boundary. In each mode, one-dimensional filtering operations are performed across the block boundaries along horizontal and vertical directions (see Figure 2.7). In Figure 2.7, we assume that filtering is performed in horizontal direction across the vertical block boundary. The block on the left is reconstructed with higher quality, and the block on the right is reconstructed from the shaper and has lower quality.

The following procedure is used to find a smooth region with blocking artifacts due to the small DC offset caused by the loss of the DC coefficient in the residual image. The flatness of the region is examined by the following measurement [87]:

$$F(\mathbf{v}) = \sum_{i=0}^{8} \phi(v_i - v_{i+1}) \tag{2.4}$$

where

$$\phi(\delta) = \begin{cases} 1, & |\delta| \leq T_1 \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

If $F(\mathbf{v}) \geq T_2$, the block is classified as smooth and *smooth region mode* is

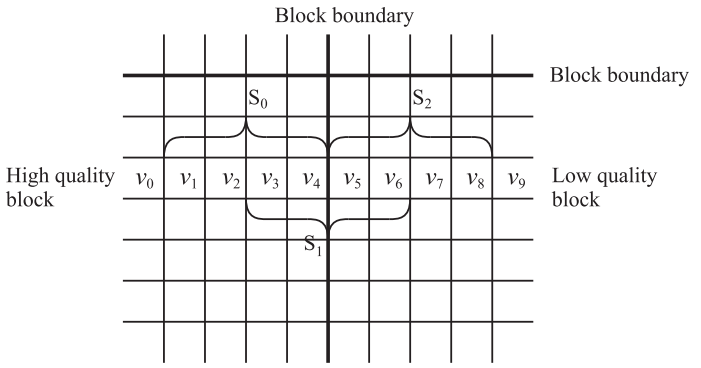


Figure 2.7: Block boundaries.

applied. Otherwise, the block is non-smooth and the *default mode* is applied. The thresholds are chosen as $T_1 = 2$ and $T_2 = 6$. In the smooth region mode, our filter is the same as the deblocking filter of MPEG-4. In this mode, 1-D nine-tap smoothing filter is applied across the block boundaries, as well as inside the block. Filtering is performed on eight pixels, four on each side of the boundary. The filter coefficients are $h(n) = \frac{1}{16}\{1, 1, 2, 2, 4, 2, 2, 1, 1\}$.

The default mode is modified compared to the MPEG-4 deblocking filter. In the default mode of MPEG-4 deblocking filter, two pixels are modified, each one from its side of the border. In our coder, the pixels from higher-quality block should not be modified. Instead, we modify only pixel v_5 from the lower-quality block. This is done in the following way. Let us define $a_{0,1}$, $a_{1,1}$, $a_{2,1}$, and $a_{3,1}$ as the four-point DCT coefficients of the pixel array S_1 (see Figure 2.7). The high-frequency coefficient $a_{3,1}$ is the major factor, affecting the blocking artifact. It is found [87] that the adjustment of this term helps to reduce the blocking artifact. In our coder, the high frequency component is modified by a factor between 0 and 1, resulting in

$$a'_{3,1} = a_{3,1} \frac{\min(|a_{3,0}|, |a_{3,1}|, \frac{|a_{3,0}|+|a_{3,2}|}{2})}{|a_{3,1}|} \tag{2.6}$$

where $a_{3,0}$ and $a_{3,2}$ are defined similarly to $a_{3,1}$ for the pixels arrays S_0 and S_2, respectively. In (2.6), the expression $\frac{|a_{3,0}|+|a_{3,2}|}{2}$ is used rather than $|a_{3,2}|$ as in [87] for the reason that $a_{3,2}$ corresponds to "over-smoothed" block of the image.

The coefficient $a_{3,1}$ can be found as

$$a_{3,1} = [k_3 \ -k_1 \ \ k_1 \ -k_3] \cdot [v_3 \ v_4 \ v_5 \ v_6]^t$$

where

$$k_1 = \frac{1}{\sqrt{2}} \cos \frac{\pi}{8},$$
$$k_3 = \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8}.$$

If the coefficient $a_{3,1}$ is modified to $a'_{3,1}$, pixel v_5 has to be modified as

$$v'_5 = v_5 + \frac{1}{k_1}(a'_{3,1} - a_{3,1}). \tag{2.7}$$

The clipping operation is applied to the value of v'_5 in order to keep it between 0 and 255. The filtering is performed separately in vertical and horizontal directions. The proposed filtering method improves both the subjective and objective quality of one-channel reconstruction of the image. The simulation results for the proposed postprocessing method can be found in Section 2.6.

2.5 Scheme analysis

2.5.1 Optimization

For the proposed scheme, we present an algorithm, which optimizes bit allocation subject to probability of the description loss under bitrate constraints. Our scheme exploits DCT transform and quantization. Thus, it is difficult to achieve exact bit allocation as in the case of progressive coders (e.g. SPIHT). However, approximate bit allocation can be performed.

Let p denote probability of the description loss and R a target bitrate. R_s is the bitrate of the shaper (coarse image approximation) and R_r is the bitrate of the residual image. The central distortion is D_0, and the side distortions are D_1 and D_2. As we consider balanced descriptions, $D_1 = D_2$. Then, our task is to minimize

$$2p(1-p)D_1 + (1-p)^2 D_0 \tag{2.8}$$

subject to

$$2R_s + R_r \leq R. \tag{2.9}$$

Consider checkerboard DCT-based coding of the residual image. The side distortion (D_1) is formed by the blocks, half of which are coded with the distortion D_0, and another half is coded with the distortion of the shaper D_s. Consequently, we can approximate D_1 as follows:

$$D_1 = \frac{1}{2}(D_s + D_0). \tag{2.10}$$

Expression (2.10) can also be used in the case of LOT coding of the residual. As LOT is by definition an orthogonal transform, the mean-squared error distortion in the spatial domain is equal to the distortion in the transform domain. However, side distortion in the transform domain is determined by the loss of half of transform coefficient blocks. Thus, the expression (2.10) is also valid in case of LOT.

Then, the constrained minimization (2.8) can be transformed to the unconstrained minimization with the Lagrangian

$$\begin{aligned} L(R_s, R_r, \lambda) = p(1-p)(D_s(R_s) + D_0(R_s, R_r)) \\ + (1-p)^2 D_0(R_s, R_r) + \lambda(2R_s + R_r - R). \end{aligned} \tag{2.11}$$

For higher bitrates, D_0 only weakly depends on R_s. It will be shown further in the simulation results. Hence, we can write $D_0(R_r)$. The minimization task is then simplified to

$$\begin{aligned} L(R_s, R_r, \lambda) = p(1-p)(D_s(R_s) + D_0(R_r)) \\ + (1-p)^2 D_0(R_r) + \lambda(2R_s + R_r - R). \end{aligned} \tag{2.12}$$

Generally, we are able to find experimentally the distortion-rate functions $D_0(R_r)$ and $D_s(R_s)$ to carry out the minimization task.

A closed-form solution can be found for i.i.d Gaussian random source. The distortion-rate function of the Gaussian source with variance σ^2 is

$$D(R) = \sigma^2 2^{-2R}. \tag{2.13}$$

Hence,

$$D_s(R_s) = \sigma^2 2^{-2R_s}. \tag{2.14}$$

It is shown [44] that Gaussian source is successively refinable with respect to the squared-error distortion measure. Hence, we can write

$$D_r(R_s, R_r) = \sigma^2 2^{-2(R_s+R_r)}. \tag{2.15}$$

Consequently, our task transforms to the unconstrained minimization of the function

$$\begin{aligned} L(R_s, R_r, \lambda) = \sigma^2 p(1-p)(2^{-2R_s} + 2^{-2(R_s+R_r)}) \\ + \sigma^2 (1-p)^2 2^{-2(R_s+R_r)} + \lambda(2R_s + R_r - R). \end{aligned} \tag{2.16}$$

We differentiate the Lagrangian with respect to R_s, R_r, and λ and solve the system of equations. The obtained optimal R_s and R_r are

$$\begin{aligned} R_s^* &= \frac{1}{2}R + \frac{1}{4}\log_2(p), \\ R_r^* &= -\frac{1}{2}\log_2(p). \end{aligned} \tag{2.17}$$

The rate allocation (2.17) is, however, sub-optimal because the results were obtained for the case of Gaussian random variable. The "true" optimal allocation must use rate-distortion characteristics of the particular image and solve the optimization task (2.12) or (2.11). Instead of using RD function of the Gaussian source (2.13), one can use another RD curve model of the form $D(R) = a2^{bR} + c$. In this case, Eq. (2.11) can also be solved analytically assuming that the source is successively refinable. The example of such solution is given in Sub-section 3.4.2 for our 2-stage video coder.

One can notice that the optimal bit allocation (2.17) has the same form as that for the copies of polyphase components in [122] if we substitute $R_0 = R_s + R_r$, and $R_1 = R_s$. This result is expected. In fact, our approach and the approach from [122] are based on the same principle. When all the descriptions are received, all pixels are reconstructed with fine quality. If some descriptions are lost, the pixels in corresponding spatial locations are reconstructed with lower quality. Thus, the results of the optimal bitrate allocation in terms of mean-squared error are expected to be the same.

2.5.2 Redundancy range

As it was mentioned earlier, the redundancy of the proposed method is $\rho = R_s$. Thus, optimal redundancy ρ^* is

$$\rho^* = R_s^* = \frac{1}{2}R + \frac{1}{4}\log_2(p). \tag{2.18}$$

We can see from (2.18) that optimal redundancy depends on the target bitrate R and the probability of description loss p. However, for $R \leq -\frac{1}{2}\log_2(p)$, optimal redundancy ρ^* is zero or negative. We interpret this result in the following way. For these values of R and p, one should not use multiple description coding. Instead, single description coding has to be used. In other words, when $p \leq 2^{2R}$, redundancy is set to zero, and single description coding is used. However, these results can differ from the actual values obtained by optimization using "true" RD characteristics of particular source. Thus, in practical implementation of the proposed bit allocation algorithm we use the following scheme. If the obtained value of R_s is less than 0.05, we fix $R_s = 0.05$ and $R_r = R - 2R_s$.

One can see from (2.18) that the upper limit for redundancy is obviously $R/2$. It corresponds to $R_r = 0$. In this case, all the bit budget is allocated to shaper, which is duplicated in both descriptions. This is achieved when $p = 1$, the situation when the channel is not functioning. Thus, when probability of the description loss is approaching unity, bit allocation algorithm simply duplicates the data in both descriptions.

2.5.3 Practical bit allocation

In the previous Sub-section we have solved the problem of optimal bit allocation. It is easy to code the shaper to the given rate R_s with SPIHT or another embedded coder. However, it is difficult to achieve the exact target bitrate when coding the shaper with JPEG coding. Fortunately, the approximate bitrate can be estimated.

Coding the shaper

Down-scaling for better transform compression at low bitrates was studied in [22] by Bruckstein et al. The authors presented an algorithm which finds

the optimal down-scaling factor for the given bitrate. The optimal down-scaling factor is found based on the target bitrate and the second-order image statistics. Thus, using the algorithm proposed in [22] one can find the appropriate down-scaling factor to code the given image with bitrate R_s^*.

Coding the residual signal

It was mentioned earlier that subtracting the coarse image approximation has the effect of applying whitening transform to the original image. As the residual signal resembles white noise, the DCT coefficients of the residual image have approximately the same variance. Thus, the same quantization step is used for all DCT coefficients. The compression ratio and the bitrate of the residual signal are determined by the variance of the residual image σ_r^2 and the quantization step Q_r.

Let us model the DCT coefficients of the residual image as i.i.d. Gaussian random variables with variance σ_r^2. DCT is an orthogonal transform. Thus, variance of DCT coefficients σ_r^2 can be estimated as the variance of pixel values in the residual image.

The rate-distortion function of the Gaussian source is

$$R = \frac{1}{2}\log_2 \frac{\sigma^2}{D}. \tag{2.19}$$

Under the assumption of fine quantization (which is valid for the residual coding stage) the squared-error distortion of the uniform quantization is [62]

$$D \cong \frac{1}{12}Q^2 \tag{2.20}$$

where Q is the quantization step. Hence,

$$R_r \cong \frac{1}{2}\log_2 \frac{12\sigma_r^2}{Q_r} \tag{2.21}$$

and the quantization step for the residual image is chosen as

$$Q_r = \lceil 2\sqrt{3}\sigma_r 2^{-R_r} \rceil \tag{2.22}$$

where $\lceil \cdot \rceil$ denotes rounding to the nearest integer towards positive infinity.

Basic coding algorithm

Briefly, the encoding is done as in the following. Having the probability of the description loss, we find the optimal rates R_s^* and R_r^* according to (2.17). Then, we estimate the optimal down-scaling factor using the procedure from [22]. Another alternative is to use SPIHT coder for bitrate R_s^*. Then, we decimate the original image with the estimated down-scaling factor and code the decimated image with JPEG to fit the shaper bitrate (R_s^*). Then, we obtain the residual image as the difference between the original image and the shaper. From the residual image, we estimate the variance of its pixels σ_r^2. We apply (2.22) to find the quantization step Q_r, and code the residual image.

2.6 Simulation results

In this section, varieties of our method are explored and compared between themselves and with other MDC algorithms. For the evaluation, first, we applied our method to the test images *Lena* (512×512, 8 bpp), *Stream and Bridge* (512×512, 8 bpp), and *Boat* (256×256, 8 bpp). For most experiments, we generated two rate-distortion curves. The first curve shows the reconstruction PSNR versus bitrate under the assumption that both descriptions are received. The second curve illustrates the case, when one description is lost. It is obtained by taking the mean result of two descriptions used separately to reconstruct the image.

The proposed coder has been tested for the gray-scale images. However, it can be easily extended to coding the color images in a way similar to JPEG coding. In this case, the color representation is converted from RGB to YCbCr followed by down-scaling the chrominance components Cb and Cr by factor 2 in both vertical and horizontal directions. Then, the luminance component Y and the chrominance components Cb and Cr are compressed independently with the proposed coder. The Huffman tables should be adjusted for each color component.

2.6.1 Proper decimation and interpolation

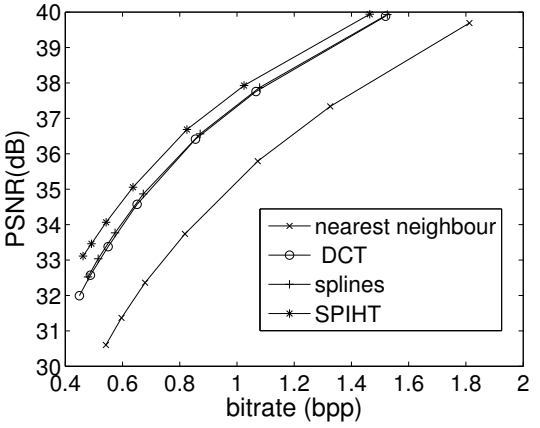
In the first experiment, we compare different decimation and interpolation methods to produce the shaper. As for the residual image coding, we fix it to perform block transform coding, involving Malvar's LOT [92]. We apply three

decimation/interpolation methods. The first is based on decimation by 2 by averaging over four nearest points and nearest neighbor interpolation, similarly to [29]. Second is DCT-based decimation and interpolation [163], and the third is a near least squares B-spline-based decimation and interpolation [53]. Those three approaches have been combined with the JPEG coder to obtain the shaper. Additionally, the shaper is obtained by a wavelet-domain SPIHT coding. Figure 2.8 shows the reconstruction results when both descriptions have been received (central PSNR) and averaged PSNR for reconstruction from one description (mean side PSNR). As can be seen from Figure 2.8, among JPEG methods proper anti-aliasing decimation and interpolation give substantial improvement. There, splines and global DCT are quite competitive as pre- and post-processing functions. However, the spline-based method is computationally less costly. Among all methods, wavelet-based SPIHT gives superior results.

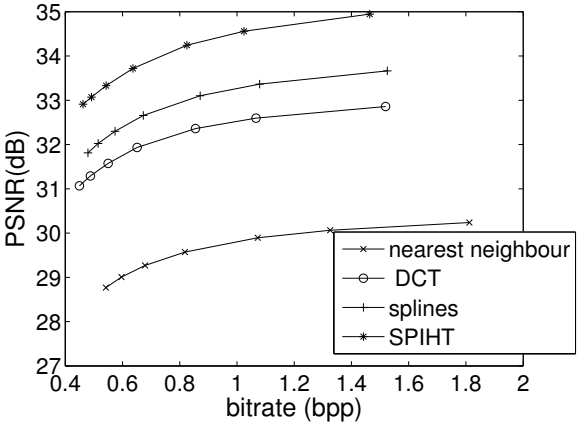
In our experiments we have used linear splines for interpolation and their biorthogonal counterparts for decimation. Higher order splines would give better results in a pure decimation/interpolation setting. However, the JPEG quantization generates artifacts and the subsequent higher-order interpolation makes them better visible. Linear interpolation plays an additional smoothing effect to these artifacts. What is more important is the least squares setting where the image is properly decimated subject to the chosen interpolation method.

Figure 2.9 shows interpolation results for *Lena* image. The original 512×512 *Lena* image is downsampled with biorthogonal splines to the resolution 128×128. The downsampled image is then quantized and coded with the JPEG algorithm. Then, quantized downsampled image is interpolated to the original resolution with linear splines. One can see that $R = 0.085$ allows to reconstruct this image with $D_s = 26.73$ dB.

Figure 2.10 compares the performance of the Spline-LOT method with that of the method introduced in [29] (denoted by WCT) for the same shaper quantization factor. Spline-LOT coder clearly outperforms WCT coder both for central and side reconstruction mainly due to the adequate decimation and interpolation.



(a) Central PSNR



(b) Mean side PSNR

Figure 2.8: Image *Lena*. Comparison of different interpolation methods.

(a) (b)

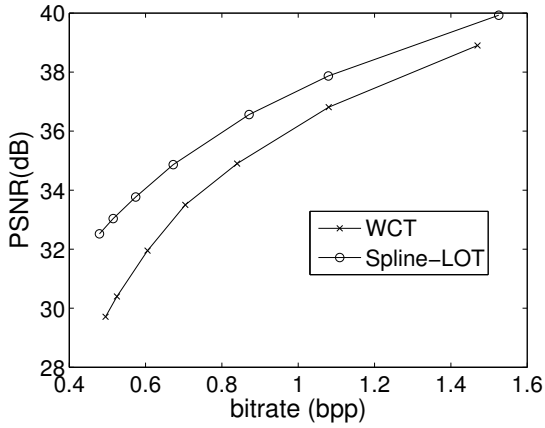
Figure 2.9: Image interpolation results. (a) Original image 512×512 pixels; (b) Image downscaled to 128×128, JPEG-coded and interpolated to original resolution, $D = 26.73$ dB, $R = 0.085$ bpp.

2.6.2 Shaper scaling and quantization

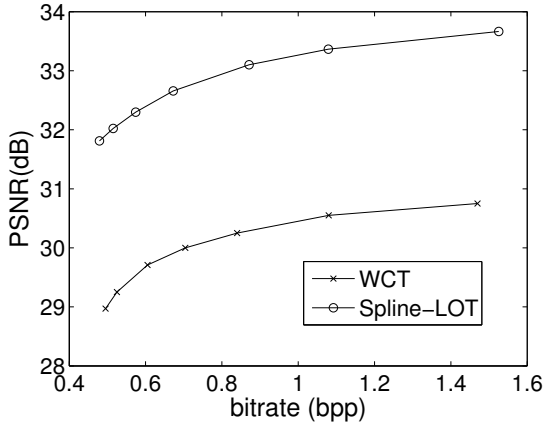
Next, we explore how the shaper quality influences the total reconstruction quality. Again, our residual image coder is the LOT-based one while the shaper coder is based on least squares spline decimation/interpolation and JPEG with different quantization factor (this scheme is denoted as Spline-LOT). The shaper quantization factor Q_s is determined as a multiplication factor applied to DCT coefficients before their quantization. The results are shown in Figure 2.11.

One can see a higher shaper quantization factor slightly reduces the PSNR for central reconstruction but at the same time increases the PSNR when one description is lost. Using the finer quantization we thereby provide more bitrate to the shaper. Thus, we introduce more redundancy that improves the side reconstruction.

In addition, the rate-distortion curves for central reconstruction have much steeper slope than the rate-distortion curves for the side reconstruction. It evidences that finer quantization of the residual image results in better central reconstruction but has little influence on the single description reconstruction. This effect is cased by the form of the side distortion $D_1 = (D_0 + D_s)/2$. As $D_s \gg D_0$, D_s contributes more to D_1. Thus, finer quantization of the residual image will decrease D_0 but will not significantly decrease D_1. In order to decrease D_1, one has to allocate more bits to the shaper. This will decrease $D_s/2$ component in D_1.

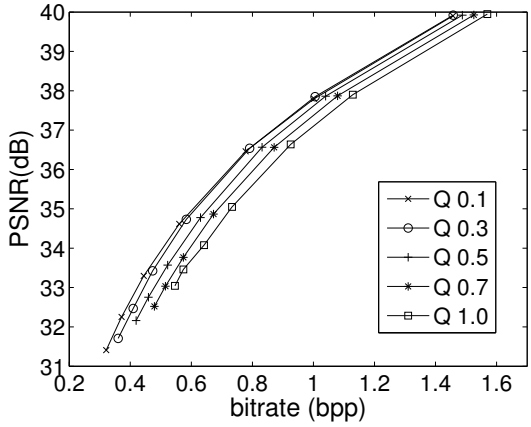


(a) Central PSNR

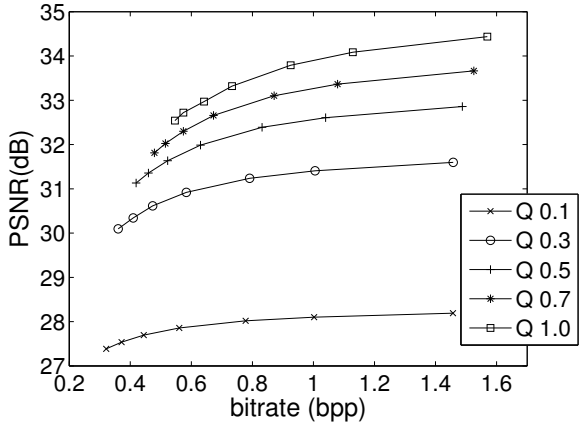


(b) Mean side PSNR

Figure 2.10: Image *Lena*. Comparison of Spline-LOT and WCT coder. $Q_s = 0.7$.



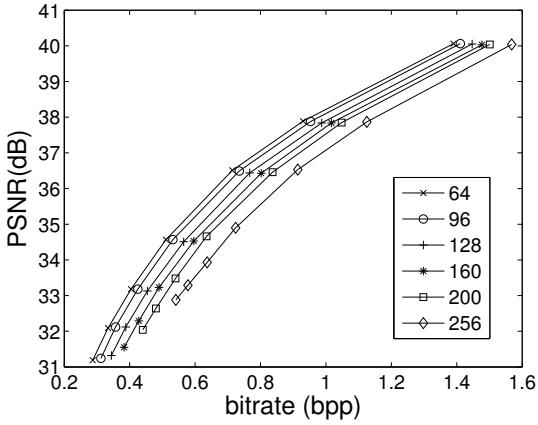
(a) Central PSNR



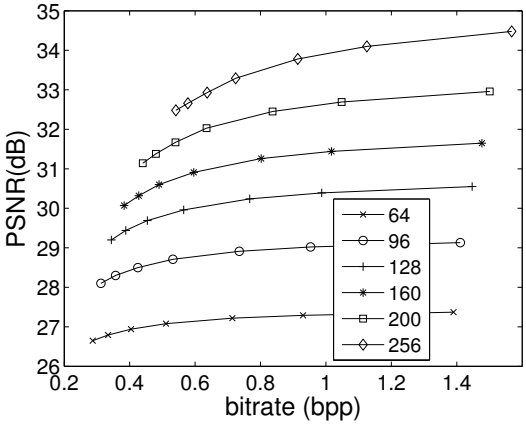
(b) Mean side PSNR

Figure 2.11: Image *Lena*. Rate-distortion performance of Spline-LOT coder for different values of Q_s.

The next algorithm uses variable down-scaling factor for coding the shaper. The results are parameterized by the shaper resolution and are shown in Figure 2.12. Decreasing the shaper resolution makes it possible to achieve low bitrates.



(a) Central PSNR



(b) Mean side PSNR

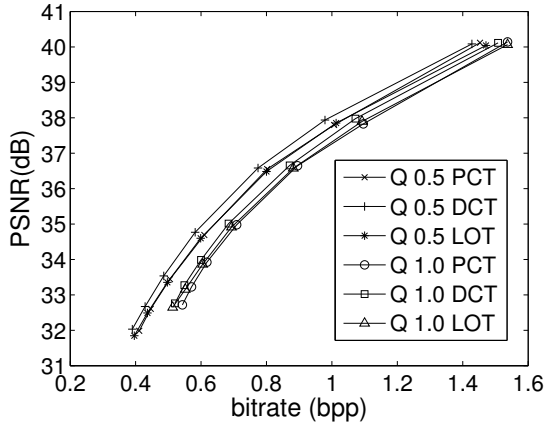
Figure 2.12: Image *Lena*. Rate-distortion performance of Spline-LOT coder for different shaper resolutions.

2.6.3 Residual signal coding

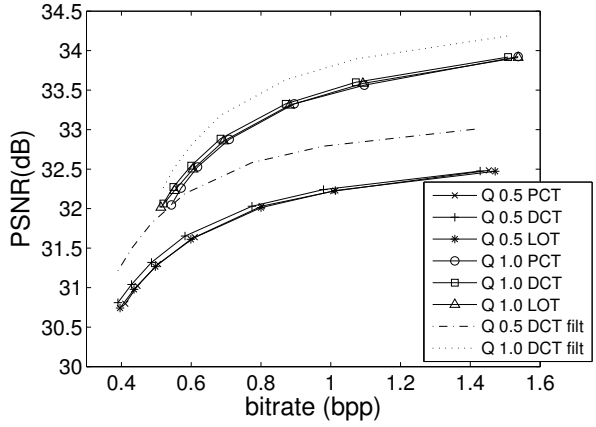
In the next set of experiments, the shaper is obtained by linear B-spline downsampling followed by JPEG-coding. The residual signal is coded as in the following. In the first coder, the residual image is coded by the method from [29]

involving DCT followed by PCT (this method is denoted Spline-PCT). The second coder exploits block-wise DCT followed by splitting the blocks between two descriptions (this method is denoted Spline-DCT). The third coder (Spline-LOT) uses LOT for coding the residual. For the Spline-DCT coder, mean side reconstruction results are obtained with and without postprocessing. Postprocessing is based on the deblocking filter described in Section 2.4.4.

Figure 2.13 compares the rate-distortion performance of the mentioned coders for central and mean side reconstruction for test image *Lena*. Figure 2.14 shows the simulation results for image *Stream and bridge*.

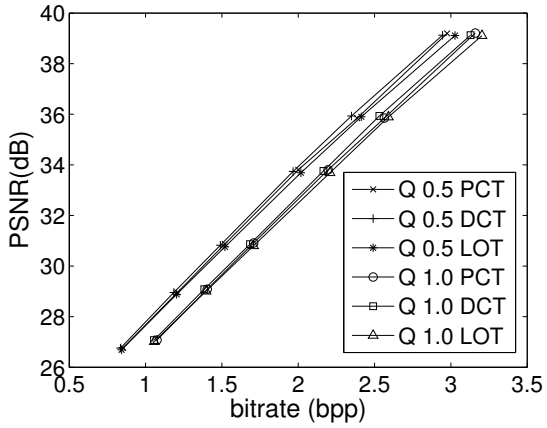


(a) Central PSNR

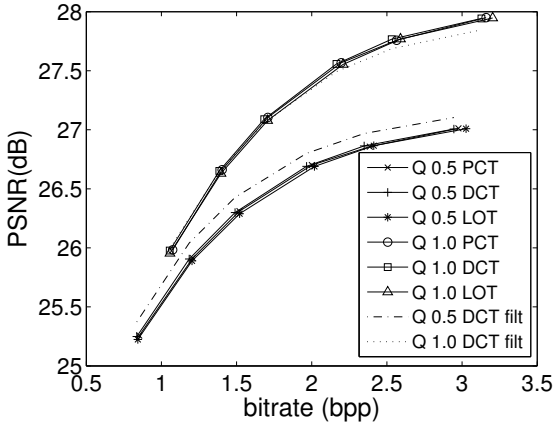


(b) Mean side PSNR

Figure 2.13: Image *Lena*. Comparison of Spline-DCT, Spline-LOT, and Spline-PCT coders for different values of Q_s.



(a) Central PSNR



(b) Mean side PSNR

Figure 2.14: Image *Stream and bridge*. Comparison of Spline-DCT, Spline-LOT, and Spline-PCT coders for different values of Q_s.

One can see from the figures that for both low-frequency (*Lena*) and high-frequency (*Stream and bridge*) images, DCT shows slightly better performance than LOT and PCT. The results are similar for both low-frequency and high-frequency images. Surprisingly, in terms of PSNR, Spline-DCT coder is competitive and even better than expected to be superior, Spline-LOT coder. While the latter is showing less blocking artifacts, it is not as efficient as DCT in compressing the high-frequency residual image. We suppose that this result is connected with the fact that originally, LOTs have been optimized to compress low-frequency signals [92]. However, the residual image mostly contains high frequencies. One can speculate that using transforms which are optimized for higher frequency content images could give certain improvement in the presented scheme.

The experiments with DCT in the residual image coding emphasize once again the importance of good shaper coding. If we keep the quality of the shaper low to achieve smaller redundancy, the blocking (checkerboard-like) artifacts are more visible. This is caused by reconstruction of neighboring blocks with different quality. However, if the shaper quality is high then, for most of the images, those kinds of artifacts are not visible. At least, they do not look visually more annoying than the artifacts caused by coding the residual image by LOT or PCT. Blocking artifacts due to DCT-based coding can be further reduced by postprocessing.

Figure 2.13 shows that postprocessing is beneficial for wide range of bitrates and different redundancies, especially for low-frequency images like *Lena*. For high-frequency images, postprocessing gives some advantage when the shaper quality is low. However, postprocessing does not improve the side reconstruction of the high-frequency image when the shaper has higher quality. This result is expected, as it is difficult to reconstruct high-frequency content of the block from its neighborhood.

Figures 2.15 and 2.17 show the images reconstructed from one and two descriptions by the Spline-DCT coder together with the images reconstructed from one and two descriptions by the Spline-LOT coder. Figures 2.16 and 2.18 are show the enlarged sub-image areas from Figures 2.15 and 2.17, respectively. One can notice that DCT and LOT produce different visual artifacts when the image is reconstructed from a single description. In particular, DCT causes blocking artifacts caused by different reconstruction quality of neighboring blocks. One can see this checkerboard-like pattern in Figures 2.16(b), 2.18(b),

when the blocks that have details (texture) are neighboring with the low-pass filtered blocks that belong to shaper. In case of LOT, these patterns are smoothed that can be observed from Figures 2.16(d) and 2.18(d). In case of DCT, the edges may be broken (like tree branches in Figure 2.18(b)). In case of LOT, the checkerboard pattern is smoothed or disappeared (Figure 2.18(d)).

Figure 2.15: Reconstructed image *Lena*, spline interpolation and DCT coding of shaper. DCT coding of the residual ($R = 0.623$ bpp, $\rho = 28\%$): (a) reconstruction from two descriptions, $D_0 = 35.24$ dB; (b) reconstruction from *Description 1*, $D_1 = 31.84$ dB. LOT coding of the residual ($R = 0.632$ bpp, $\rho = 27.5\%$): (c) reconstruction from two descriptions, $D_0 = 34.80$ dB; (d) reconstruction from *Description 1*, $D_1 = 30.78$ dB.

Figure 2.19 shows the effect of postprocessing for one-channel reconstruction of the Spline-DCT coder. Although postprocessing does not reconstruct

(a) DCT. Both descriptions

(b) DCT. *Description 1*

(c) LOT. Both descriptions

(d) LOT. *Description 1*

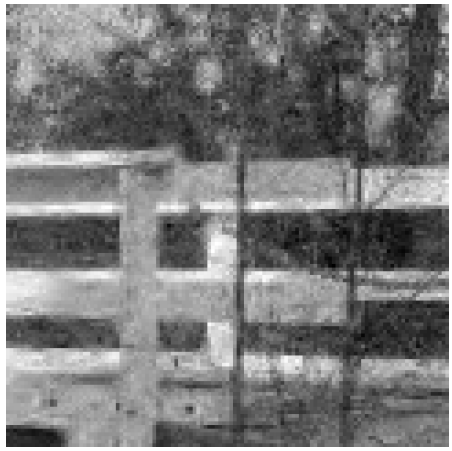
Figure 2.16: Reconstructed image *Lena*. Magnified area of image in Figure 2.15.

Figure 2.17: Reconstructed image *Stream and bridge*, spline interpolation and DCT coding of shaper. DCT coding of the residual ($R = 1.700$ bpp, $\rho = 14.7\%$): (a) reconstruction from two descriptions, $D_0 = 32.10$ dB; (b) reconstruction from *Description 1*, $D_1 = 26.52$ dB. LOT coding of the residual ($R = 1.705$ bpp, $\rho = 14.7\%$): (c) reconstruction from two descriptions, $D_0 = 31.58$ dB; (d) reconstruction from *Description 1*, $D_1 = 25.63$ dB.

(a) DCT. Both descriptions

(b) DCT. *Description 1*

(c) LOT. Both descriptions



(d) LOT. *Description 1*

Figure 2.18: Reconstructed image *Stream and bridge*. Magnified area of image in Figure 2.17.

the lost high-frequency content of the lower-quality blocks, it increases both the subjective and objective quality of the image reconstructed from one description. The results of postprocessing for one-channel reconstruction for wider range of redundancies can be found in Figure 2.21 and Tables 2.2 and 2.3. The results show that postprocessing is beneficial for wide range of redundancies.

(a) Before postprocessing (b) After postprocessing

(c) Before postprocessing (d) After postprocessing

Figure 2.19: Effect of postprocessing on one-description reconstruction; $D_0 =$ 35.81 dB, 0.636 bpp: (a) Not filtered, $D_1 = 28.78$ dB; (b) Filtered, $D_1 =$ 29.88 dB; (c) Magnified area of image in (a); (d) Magnified area of image in (b).

2.6.4 Bit allocation

This section shows the results of bit allocation algorithm proposed in Subsection 2.5.3. Test image is *Boat* (256×256, 8 bpp). The bit allocation algorithm is given the desired bitrate and the probability of the description loss. Bit allocation procedure exploits the closed-form solution (2.17). One has to remember that this bit allocation is suboptimal as it does not use the "true" rate-distortion characteristics of the source.

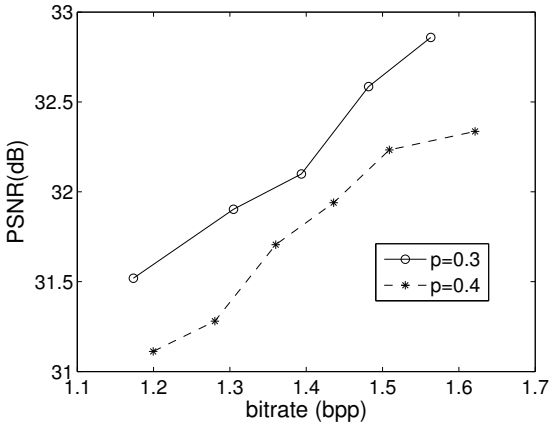
Figure 2.20 shows the RD curves obtained with the bit allocation algorithm for different probabilities of description loss. The coder in these experiments exploits linear splines for decimation and interpolation, and DCT for residual signal coding. This figure shows the desired behavior of the rate allocation algorithm. In particular, higher probability of description loss results in bit allocation, which produces higher side PSNR and lower central PSNR.

2.6.5 Comparison with other MD image coders

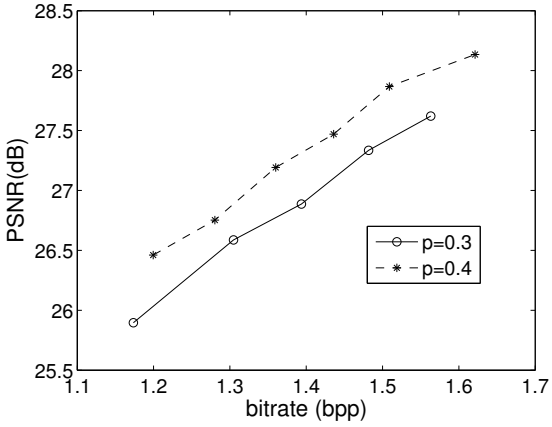
In the above, we compared our coder with the WCT coder of [29]. Figures 2.10(a) and 2.10(b) prove that our coder clearly outperforms the WCT coder. We suggest that this is due to proper decimation/interpolation.

Here we compare our 2-stage coder with other MD image coders based on JPEG. The 2-stage coder exploits B-splines and JPEG for coding the shaper. The residual image is coded with DCT. Two modifications of this coder are used for comparison: with and without postprocessing described in Section 2.4.4. These coders are called 2-stage+post-filt and 2-stage, respectively. We compare our 2-stage coder with two MD coders presented in [156]. One of those coders is a JPEG-based MDTC image coder. Correlating transform is applied to the pairs of DCT coefficients. The correlation added by this transform helps to estimate the value of the lost coefficient from the received coefficient. Another coder (MDSQ) is based on applying multiple description scalar quantization [145] to DCT coefficients of the JPEG coder. Test image *Lena* (512×512, 8 bpp) is used for comparison.

The central distortion for MDTC and MDSQ coders is $D_0 = 35.78$ dB. The central distortion for our 2-stage coder ranges from 35.80 dB to 36.00 dB. Figure 2.21 represents the RD curves for one-channel reconstruction for the 2-stage, 2-stage+post-filt, MDTC, and MDSQ coders. Different operating points for the 2-stage coder are obtained by varying the down-sampling factor for the shaper.



(a) Central PSNR



(b) Mean side PSNR

Figure 2.20: Image *Boat*. Bit allocation for different values of p.

Figure 2.21 demonstrates that even without postprocessing, 2-stage coder substantially outperforms MDTC coder for the whole range of redundancies. The difference is even larger for low redundancies. We suggest that the superior performance of the 2-stage coder in the low redundancy region is due to down-sampling before JPEG coding of shaper. This down-sampling before compression helps to obtain higher PSNR for low bitrates compared to the conventional JPEG compression. The 2-stage coder also performs better than MDSQ coder or comparable with it for higher redundancies. For moderate redundancies, our coder works better than MDSQ. Moreover, our coder is able to achieve smaller redundancies than the coders based on MDSQ and MDCT. One can notice that the 2-stage coder is able to produce meaningful side reconstruction even with the redundancy less than 5%. Figure 2.21 shows that postprocessing can improve both the subjective and objective quality, especially for low bitrates. The increase in PSNR due to the postprocessing is up to 1 dB.

The simulation results for the 2-stage coder with and without postprocessing for image *Lena* are given in Table 2.2. The simulation results for image *Stream and bridge* can be found in Table 2.3.

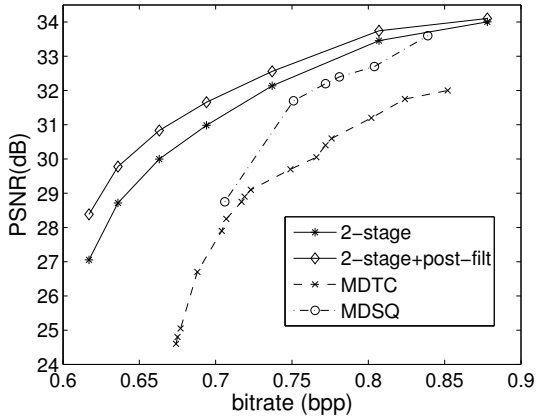


Figure 2.21: RD performance of different coders; image *Lena* (512×512). Reconstruction from a single description. For MDTC and MDSQ, $D_0 = 35.78$ dB. For 2-stage and 2-stage with postprocessing, $D_0 \approx 35.80$ – 36.00

Bitrate (bpp)	Mean side PSNR (dB)	Mean side PSNR (post-filtering) (dB)	Central PSNR (dB)	Shaper PSNR (dB)	Redundancy (%)
0.617	27.053	28.383	35.834	24.340	4.6
0.636	28.714	29.776	35.813	26.149	8.8
0.663	29.998	30.832	35.792	27.600	13.7
0.694	30.984	31.658	35.828	28.751	19.1
0.737	32.138	32.559	35.839	30.169	26.4
0.807	33.458	33.743	35.983	31.871	38.6
0.878	34.006	34.104	35.953	32.667	45.3

Table 2.2: Performance of 2-stage coder. Image *Lena* (512×512, 8 bpp).

Bitrate (bpp)	Mean side PSNR (dB)	Mean side PSNR (post-filtering) (dB)	Central PSNR (dB)	Shaper PSNR (dB)	Redundancy (%)
1.577	24.026	24.630	32.383	21.345	3.1
1.612	25.271	25.661	32.391	22.703	6.5
1.659	25.895	26.152	32.396	23.400	9.6
1.748	26.538	26.656	32.396	24.131	14.3
1.833	26.954	26.995	32.403	24.612	18.1
1.911	27.25	27.228	32.415	24.957	21.3
1.950	27.373	27.323	32.419	25.101	22.7

Table 2.3: Performance of 2-stage coder. Image *Stream and bridge* (512×512, 8 bpp).

2.7 Chapter conclusions

This chapter has surveyed MDC approaches to coding random variables and images. Image coders consist of several typical blocks, e.g. transform, quantization, coefficient scanning and entropy coding. Multiple descriptions can be generated at one of these blocks, introducing a controlled amount of redundancy which allows reconstruction from any single received description. For example, redundancy can be added at the transform (MDTC, quantized frames), quantization (MDSQ, MDLVQ), scanning process, or even at the preprocessing stage (polyphase transforms) and the channel coding stage (MD-FEC).

We have developed a practical MDC method that improves the two-stage scheme proposed previously in [29]. The first stage of our coder employs

spline interpolation in order to obtain the image with lower resolution, which is then coded and sent to both channels. This coarse image is coded this way in order to have a lower bitrate, yet providing smooth reconstruction. Then, properly interpolated, this image is subtracted from the original one, yielding the residual image. We spend no redundancy in dividing it into two descriptions. To achieve this, a checkerboard splitting of block transform coefficients is applied.

Two block-transform coders have been compared for coding the residual image. The simpler DCT-based coder has shown the results competitive to the LOT-based coder. While the latter was expected to yield reconstructed images with less blocking artifacts, the good results for the former prove that we have succeeded to make the residual image as high-frequency (noisy-like) as possible and correspondingly better compressible by DCT. The improved performance is due to the adequate decimation/interpolation scheme we have applied, which is based on biorthogonal projection (either spline or wavelet). The postprocessing in case of the DCT-based coder reduces blocking artifacts and increases the side reconstruction quality.

Our MDC method shows better performance comparing to the method in [29] both for reconstruction from one and two descriptions. It also outperforms the MD coders from [156].

Chapter 3

Multiple Description Coding of Video

Recent video coding standards employ motion-compensated prediction to efficiently remove the temporal correlation in video sequences and to split the video information into the form of prediction error and motion vectors. Consequently, typical MDC for video has to address coding of both prediction errors and motion vectors. In addition, special attention has to be given to the synchronization between the encoder and decoder. Encoding the prediction error into multiple descriptions can be handled exactly as in the case of images; that is, by subsampling in spatial or transform domain, and/or by proper scanning and quantization of transform coefficients. The correlation in the temporal domain can be utilized by suitable temporal domain subsampling.

This chapter is organized as follows. Section 3.1 addresses the most important problems of MDC of video by surveying the relevant papers. For more extensive treatment of the topic, we refer to the excellent overview paper by Wang et al [157]. Sections 3.2-3.7 describe the proposed MD video coding scheme. Section 3.2 gives an overview of the encoding and decoding process in general while Section 3.3 describes each block of the proposed scheme in detail. Section 3.4 presents the analysis of the proposed scheme and Section 3.5 discusses its computational complexity. Section 3.6 offers a packetization strategy. Section 3.7 presents the simulation results. Finally, Section 3.8 concludes the chapter.

3.1 Survey of MD video coding methods

3.1.1 Prediction loops mismatch control

One general problem connected with multiple description video coders based on motion-compensated prediction is the mismatch between the states of the encoder and decoder when the description is lost. This problem can be solved either by generating extra prediction loops for any potential loss of description, or by embedding the prediction in each description separately.

An MDC approach using three prediction loops in the encoder to mimic different description loss scenarios was proposed by Reibman et al. [115, 116]. In this approach, motion vectors and headers are simply duplicated in both descriptions. I-frames and prediction errors in P-frames are coded using one of the previously developed MD image coding approaches, e.g. pairwise correlating transform [106, 155]. The coder uses the central (two-channel-based) reconstruction as a reference picture in motion-compensated prediction. To deal with the prediction loop mismatch, the encoder has three different prediction loops, emulating three possible decoding scenarios: both descriptions received or either of the descriptions lost [115, 116].

A general diagram of the MD video coder based on three prediction loops is shown in Figure 3.1. The encoder stores three previous frames P_0, P_1, and P_2 reconstructed from both descriptions, *Description 1* and *Description 2*, respectively. For each block X, the decoder forms the corresponding block from the previously reconstructed frames through motion-compensated prediction.

The residual R (the motion-compensated difference) in the central prediction loop (assuming both descriptions are available) is coded using a general MD image coder (labelled in the scheme as MDC encoder). The MDC encoder produces two descriptions $\widetilde{R}_1$ and $\widetilde{R}_2$ which are sent to *Channel 1* and *Channel 2*. The encoder also has the MDC decoder block, which receives $\widetilde{R}_1$ and $\widetilde{R}_2$ and produces $\widehat{R}_1$ and $\widehat{R}_2$: two estimates of R from *Description 1* and *Description 2*, respectively. If the decoder gets only one of $\widehat{R}_1$ and $\widehat{R}_2$, the reconstruction in the absence of any additional information is $P_i + \widehat{R}_i$. The side loop prediction errors $M_1 = X - P_1 - \widehat{R}_1$ and $M_2 = X - P_2 - \widehat{R}_2$ are also coded by the single description encoders and included in the corresponding descriptions. Coding the M_1 and M_2 controls the mismatch between the prediction loops in the encoder and decoder in the case of description loss. Three

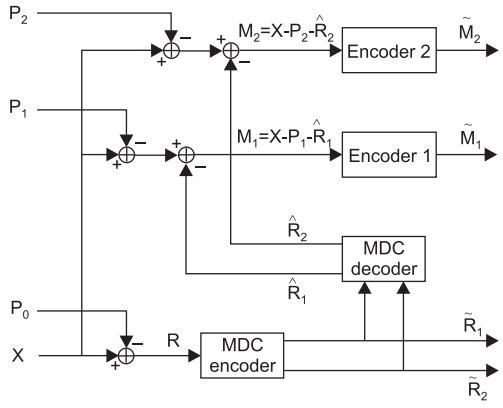


Figure 3.1: Framework for 3-prediction-loop MDC in P mode [116].

different algorithms for mismatch signal coding were proposed [115]. Algorithm 1 completely codes the mismatch error to the precision of quantization error. Algorithm 2 uses partial control of the mismatch error. Algorithm 3 does not use side prediction loops at all. Instead, it allocates more redundancy to the single-channel reconstruction of signal R. The decoder never uses the mismatch error signal when both descriptions are received [115].

The *matching-pursuits multiple description coding* (MP-MDC) of Tang et al. [138] is also based on the 3-loop scheme described earlier. This approach modifies DCT structure to the matching pursuits framework, where the redundancy in multiple descriptions is controlled by the number of shared basis functions. Unlike the approach of Reibman, where the side loops are considered as pure redundancy, the approach of Tang uses maximum likelihood estimation in order to improve the reconstruction quality of the central decoder by using the side loop mismatch error signal.

The three-prediction-loop architecture, as illustrated by Figure 3.1 [115], is constrained to have only two descriptions for the reason that the number of drift-compensating contributions grows exponentially with the number of descriptions. The *independent flow MD video coder* (IF-MDVC) introduced in [48] exploits polyphase transform to create two or more descriptions. The input video undergoes a polyphase transform, providing N independent flows. Motion-compensated prediction is performed in each flow separately and is followed by the entropy coding. When one description is lost, the lost samples are interpolated from the received samples [48]. This coder does not require

any drift compensation as the prediction is made in each description separately. The redundancy in this coder is due to the reduced coded efficiency caused by breaking the cross-dependencies in the original video sequence.

An approach called *mutually refining DPCM* (MR-DPCM) solves the problem of prediction loop mismatch by using two independent prediction loops in the encoder [145]. Each description uses its own prediction loop, where the reference frame is the single description reconstruction. The DCT coefficients of motion-compensated differences are quantized with shifted quantizers (MDSQ) described in Sub-section 2.1.1. When the decoder receives both descriptions, it combines them to produce a higher quality reconstruction. Regunathan and Rose have enhanced the MR-DPCM scheme by using an estimation-theoretic approach [113].

A drift-free multiple description coder has been introduced by Boulgouris et al. [18]. This coder has one prediction loop. Only the information common for both descriptions is used for motion-compensated prediction. This is achieved by coding the prediction error with a wavelet coder. Wavelet coefficients from the lower subbands are duplicated in both descriptions and used in a reference frame. The coefficients from higher subbands are split between two descriptions and are not used in motion-compensated prediction.

A simple MD video coder has been introduced by Reibman et al. [114]. Motion vectors and header information are duplicated in both descriptions. DCT coefficients higher than the optimal threshold for this block are duplicated in both descriptions, while lower coefficients are alternated between two descriptions. Thus, the decoder can decode a single description with an H.263 compliant decoder. When two descriptions are received, the decoder simply merges them into one H.263 stream. The drift in the case of description loss is corrected when receiving the I-frame.

3.1.2 Coding motion vectors

Motion vectors are essential for video sequence reconstruction. Thus, the full MDC system has to take into account coding of the motion vectors. A straightforward approach in treating the motion information is to duplicate the motion vectors in all the descriptions in order to protect the motion information in the case of channel losses. However, motion vectors can take a substantial part of the target bitrate, especially in low-bitrate budgets. This justifies the

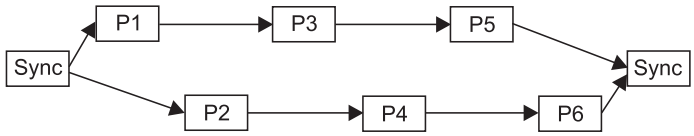


Figure 3.2: VRC with two threads and three frames per thread [160].

necessity for efficient motion vector coding in MDC of video [84, 85].

The approach of Kim and Lee [85] exploits *overlapped block motion compensation* (OBMC) [105], which generates a smoother motion field than the conventional block-based motion compensation. Blocks from the motion vector field are partitioned into two coarse fields using quincunx subsampling and are split between two descriptions. The decoder reconstructs a fine motion field if both descriptions are received, and a coarse motion field if only one description is received. The redundancy in this algorithm is minimal, which may cause high side distortions. The coder in [85] does not code the prediction mismatch between the encoder and decoder prediction loops. However, a combination with the 3-loop motion compensation scheme as in [115] is possible.

Another MDC approach generates two descriptions from a mesh-based motion compensation field [159]. Motion vectors are associated with the nodes of the mesh and then split between two descriptions using quincunx subsampling. If both descriptions are received, a fine motion mesh is generated. When one description is lost, a coarse motion mesh is generated.

3.1.3 Temporal subsampling

Motion information is treated by applying temporal subsampling in an approach known as *video redundancy coding (VRC)* [160]. Temporally subsampled frame sequences (*threads*) are formed and processed in parallel, thus becoming independently decodable. In a fixed time, the threads converge to the *Sync* frame. Even if some of the threads are damaged, the other threads can be used to correctly reconstruct the Sync frame. This minimizes the number of I-frames as there is no need for complete resynchronization. Figure 3.2 illustrates VRC with two threads and three frames per thread.

As each thread is a subsampled version of the original video along the

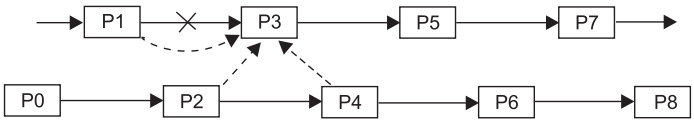


Figure 3.3: Lost frame reconstruction in balanced multi-state video coding [9].

temporal axis and the prediction is done independently of the other threads, the motion vectors estimated from two consecutive P-frames may be longer and the prediction error may be greater. This adds a substantial penalty to the coding efficiency. Another source of redundancy in VRC are the Sync frames coded multiple times [160]. In the H.264 video coding standard [74], one can use SP-frames instead of VRC Sync frames [79]. This would eliminate the drift caused by inaccurate frame reconstruction from different Sync frames.

The VRC approach has been adopted and further developed by Apostolopoulos [7] into the approach referred to as *multiple states*. The novelty of the Multiple States approach is that it uses the data from multiple streams to recover the lost state. The frame reconstruction process is illustrated in Figure 3.3. The lost frame can be reconstructed with sufficient quality to be used as a reference frame in the decoding of the stream it belongs to. Thus, the source video can be efficiently reconstructed to its full rate without using VRC Sync frames.

A *multiple description motion compensation* (MDMC) approach has also been motivated by VRC. MDMC reduces redundancy in VRC by utilizing a second-order predictor, i.e. by predicting the current frame from two previously decoded frames [152, 153]. A motion-compensated block of the frame i has two motion vectors. The first motion vector MV_1 points to the frame $i-1$. The second vector MV_2 points to the frame $i-2$. Both the central and side prediction errors are coded by DCT followed by the entropy coding [153]. When both descriptions are received, motion compensation is performed as the weighted sum of two compensated pictures. When only one description is received (e.g. containing odd frames), the decoder predicts the current frame from the previous frame of the same description. The mismatch between the predicted frames at the encoder and frames at the decoder is explicitly coded with a lower accuracy, following the three-prediction-loop approach [115] (see Sub-section 3.1.1). The predictor and the quantizer of the mismatch signal

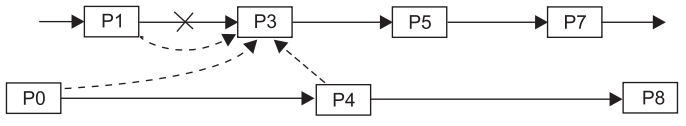


Figure 3.4: Lost frame reconstruction in unbalanced multi-state video coding [9].

can be adjusted to control the redundancy and reconstruction quality. Thus, unlike VRC, which has only few operational points in its redundancy rate-distortion region, MDMC operates over a wide range of redundancy.

A similar approach, called double-vector motion compensation (DMC), has been proposed in [25]. It uses a weighted superposition of two blocks in two previous frames without coding the mismatch signal.

3.1.4 Unbalanced multiple description coding

Most MDC approaches produce balanced descriptions ($D_0 < D_1,\ D_0 < D_2,\ D_1 = D_2,\ R_1 = R_2$). Thus, lower quality reconstruction can be obtained from each description separately and high quality reconstruction is obtained from both descriptions together. However, some applications may require unbalanced descriptions. An unbalanced MDC (UMDC) is especially beneficial when combined with multiple-path transport. In multi-path transport, each description is transmitted to the receiver over a separate link. If those links have different probabilities of packet losses, UMDC can improve the average reconstruction quality at the receiver [9]. In particular, UMDC assumes that one description has higher rate and is more important than the other description. An example of unbalanced video coding is given in Figure 3.4. The descriptions in this approach have different frame rate.

Another approach to UMDC gives high priority to one description, while the other description is considered as pure redundancy [34, 35, 133]. More specifically, the encoder creates a high resolution (HR) description and a low resolution (LR) description, where the central distortion is assumed equal to the higher quality side distortion ($D_0 = D_1 < D_2$). The low-quality description is obtained by spatial downsampling of the original sequence and used only when the high-resolution description is lost. An erasure recovery algorithm [50, 133] is used to estimate the lost samples in the HR sequence.

3.1.5 MD-FEC for video

Sub-section 2.1.3 has described how the MD-FEC scheme can be applied to the output of the SPIHT coder. MD-FEC has also been applied to the output of the MC-based video coder [51]. To generate a progressive bitstream, the output of the MC-based video coder is rearranged in such a way that more important information is transmitted in the beginning of the stream. The bitstream is split into groups of pictures (GOP) of a predefined size; each GOP is rearranged independently of the other GOPs. In the rearranged GOP, I-frame goes first and the other important data (such as header information and motion vectors) is transmitted next. Finally, VLC codewords from each block corresponding to k-th DCT coefficients are grouped together (the groups are ordered from low to high frequency content) [51]. MD-FEC algorithm is applied to the rearranged bitstream similarly to as it is described in Subsection 2.1.3. Nevertheless, reordering does not make the bitstream truly embedded. MD-FEC algorithm can also be used with data partitioning of the H.264 standard [74]. Quite naturally, MD-FEC can be applied to 3D wavelet coders such as 3D-SPIHT.

3.1.6 3D transform based coding

Yet another approach to video coding is *three-dimensional subband coding* (3D-SBC) using wavelet transforms. Motion compensation can be incorporated into 3D-SBC using motion-compensated lifting [128]. This implementation, called *motion-compensated temporal filtering* (MCTF), can be used with any wavelet kernel and any motion model; it also enables temporal scalability in video coding. In the MDC scheme based on MCTF, low-frequency frames are duplicated in both descriptions and high frequency frames are divided between two descriptions. When reconstructing from a single description, missing frames are estimated using motion vectors [125].

Another MD video coding method with a flexible number of descriptions is based on interframe wavelet-based scalable video coding [2]. Motion-compensated temporal filtering and 2D spatial wavelet filtering are used to perform the 3D wavelet transform. The MD algorithm generates N descriptions from the spatio-temporal code-blocks coded at M different rates. A code-block copy with a higher rate goes to one description while the redundant copies with lower rates are included in other descriptions to facilitate reconstruction in

case of description loss. Motion vectors and the lowest frequency code-blocks in both time and space are duplicated in all descriptions at a higher rate.

A number of other MD and error resilient video coders based on 3D-transforms (including wavelets, LOT, and DCT) have been proposed [86, 93, 135, 166].

3.2 2-stage 3D-DCT transform coding scheme

Nowadays, video is more often being encoded in mobile devices and transmitted over less reliable wireless channels. Traditionally, the objective in video coding has been to achieve high compression, which was attained with the cost of increasing encoding complexity. However, portable devices, such as camera phones, still lack enough computational power and are energy-consumption constrained. Besides, a highly compressed video sequence is more vulnerable to transmission errors, which are often present in wireless networks due to multi-path fading, shadowing, and environmental noise. Thus, there is a need of a low-complexity video coder with acceptable compression efficiency and strong error-resilience capabilities.

Lower computational complexity in transform-based video coders can be achieved by properly addressing the motion estimation problem, as the motion estimation is the most complex part of such coders. In case of high and moderate frame rates ensuring smooth motion, motion-compensated prediction can be replaced by a proper transform along the temporal axis to handle the temporal correlation between the frames in the video sequence. Thus, the decorrelating transform adds one more dimension, becoming a 3D one, and if a low complexity algorithm for such transform exists, savings in overall complexity and power consumption can be expected compared to traditional video coders [11, 23, 26, 124]. Discrete cosine transform (DCT) has been favored for its very efficient 1D implementations. As DCT is a separable transform, efficient implementations of 3D-DCT can be achieved too [20, 23, 124]. Previous research on this topic shows that a simple (baseline) 3D-DCT video encoder is three to four times faster than the optimized H.263 encoder [72], for the price of some compression efficiency loss, quite acceptable for portable devices [88].

A great number of multiple description (MD) video coders surveyed in Section 3.1 are based on MC prediction. However, MC-based MD video coders

risk having a mismatch between the prediction loops in the encoder and decoder when one description is lost. The mismatch can propagate further in the consequent frames if not corrected. The solutions of this problem include three separate prediction loops at the encoder [115], a separate prediction loop for every description [7, 48, 149], or prediction from a coarser reference frame [18]. However, all these approaches decrease the compression efficiency and the approach in [115] also leads to increased computational complexity and possibly to increased power consumption.

In this monograph, we investigate a two-stage multiple description coder based on 3D-transforms, denoted *3D-2sMDC*. This coder does not exploit motion compensation as initially proposed in [102] and developed further in [104]. Using 3D-transform instead of motion-compensated prediction reduces the computational complexity of the coder, meanwhile eliminating the problem of mismatch between the encoder and decoder. The proposed MD video coder is a generalization of our 2-stage image MD coding approach [103] to coding of video sequences [102, 104]. Designing the coder, we are targeting balanced computational load between the encoder and decoder, the ability to work at very low redundancy introduced by MD coding, and competitiveness with MD video coders based on motion-compensated prediction (H.263) in terms of compression ratio.

3.2.1 Encoder operation

In our scheme, a video sequence is coded in two stages as shown in Figure 3.5. In the first stage (dashed rectangle), a coarse sequence approximation, called *shaper*, is obtained and included in both descriptions. The second stage produces enhancement information, which has higher bitrate and is divided between two descriptions. The idea of the method is to get a coarse signal approximation which is the best possible for the given bitrate while decorrelating the residual sequence as much as possible.

The operation of the proposed encoder is described in the following. First, a sequence of frames is split into groups of 16 frames. Each group is split into 3D cubes of size $16 \times 16 \times 16$. 3D-DCT is applied to each cube. The lower-frequency DCT coefficients in the $8 \times 8 \times 8$ cube are coarsely quantized with quantization step Q_s and entropy-coded (see Figure 3.6(a)) composing the shaper, other coefficients are set to zero. Inverse quantization is applied

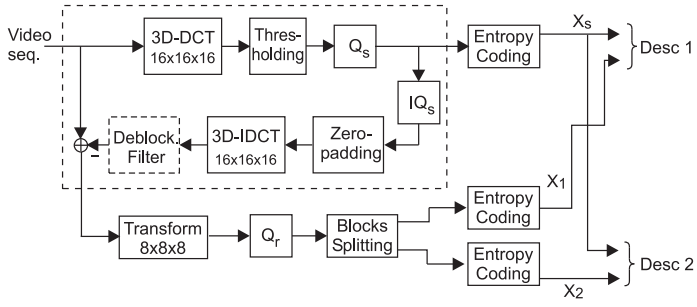
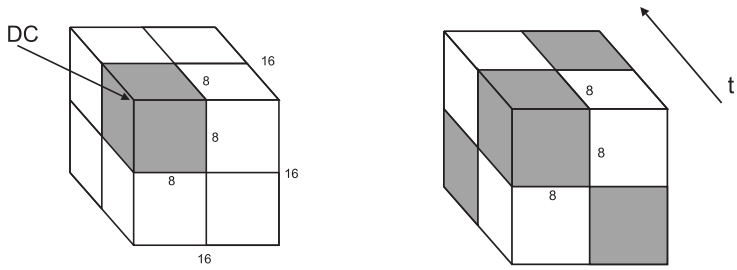


Figure 3.5: Proposed encoder scheme.



(a) (b)

Figure 3.6: Coding patterns: (a) 3D-DCT cube for shaper coding: only coefficients in the gray volumes are coded, other coefficients are set to zero; (b) Split pattern for volumes of a residual sequence: "gray" - *Description 1*; "white" - *Description 2*.

to these coefficients followed by the inverse 3D-DCT. An optional deblocking filter serves to remove the block edges in spatial domain. Then, the sequence reconstructed from the shaper is subtracted from the original sequence to get the residual sequence.

The residual sequence is coded by a 3D block transform (DCT or a hybrid transform) and transform coefficients are finely quantized with a uniform quantization step (Q_r), split into two parts in a manner shown in Figure 3.6(b), and entropy-coded. One part together with the shaper form *Description 1*, while the second part and the shaper form *Description 2*. Thus, each description consists of the shaper and half of the transform volumes of the residual sequence.

The shaper is included in both descriptions to facilitate successful reconstruction when one description is lost. Thus, the *redundancy* of the proposed coder is only determined by the shaper bitrate, which is controlled by the

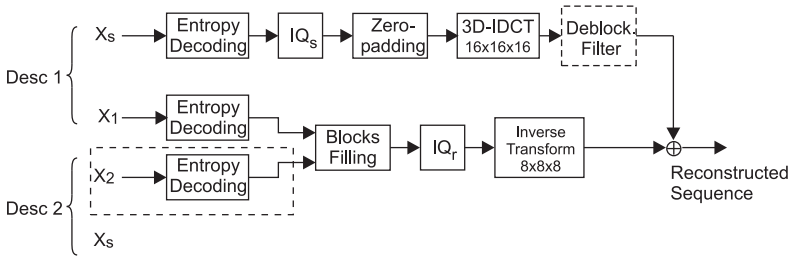


Figure 3.7: Decoder scheme. Central reconstruction. Side reconstruction (from *Description 1*) when the content of the dashed rectangle is removed.

shaper quantization step Q_s. A larger quantization step corresponds to lower level of redundancy and lower quality of side reconstruction (reconstruction from one description). Alternatively, a smaller quantization step results in higher quality side reconstruction. The *quality* of the two-description reconstruction is controlled by the quantization step Q_r used in coding the residual sequence. As the residual volumes are divided into two equal parts, the encoder produces two balanced descriptions both in terms of PSNR and bitrate.

3.2.2 Decoder operation

The decoder (Figure 3.7) operates as follows. When the decoder receives two descriptions, it extracts the shaper (X_s) from one of the descriptions. Then, the shaper is entropy-decoded and inverse quantization is applied. The $8\times8\times8$ volume of coefficients is zero-padded to the size $16 \times 16 \times 16$, and the inverse DCT is applied. The deblocking filter is applied if it was used in the encoder.

In case of *central* reconstruction (reconstruction from two descriptions), each part of the residual sequence (X_1 and X_2) is extracted from the corresponding description and entropy decoded. Then, volumes of the corresponding descriptions are decoded and combined together as in Figure 3.6(b). The inverse quantization and the inverse transform (inverse DCT or inverse hybrid transform) are applied to coefficients and the residual sequence is added to the shaper to obtain the reconstructed sequence.

The scheme of the side decoder (e.g. reconstructing from *Description 1*) can be obtained from Figure 3.7 if the content of the dashed rectangle is removed. In this case, the shaper is reconstructed from its available copy in *Description 1*. The residual sequence, however, has only half of the coefficient volumes (X_1). The missing volumes X_2 are simply filled with zeros. After

that, the decoding process is identical to that of the central reconstruction. As the residual sequence has only half of the coefficient volumes, the side reconstruction has lower, however, still acceptable quality. For example, the test sequence *Silent voice* coded at 64.5 kbps with 10% redundancy can be reconstructed with PSNR $D_0 = 31.49$ dB from two descriptions, and $D_1 = 26.91$ dB from one description (see Table 3.2).

3.3 Detailed system description

3.3.1 Coarse sequence approximation

The idea of the first coding stage is to concentrate as much information as possible into the shaper within strict bitrate constraints. We would also like to reduce artifacts and distortions appearing in the reconstructed coarse approximation. The idea is to reduce spatial and temporal resolution of the coarse sequence approximation in order to encode it more efficiently at lower bitrate [22]. Then, the original resolution sequence can be reconstructed by interpolation as a post-processing step. A good interpolation and decimation method would concentrate more information in the coarse approximation and correspondingly make the residual signal closer to white noise in a similar way to our 2-stage image coder described in Section 2.2. However, when designing a video coder for mobile devices, we also have to consider the computational complexity of the encoding process. A computationally inexpensive approach is to embed interpolation in the 3D-transform.

The downscaling factor for the shaper was chosen equal to two in both spatial and temporal directions. The proposed scheme is able to use other downscaling factors equal to powers of two. However, the factor 2 has been chosen as the one producing the best results for QCIF and CIF resolution. To reduce computational complexity, we combine downsampling with forward transform (and backward transform with interpolation). The original sequence is split into volumes of size $16 \times 16 \times 16$, and 3D-DCT is applied to each volume. Pruned DCT is used in this stage that allows to reduce computational complexity (see Figure 3.6(a)). The transform size of $16 \times 16 \times 16$ has been chosen as a compromise between the compression efficiency and computational complexity.

Only $8 \times 8 \times 8$ cubes of low-frequency coefficients in each $16 \times 16 \times 16$ co-

efficient volume are used; other coefficients are set to zero (see Figure 3.6(a)). The AC coefficients of the $8 \times 8 \times 8$ cube are uniformly quantized with quantization step Q_s. The DC coefficient is quantized with the quantization step Q_{DC}.

In the $8 \times 8 \times 8$ volume, we use coefficient scanning described in [164], which is similar to a 2-D zigzag scan. Although there exist more advanced types of quantization and scanning of 3D volumes [21, 26], we have found that simple scanning performs quite well. An optional deblocking filter can be used to reduce blocking artifacts caused by quantization and coefficient thresholding.

The DC coefficients of the transformed shaper volumes are coded by DPCM prediction. The DC coefficient of the volume is predicted from the DC coefficient of the temporally preceding volume. As the shaper is included in both descriptions, there is no mismatch between the states of the encoder and decoder when one description is lost.

First, the DC coefficient prediction errors and the AC coefficients undergo zero run-length encoding (RLE). It combines runs of successive zeros and the following non-zero coefficients into two-tuples where the first number is the number of leading zeros, and the second number is the absolute value of the first non-zero coefficient following the zero-run.

Variable-length encoding is implemented as a standard Huffman encoder similar to one in H.263 [72]. The codebook has the size 100 and is calculated for the two-tuples which are the output of RLE. All values exceeding the range of the codebook are encoded with the "escape" code followed by the actual value. Two different codebooks are used: one for coding the shaper and another for coding the residual sequence. Six codebooks are used in case of color video coding.

3.3.2 Residual sequence coding

The residual sequence is obtained by subtracting the reconstructed shaper from the original sequence. As the residual sequence consists of high-frequency details, we do not add any redundancy at this stage. The residual sequence is split into groups of 8 frames in such a way that two groups of 8 frames correspond to one group of 16 frames obtained from the coarse sequence approximation. Each group of 8 frames undergoes the 3D block transform. The transform coefficients are uniformly quantized with the quantization step Q_r

and split between two descriptions in a pattern shown in Figure 3.6(b).

Two different transforms are used in this work to code the residual sequence. The first transform is 3D-DCT and the second is a hybrid transform. The latter consists of the lapped orthogonal transform (LOT) [92] in vertical and horizontal directions, and DCT in temporal direction. Both DCT and the hybrid transform produce $8 \times 8 \times 8$ volumes of coefficients, which are split between two descriptions. Using LOT in spatial domain smoothes blocking artifacts when reconstructing from one description. In this case, LOT spatially spreads the error caused by loosing transform coefficient blocks. One can also apply LOT in along the temporal direction in order to reduce blocking artifacts in the temporal domain. However, in our coder, we avoid using LOT in temporal direction because of the additional delay it introduces in the encoding process. The calculation of the residual sequence coefficients requires first to process the temporally subsequent block. Namely, the encoder would have to wait until the subsequent batch of 16 frames is obtained, then to calculate and interpolate the shaper in order to obtain the residual sequence. This would increase the memory requirements of the scheme as the encoder would have to keep M residual frames preceding to the group of frames being processed, where M is the overlap parameter in LOT. For this reasons, we use the hybrid transform with DCT in temporal domain. Blocking artifacts in temporal domain are smoothed by optional post-filtering at the decoder.

As will be demonstrated in Section 3.7, the hybrid transform outperforms DCT in terms of PSNR and visual quality. Moreover, using LOT in spatial dimensions gives better visual results compared to DCT. However, blocking artifacts introduced by coarse coding of the shaper are not compensated by coding the residual sequence with the hybrid transform. Therefore, the *deblocking filter* is applied to the reconstructed shaper (see Figure 3.5) prior to subtracting it from the original sequence. In the experiments, we use the deblocking filter from the H.263+ standard [72].

In the residual sequence coding, the transform coefficients are uniformly quantized with the quantization step Q_r. DC prediction is not used in the second stage to avoid the mismatch between the states of the encoder and decoder if one description is lost. The scanning of coefficients is 3D-zigzag scanning [164]. The entropy coding is RLE followed by Huffman coding with the codebook different from the one used for coding the coarse sequence approximation.

3.4 Scheme analysis

3.4.1 Redundancy and reconstruction quality

Denote by D_0 the *central distortion* (distortion when reconstructing from two descriptions), and by D_1 and D_2 the *side distortions* (distortions when reconstructing from one description). In case of balanced descriptions, $D_1 = D_2$. Denote as D_s the distortion of the video sequence reconstructed from the shaper. Consider 3D-DCT coding of the residual sequence. The side distortion D_1 is formed by the blocks, half of which are coded with the distortion D_0, and half with the shaper distortion D_s. Similar to (2.10), we can write

$$D_1 = \frac{1}{2}(D_s + D_0). \tag{3.1}$$

Expression (3.1) can also be used when the hybrid transform is used for coding the residual. As LOT is by definition an orthogonal transform, mean-squared error distortion in spatial domain is equal to the distortion in the transform domain. The side distortion in the transform domain is determined by loss of half of residual transform blocks. Thus, expression (3.1) is also valid for the hybrid transform. The shaper's distortion D_s depends on the bitrate R_s allocated to the shaper. Then, we can write (3.1) as

$$D_1(R_s, R_r) = \frac{1}{2}(D_s(R_s) + D_0(R_r, R_s)) \tag{3.2}$$

where R_r is the bitrate allocated for coding the residual sequence and R_s is the bitrate allocated to the shaper. For higher bitrates, $D_s(R_s) \gg D_0(R_r)$, and D_1 mostly depends on R_s.

The redundancy ρ of the proposed scheme is the bitrate allocated to the shaper, $\rho = R_s$. The shaper bitrate R_s and the side reconstruction distortion D_1 depend on the quantization step Q_s and the characteristics of the video sequence. The central reconstruction distortion D_0 is mostly determined by the quantization step Q_r.

Thus, the encoder has two control parameters: Q_s and Q_r. By changing Q_r, the encoder controls the central distortion. By changing Q_s, the encoder controls the redundancy and the side distortion.

3.4.2 Optimization

The proposed scheme can be optimized for changing channel behavior. Denote by p the probability of the packet loss and by R the target bitrate. Then, in case of balanced descriptions we have to minimize

$$2p(1-p)D_1 + (1-p)^2 D_0 \tag{3.3}$$

subject to

$$2R_s + R_r \leq R. \tag{3.4}$$

Taking into consideration (3.1), expression (3.3) can be transformed to the unconstrained minimization task with the Lagrangian

$$\begin{aligned} L(R_s, R_r, \lambda) = p(1-p)(D_s(R_s) + D_0(R_s, R_r)) \\ + (1-p)^2 D_0(R_s, R_r) + \lambda(2R_s + R_r - R). \end{aligned} \tag{3.5}$$

It is not feasible to find the distortion-rate functions $D_0(R_s, R_r)$ and $D_s(R_s)$ in real-time to solve the optimization task. Instead, the distortion-rate (D-R) function of a 3D-coder can be modeled as

$$D(R) = b2^{-aR} - c \tag{3.6}$$

where a, b, and c are parameters, which depend on the characteristics of the video sequence. Hence,

$$D_s(R_s) = b2^{-aR_s} - c. \tag{3.7}$$

Assuming that the source is successfully refinable with respect to the squared-error distortion measure (e.g., this is true for i.i.d Gaussian source [44]) we can write

$$D_0(R_s, R_r) = b2^{-a(R_s+R_r)} - c. \tag{3.8}$$

Then, substituting (3.7) and (3.8) into (3.5) and differentiating the resulting Lagrangian with respect to R_s, R_f, and λ, we can find a closed form solution of the optimization task (3.5). The obtained optimal R_s and R_r are

$$\begin{aligned} R_s^* &= \frac{1}{2}R + \frac{1}{2a}\log_2(p), \\ R_r^* &= -\frac{1}{a}\log_2(p) \end{aligned} \tag{3.9}$$

where R_s^* and R_r^* are the rates of the shaper and the residual sequence, respectively.

Hence, the optimal redundancy ρ^* of the proposed scheme is

$$\rho^* = R_s^* = \frac{1}{2}R + \frac{1}{2a}\log_2(p). \tag{3.10}$$

The optimal redundancy ρ^* depends on the target bitrate R, the probability of packet loss p, and parameter a of the source D-R function. It does not depend on D-R parameters b and c. We have found that the parameter a usually takes similar values for video sequences with the same resolution and frame rate. Thus, one does not need to estimate a in real-time. Instead, one can use a typical value of a to perform optimal bit allocation during encoding. For example, sequences with CIF resolution and 30 frames per second usually have the value of a between 34 and 44 for bitrates under 1.4 bits per pixel.

One can notice that for values $R \leq -\frac{1}{a}\log_2(p)$, the optimal redundancy ρ^* is zero or negative. For these values of R and p, the encoder should not use MDC. Instead, single description coding has to be used. It is seen from (3.10) that the upper limit for redundancy is $R/2$, which is obtained for $p = 1$. That means that all the bits are allocated to the shaper, which is duplicated in both descriptions.

3.5 Computational complexity

To perform a 3D-DCT of an $N \times N \times N$ cube, one has to perform $3N^2$ one-dimensional DCTs of size N. However, if one needs only the $N/2 \times N/2 \times N/2$ low-frequency coefficients, as in the case of the shaper coding, a smaller amount of DCTs need to be computed. Three stages of separable row-column-frame (RCF) transform require $[N^2 + 1/2N^2 + 1/4N^2] = 1.75N^2$ DCTs for one cube. The same is true for the inverse transform.

The encoder needs only the 8 lowest coefficients of 1D-DCT. For this reason, we use pruned DCT as in [127]. The computation of the 8 lowest coefficients of the pruned DCT II [112] of size 16 requires 24 multiplications and 61 additions [127]. That gives 2.625 multiplications and 6.672 additions per point and brings significant reduction in computational complexity. For comparison, full separable DCT II (decimation in frequency (DIF) algorithm) [112] of size 16 would require 6 multiplications and 15.188 additions per point.

Transform	Pruned $16 \times 16 \times 16$	3D VR $16 \times 16 \times 16$	RCF $16 \times 16 \times 16$	3D VR $8 \times 8 \times 8$	RCF $8 \times 8 \times 8$
Mults/point	2.625	3.5	6	2.625	4.5
Adds/point	6.672	15.188	15.188	10.875	10.875
$\frac{\text{Mults+adds}}{\text{point}}$	9.297	18.688	21.188	13.5	15.375

Table 3.1: Operations count for 3D-DCT II. Comparison of algorithms.

The operation count for different 3D-DCT schemes is provided in Table 3.1. The adopted "pruned" algorithm is compared to fast *3D vector-radix decimation-in-frequency DCT* (3D VR DCT) [20] and *row-column-frame* (RCF) approach, where 1D-DCT is computed by DIF algorithm [112]. One can see that the adopted "pruned" algorithm has the lowest computational complexity. In terms of operations per pixel, partial DCT $16 \times 16 \times 16$ is less computationally expensive than the full $8 \times 8 \times 8$ DCT used to code the residual sequence.

In [88], a baseline 3D-DCT encoder is compared to the optimized H.263 encoder [165]. It was found [88] that the baseline 3D-DCT encoder is up to four times faster than the optimized H.263 encoder (3.7 times on average). In the baseline 3D-DCT encoder [88], DCT was implemented by RCF approach, which gives 15.375 operations/point. In our scheme, the forward pruned 3D-DCT for the shaper requires only 9.3 op/point. Adding the inverse transform, one gets 18.6 op/points. The $8 \times 8 \times 8$ DCT of the residual sequence can be implemented by the 3D VR DCT [20], which requires 13.5 op/point. Thus, the overall complexity of the transforms used in the proposed encoder is estimated as 32.1 op/point that is twice higher than the complexity of the transforms used in the baseline 3D-DCT encoder (15.375 op/point).

The overall computational complexity of the encoder includes quantization and entropy coding of the shaper coefficients. However, the number of coefficients coded in the shaper is eight times lower than the number of coefficients in the residual sequence as only 512 lower DCT coefficients in each $16 \times 16 \times 16$ block are coded. Thus, quantization and entropy coding of the shaper would take about 8 times less computations than quantization and entropy coding of the residual sequence. Thus, we estimate that the overall complexity of the proposed encoder is not more than twice the complexity of the baseline 3D-DCT encoder [88]. This means that the proposed coder has up to two times (factor of 1.85 on average) lower computational complexity than the

optimized H.263 [165]. The difference in computational complexity between the proposed coder and H.263+ with scalability (to provide error resilience analogous to the proposed method) is even greater. Being less computationally complex, the proposed coder has single description performance similar or even higher than H.263+ [72] with SNR scalability, as shown in Section 3.7.

3.6 Packetization and transmission

The bitstream of the proposed video coder is packetized as follows. A group of pictures (16 frames) is split into 3D-volumes of size $16 \times 16 \times 16$. One packet should contain one or more shaper volumes, which gives 512 entropy-coded coefficients (due to thresholding).

In case of single description coding, one shaper volume is followed by eight spatially corresponding volumes of the residual sequence, which have the size $8 \times 8 \times 8$. In case of multiple description coding, a packet from *Description 1* contains a shaper volume and four residual volumes taken in the pattern shown in Figure 3.6(b). *Description 2* contains the same shaper volume and four residual volumes, which are not included into *Description 1*. If the size of such group (one shaper volume and four residual volumes) is small, several groups are packed into one packet.

The proposed coder uses DPCM prediction of DC coefficients in the shaper volumes. The DC coefficient is predicted from the DC coefficient of the temporally preceding volume. If both descriptions containing the same shaper volume are lost, DC coefficient is estimated as the previous DC coefficient in the same spatial location or as an average of DC coefficients of the spatially adjacent volumes. This concealment may introduce a mismatch in the DPCM loop between the encoder and decoder. However, the mismatch does not spread out of the border of this block. The mismatch is corrected by the DC coefficient update which can be requested over the feedback channel or may be done periodically.

To further improve the robustness against burst errors, the bitstream can be reordered in a way that descriptions corresponding to one 3D volume are transmitted in the packets which are not consecutive. It will decrease the probability that both descriptions are lost due to consequent packet losses. Another solution to improve the error resilience is to send the packets of *De-*

scription 1 over one link, and packets of *Description 2* over another link.

3.7 Simulation results

This section presents the comparison of the proposed MD coder with other MD coders. The experiments are performed on sequences *Tempete* (CIF, 30 fps, 10 s), *Silent voice* (QCIF, 15 fps, 10 s), and *Coastguard* (CIF, 30 fps). We measure the reconstruction quality by using the *peak signal-to-noise ratio* (PSNR). The distortion is the average luminance PSNR over time, all color components are coded. We compare our scheme mainly with H.263-based coders as our goal is low-complexity encoding. Apparently, the proposed scheme cannot compete with H.264 in terms of compression efficiency. However, the H.264 encoder is much more computationally complex than the proposed scheme. It is difficult to compare the computation complexity of the H.264 encoder with that of our encoder as we can talk only about the particular realization of the encoders on the same platform. However in [89], the computational complexity of the optimized H.264 encoder (actually the H.26L prototype corresponding to the H.264 baseline profile with one reference frame) has been found 3.8 times higher than that of analogously optimized H.263. Therefore, we could estimate the computational complexity of our coder as at least 7 times lower than the complexity of the H.264 baseline profile. In the subsequent subsections, we mostly compare the proposed coder with H.263, which has the complexity comparable with our coder.

3.7.1 Single description performance

Figure 3.8 plots PSNR versus bitrate for the sequence *Tempete*. The compared coders are single description coders. *3D-2stage* coder is a single-description variety of the coder described above. The shaper is sent only once, and the residual sequence is sent in a single description. *3D-DCT* is a simple 3D-DCT coder described in [26, 88]. *H.263* is the Telenor implementation of H.263. *H.263-SNR* is the H.263+ with SNR scalability, implemented at the University of British Columbia [36, 119]. We also show the results for the H.264 coder (baseline profile) using one reference frame. The encoder used is the JM 11.0. We can observe that H.264 outperforms all other coders. However, one should also take into account that the H.264 encoder, even the baseline profile, is

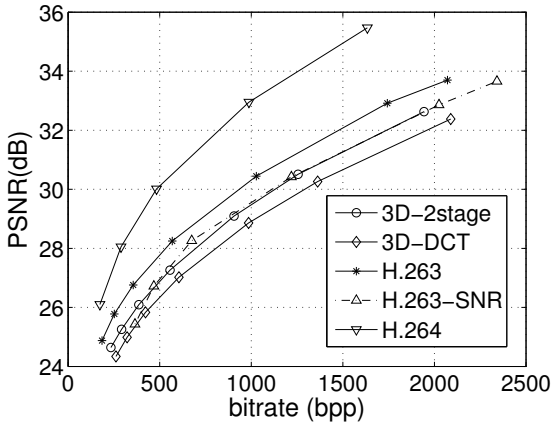


Figure 3.8: Sequence *Tempete*, single description coding.

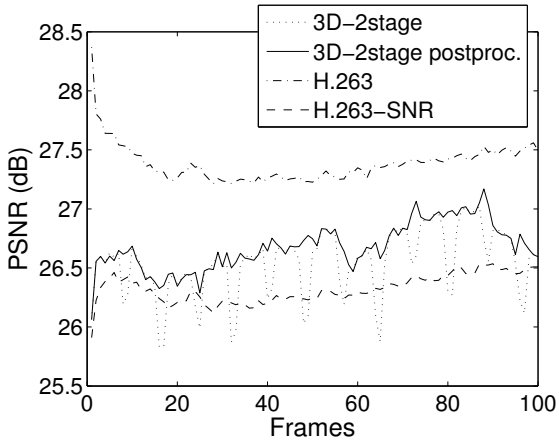


Figure 3.9: Sequence *Tempete* coded at 450 kbps, single description coding.

more computationally complex than all the other encoders compared. From the remained coders, the best performance is shown by H.263. Our 3D-2stage has approximately the same performance as H.263+ with SNR scalability and looses half to one dB to H.263+. Simple 3D-DCT coder shows the worst performance from all the compared coders.

Figure 3.9 shows PSNR of the first 100 frames of *Tempete* sequence. The sequence is encoded to target bitrate 450 kbps. Figure 3.9 demonstrates that 3D-DCT coding exhibits temporal degradation of quality on the borders of 8-frame blocks. These temporal artifacts are caused by block-wise DCT and perceived like abrupt movements. These artifacts can be efficiently concealed with

postprocessing on the decoder side. In this experiment, we applied MPEG-4 deblocking filter [87] to block borders in the *temporal* domain. As a result, temporal artifacts are smoothed. The perceived quality of the video sequence has also improved. Some specialized methods for deblocking in temporal domain can be applied as in [121]. Postprocessing in temporal and spatial domain can also improve the reconstruction quality in case of description loss. In the following experiments, we do not use postprocessing in order to have fair comparison with other MDC methods.

3.7.2 Performance of different residual coding methods

In the following, we compare the performance of MD coders in terms of side reconstruction distortion, while they have the same central distortion. Three variants of the proposed 3D-2sMDC coder are compared. These MD coders use different schemes for coding the residual sequence. *Scheme 1* is the 2-stage coder, which uses hybrid transform for the residual sequence coding and deblocking filtering of the shaper. *Scheme 2* employs DCT for coding the residual sequence. *Scheme 3* is similar to Scheme 2 except that it uses the deblocking filter (see Figure 3.5). We have compared these schemes with a simple MD coder based on a simple 3D-DCT video coder and MDSQ [145], which we call *3D-MDSQ*. MDSQ is applied to the first N coefficients of $8\times8\times8$ 3D-DCT cubes. Then, MDSQ indices are sent to the corresponding descriptions, and the rest of $512 - N$ coefficients are split between two descriptions (even coefficients go to *Description 1* and odd coefficients to *Description 2*).

Figure 3.10 shows the result of side reconstruction for the reference sequence *Tempete*. The average central distortion (reconstruction from both descriptions) is fixed for all encoders, $D_0 = 28.3$ dB. The mean side distortion (reconstruction from one description) versus bitrate is compared. One can see that Scheme 1 outperforms other coders, especially in the low-redundancy region. One can also see that deblocking filtering of the shaper (Scheme 3) does not give much advantage for the coder using 3D-DCT for coding the residual sequence. However, deblocking filtering of the shaper is necessary in Scheme 1 as it significantly enhances the visual quality. Deblocking filtering in the proposed coder requires twice less operations than in H.263+ because the block size in the shaper is twice larger than that in H.263+. All three proposed coders outperform the simple 3D-MDSQ coder to the extent of 2 dB.

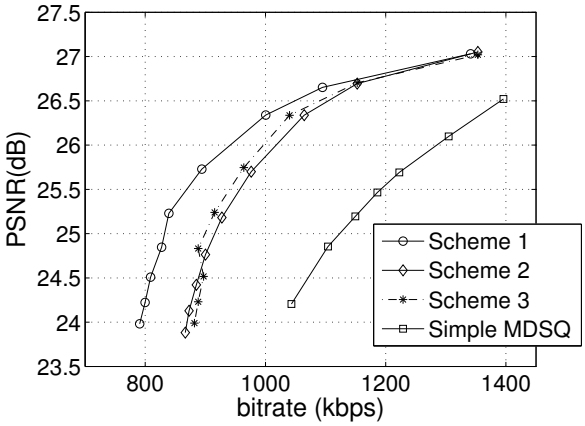


Figure 3.10: Sequence *Tempete*, 3D-2sMDC, mean side reconstruction. $D_0 \approx 28.3$ dB.

3.7.3 Network performance of the proposed method

Figure 3.11 introduces performance of the proposed coder in the network environment with error bursts. In this experiment, bursty packet loss behavior is simulated by the two-state Markov model. These two states are G (good) when the packets are correctly received and B (bad) when the packets are either lost or delayed. This model is fully described by transition probabilities p_{BG} from state B to state G and p_{GB} from G to B. The model can also be described by average loss probability $P_{\mathrm{B}} = \Pr(\mathrm{B}) = p_{\mathrm{GB}}/(p_{\mathrm{GB}} + p_{\mathrm{BG}})$ and the average burst length $L_{\mathrm{B}} = 1/p_{\mathrm{BG}}$.

In the following experiment, the sequence *Tempete* (CIF, 30 fps) has been coded to bitrate 450 kbps into packets not exceeding the size of 1000 bytes for one packet. The coded sequence is transmitted over two channels modeled by two-state Markov models with $P_{\mathrm{B}} = 0.1$ and $L_{\mathrm{B}} = 5$. Packet losses in *Channel 1* are uncorrelated with errors in *Channel 2*. Packets corresponding to *Description 1* are transmitted over *Channel 1* and packets corresponding to *Description 2* are transmitted over *Channel 2*. Two channels are used in order to guarantee that packet losses are not correlated between the descriptions. Similar results can be achieved by packet interleaving. When both descriptions are lost, error concealment described in Section 3.6 is used. Optimal redundancy for *Tempete* sequence found with (3.10) for bitrate 450 kbps (0.148 bpp) is 21%.

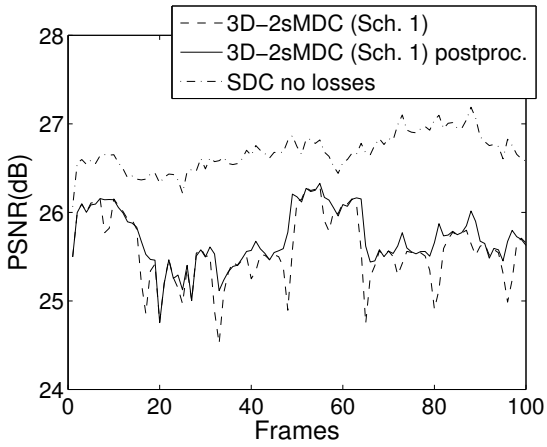


Figure 3.11: Network performance, packet loss rate 10%. Sequence *Tempete*, coded at 450 kbps. Comparison of 3D-2sMDC and 3D-2sMDC with posfiltering. Performance of single description coder without losses is given as reference.

Figure 3.11 shows network performance of 3D-2sMDC and *3D-2sMDC with postrocessing (temporal deblocking)*. The performance of a single description 3D-2stage coder in a lossless environment is also given in Figure 3.11 as a reference. One can see that using MDC for error resilience helps to maintain an acceptable level of quality when transmitting over the network with packet losses.

3.7.4 Comparison with other MD coders

The next set of experiments is performed on the reference sequence *Coastguard* (CIF, 30 fps). The first coder is the proposed 3D-2sMDC coder Scheme 1. The *H.263 spatial* method exploits H.263+ [119] to generate layered bitstream. The base layer is included in both descriptions while the enhancement layer is split between two descriptions on a GOB basis. The *H.263 SNR* is similar to the previous method with the difference that it uses SNR scalability to create two layers.

Figure 3.12 plots the single description distortion versus bitrate of the *Coastguard* sequence for the three coders described above. The average central distortion is $D_0 = 28.5$ dB. One can see that 3D-2stage method outperforms the other two methods.

The results indicate that the proposed MD coder based on 3D-transforms outperforms the simple MD coders based on H.263+. For the coder with SNR

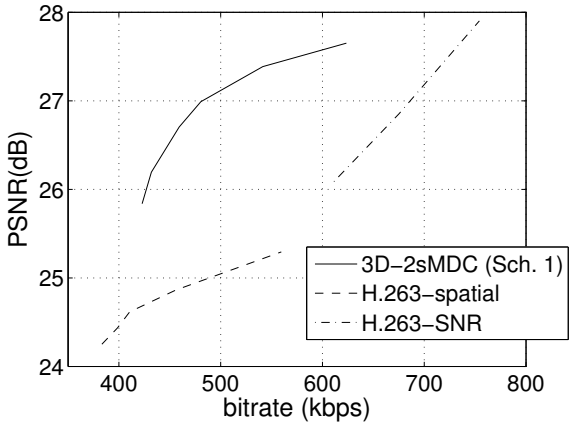


Figure 3.12: Sequence *Coastguard*, mean side reconstruction. $D_0 \approx 28.50$ dB.

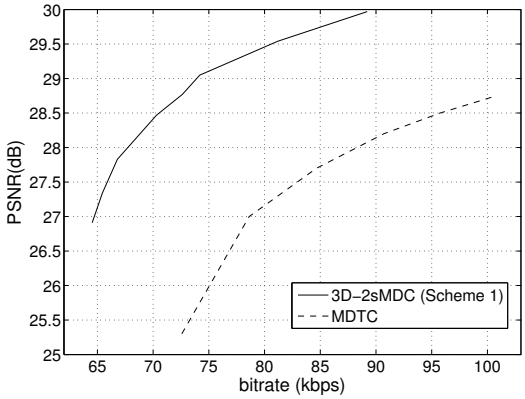


Figure 3.13: Sequence *Silent voice*, mean side reconstruction. $D_0 \approx 31.53$ dB.

scalability, we were not able to get the bitrates as low as we have got with our 3D-2stage method.

Another set of experiments is performed on the reference sequence *Silent voice* (QCIF, 15 fps). The proposed 3D-2sMDC coder is compared with MDTC coder that uses three prediction loops in the encoder [115, 116]. The 3D-2sMDC coder exploits Scheme 1 as in the previous set of experiments. The rate-distortion performance of these two coders is shown in Figure 3.13. The PSNR of two-description reconstruction of 3D-2sMDC coder is $D_0 \cong 31.47 - 31.57$ dB and central distortion of MDTC coder is $D_0 = 31.49$ dB.

The results show that the proposed 3D-2sMDC coder outperforms the

MDTC coder, especially in a low-redundancy region. The superior side reconstruction performance of our coder can be explained by the following. MC-based multiple description video coder has to control the mismatch between the encoder and decoder. It can be done, for example, by explicitly coding the mismatch signal, as in [115, 116]. In opposite, MD coder based on 3D-transforms does not need to code the residual signal, thus, gaining advantage of very low redundancies (see Table 3.2). The redundancy in Table 3.2 is calculated as the additional bitrate for MD coder comparing to the single description 2-stage coder based on 3D block transforms.

A drawback of our coder is relatively high delay. High delays are common for coders exploiting 3D-transforms (e.g., coders based on 3D-DCT or 3D-wavelets). Waiting for 16 frames to apply the 3D transform introduces additional delay of slightly more than half a second for the frame rate 30 fps and about one second for 15 fps. The proposed coder also needs larger memory than an MC-based video coder, as it needs to store the 16 frames in the buffer before applying the DCT. This property is common for most 3D transform video coders. We suppose that most mobile devices nowadays have enough memory to perform the encoding.

Figures 3.14 and 3.15 show frame 13 of the reference sequence *Tempete* reconstructed from two descriptions and from *Description 1* alone. The sequence is coded by 3D-2sMDC (Scheme 1) encoder to bitrate $R = 880$ kbps.

Comparing Figure 3.15(a) reconstructed from two descriptions with Figure 3.15(b) reconstructed from one description, one can notice two types of visual artifacts. The first type of artifacts is characterized by blurred areas caused by lost residual sequence volumes. These artifacts can be noticed when comparing the green leaves in Figures 3.15(a) and 3.15(b). Another type of artifacts appears in fast moving objects like the gray leaf in the upper-right corner of Figures 3.15(a) and 3.15(b). The fast moving object is not present in the shaper due to thresholding the transform coefficients in the temporal domain. Therefore, when reconstructing from one description, the checkerboard-like pattern is observed (see Figure 3.15(b)) because of lost residual sequence volumes. This problem could be solved by not scaling the shaper in the temporal direction. However, this would increase the coding redundancy due to greater size of the shaper. One can also see from Figure 3.15 that there are no spatial blocking artifacts when using LOT for coding the residual sequence. One can notice some ringing artifacts in Figures 3.15(c) and 3.15(d). Although the

Central PSNR (dB)	Mean-side PSNR (dB)	Bitrate (kbps)	Redundancy (%)
31.49	26.91	64.5	9.8
31.51	27.34	65.5	11.4
31.51	27.83	66.8	13.7
31.57	28.47	70.3	19.6
31.52	29.05	74.2	26.3
31.47	29.54	81.2	38.2
31.53	29.97	89.2	51.8

Table 3.2: Reconstruction results. Sequence *Silent voice*.

image reconstructed from one description has the artifacts described above, the overall picture is smooth and pleasant to the eye.

3.8 Chapter conclusions

A general problem connected with multiple description video coders based on motion-compensated prediction is the mismatch between the states of the encoder and decoder when the description is lost. This problem can be solved either by generating extra prediction loops for any potential loss of description, by embedding the prediction in each description separately, or by using less efficient prediction from the lower quality reference frame.

We have proposed an MDC scheme for video which does not use motion-compensated prediction. The proposed coder exploits 3D transforms to remove correlation in video sequence. Coding is done in two stages: the first stage produces coarse sequence approximation (shaper) trying to fit as much information as possible in the limited bit budget. The second stage encodes the residual sequence, which is the difference between the original sequence and the shaper-reconstructed one. The shaper is obtained by pruned 3D-DCT, and the residual signal is coded by 3D-DCT or hybrid 3D-transform. The redundancy is introduced by including the shaper in both descriptions. The amount of redundancy is easily controlled by the shaper quantization step. The scheme can also be easily optimized for suboptimal bit allocation. This optimization can run in real time during the encoding process.

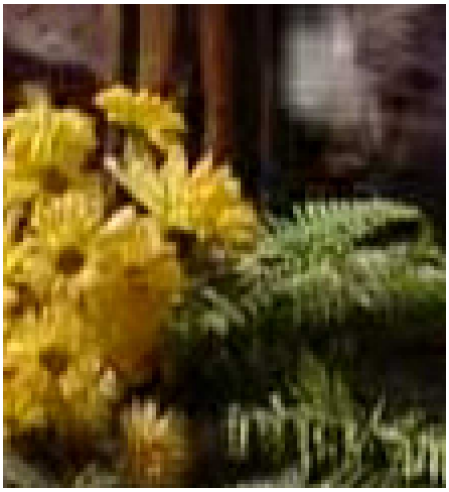
The proposed MD video coder has low computational complexity, which makes it suitable for mobile devices with low computational power and limited battery life. The coder has been shown to outperform MDTC video coder and

some simple MD coders based on H.263+. The coder performs especially well in the low-redundancy region. The encoder is also less computationally expensive than the H.263 encoder.

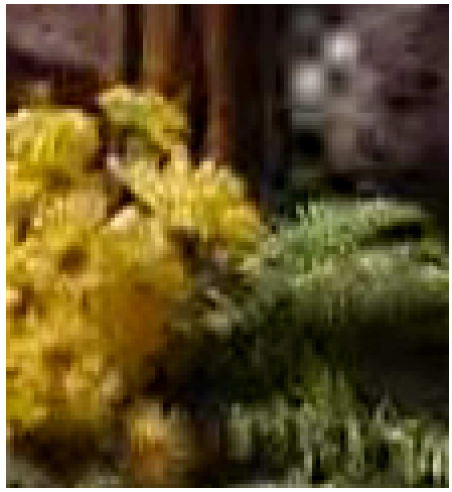
(a) Reconstruction from two descriptions, $D_0 = 28.52$.

(b) Reconstruction from *Description 1*, $D_1 = 24.73$.

Figure 3.14: Sequence *Tempete*, frame 13.



(a) Two descriptions



(b) *Description 1*

(c) Two descriptions

(d) *Description 1*

Figure 3.15: Sequence *Tempete*, frame 13. Magnified areas of frame in Figure 3.14.

Chapter 4

Multiple Description Coding of Stereo Video

In this chapter, we introduce some recent methods developed for multiple description coding of stereoscopic video. A straightforward way to compress the stereo sequence is *simulcast coding*, where the left-eye view and the right-eye view are encoded independently (see Figure 4.1). However, this approach does not exploit efficiently the correlation between two sequences. In order to exploit this correlation, one can use the *joint* coding structure, as shown in Figure 4.2. In the joint coding scheme, prediction in the left sequence employs only motion estimation and prediction in the right sequence employs both motion and disparity estimation.

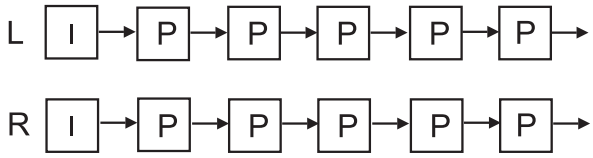


Figure 4.1: Reference structure of *simulcast* stereo video coding. Left- and right-view sequences are encoded independently.

Chapter 3 addressed multiple description coding of monoscopic video. Similar ideas can be applied to MD coding of stereoscopic video. Compressed stereoscopic video is also vulnerable to transmission errors. Moreover, due to more complicated structure of the prediction path, errors in the left sequence can propagate to the subsequent left frames and also to the right frames. Designing an MDC scheme for stereoscopic video, one has to keep in mind

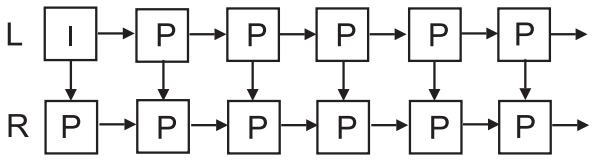


Figure 4.2: Reference structure of *joint* stereoscopic video coding. Left sequence is coded independently; frames of right sequence are predicted from either right or left frames.

that the depth information is essential in stereoscopic video and should be maintained in each description.

This chapter is organized as follows. Section 4.1 surveys state-of-the-art approaches to stereoscopic video coding and standardization activities in this area. Section 4.2 briefly describes a multi-view video coder which has been used to build the proposed MDC schemes. Section 4.3 presents the spatial scaling-based approach to MDC of stereoscopic video, whereas Section 4.4 introduces the approach based on temporal subsampling. Section 4.5 discusses the experimental results and offers a criterion for switching between the proposed schemes. Section 4.6 concludes the chapter.

4.1 Survey of stereo video coding

In stereo video, a 3D scene is captured by two cameras positioned at viewpoints with the distance between them corresponding to the distance between the human eyes. As the video sequences from two cameras share common content, better compression ratio can be achieved when encoding the sequences jointly in comparison to simulcast coding. This adds the fourth (the inter-view) compression dimension to video coding. Thus, the efficient scheme for stereo video compression exploits the combination of temporal and inter-view prediction. However, the gain in compression efficiency of the joint scheme compared with simulcast coding is usually limited because temporally neighboring pictures are typically more similar than spatially neighboring pictures.

The research on stereo video compression has continued into several directions, including optimum joint bit allocation for two channels or designing more efficient prediction structures. A joint coding method shown in Figure 4.2 has been developed further by interpolating the right sequence frames from the motion- and disparity-compensated frames. This makes it possible to achieve

the increased compression performance. In [43], frames of the right sequence are predicted as a weighted sum of the motion- and disparity-compensated frames. In [162], the B-frames in the right sequence are not transmitted to the decoder. Instead, these frames are interpolated at the decoder from the I- and P-frames of the right sequence and frames of the left sequence. This makes it possible to encode stereo video with only 8 to 30 % additional bitrate over a single-view stream. In [39], a stereo video coding system employs block prediction from motion- and disparity-compensated blocks by choosing the appropriate weighting function. Fast ME algorithm for the right sequence and mode pre-decision before ME decrease the system complexity. A stereoscopic H.264-based video coder based on the joint prediction scheme has been developed in [12]. This coder has been further improved in [13] by adding the parallel encoding of left and right video sequences in order to increase the encoding speed on the parallel virtual machine architecture. The prediction of the right sequence frames has also been improved by using more reference frames from the left sequence (including the subsequent frames). Apart from the methods mentioned above, a compression method based on the *human visual system* (HVS) has been developed in [109]. This compression method is referred to as hybrid HVS coding and is not compatible with video coding standards. In this system, the left sequence is coded with the lower quality by a hybrid 3D-/2D-DCT coder based on HVS model. The right sequence frames are encoded with higher quality by disparity compensation and coding the difference frames.

A predictive stereo video encoding system consists of several stages such as motion and disparity estimation and compensation, transform, quantization, and entropy coding. Using bit allocation and quantization strategies optimized for stereo coding increases the overall system performance. Several stereo image and video coding optimization schemes have been developed [41, 161]. These approaches minimize the total distortion measure while staying within the overall bit budget for both the left and right pictures.

The research on stereo video coding has resulted in video coding standards. A standard specification H.262 / MPEG-2 video, the Multiview Profile [73] has been developed [134]. In this standard, the left video sequence is encoded with standard MPEG-2 without reference to the right sequence frames. Therefore, the left-view bitstream can be separately decoded to display conventional video providing backward compatibility with the Main Profile of

MPEG-2. The right view exploits both the inter-view prediction and temporal prediction (similar to Figure 4.2). The standardization efforts have later been directed to *multi-view video coding* (MVC), which has recently received a lot of attention. ISO/MPEG and ITU/VCEG decided developing a dedicated MVC specification [151]. This standard will be an extension of H.264/AVC.

An alternative to classical stereo video coding is to transmit a video signal and a per sample depth map. From the video and depth information, the decoder can render a stereo pair [45,46]. The depth range is linearly quantized thus specifying a depth map. The depth map can be treated as a grey scale image. In [144], two methods were evaluated for coding the isolated depth maps using the JPEG coding technique and a wireframe model for coding the depth map information. In case of stereo video coding, a sequence of depth map images can be processed by any state-of-the-art video codec.

A number of depth/disparity coding techniques have been evaluated in the European ATTEST project [46]. The results have shown that even very low bitrate coding of depth images can produce stereoscopic images with acceptable quality since the coding distortions of the depth images do not affect so much the synthesized views. These research efforts have resulted in a new approach for 3DTV [46]. The approach uses a layered bitstream syntax. A base layer contains conventional single-view video encoded with MPEG-2 and can be processed by any MPEG-2 compatible decoder. An advanced layer carries the encoded depth map. Stereoscopic systems may decode the depth stream and generate a sequence of stereo-pairs. This concept specifies just a high-level syntax, so that the decoder interprets the video streams as video and depth. This approach has been adopted by MPEG as a container format "Representation of Auxiliary Video and Supplemental Information" and is known as MPEG-3 Part 3, for video plus depth data [69, 70, 134]. The H.264/AVC [74] is also able to transmit the depth data using its auxiliary picture syntax. This approach is backward compatible with any H.264/AVC video decoder.

4.2 MMRG H.264 multiview extension

An MMRG multi-view H.264 extension has been proposed and implemented by *Multimedia Research Group* (MMRG) of Middle East Technical University, Turkey [16]. MMRG coder is based on H.264. This coder exploits correlation between the views in order to achieve higher compression ratio than in

simulcast coding.

MMRG coder utilizes H.264 motion estimation and compensation. Several modifications have been done to original H.264 reference software [65]. The input/output multiplexing/demultiplexing blocks are added in order to multiplex frames from different cameras into one stream. A *decoded picture buffer* (DPB) and the referencing conditions of H.264 are also modified to provide more efficient motion and disparity estimation. Depending on the encoding parameters, a list of reference frames is formed from the frames in DPB in such a way, that the temporally and spatially closest frames are placed first in the list [16]. MMRG H.264 multiview extension can provide different reference modes for multi-view video. In the stereoscopic case, the encoder has an H.264-compatible mode thus making it possible to decode the frames from the left camera with any standard H.264/AVC decoder.

The MMRG coder has been used to implement the MD schemes for stereo video coding which will be introduced in Sections 4.3 and 4.4.

4.3 Spatial scaling stereo-MDC

MDC has a number of similarities to stereoscopic video coding. In MDC, several bitstreams (descriptions) are generated from the source information. The resulting descriptions are correlated and have similar importance. Descriptions are independently decodable at the basic quality level. The more descriptions are received, the better is reconstruction quality. MDC is especially beneficial when combined with multi-path transport [8], i.e. when the descriptions are sent to the decoder over the separate paths.

It is believed that coding distortions can be less perceivable in stereo image pairs than in single images [28, 126]. If the distortion is only due to low-pass filtering one of the sequences then the perceived image quality is determined by the higher quality image [136, 137]. In case of distortion caused by MPEG-2 [73] quantization, the perceived image quality is the average of the qualities of the left and right sequences [137].

In general, the overall distortion measure (D) of a stereo-pair can be computed as a linear combination of the individual distortions of the left and right images D_L and D_R, respectively [161]:

$$D = D_L + \alpha D_R. \tag{4.1}$$

There are two theories about the effects of unequal bit allocation between the left and the right images in a stereo-pair. Those theories are *suppression theory* and *fusion theory* [40, 161]. In fusion theory, it is believed that the total bit budget should be equally distributed between two views. According to suppression theory, the overall perception in a stereo-pair is determined by the highest quality image in the case when one of the images is low-pass filtered. Therefore, one can compress the target image as much as possible to save bits for the reference image, so that overall distortion is the lowest.

Our SS-MDC approach is based on these two theories. In [1], the perception performance of spatial and temporal down-scaling for stereoscopic video compression has been studied. The obtained results has indicated that spatial and spatiotemporal scaling provide acceptable perception performance while reducing the bitrate. It gave us the idea of using scaled stereoscopic video as the side reconstruction in our MD coder.

4.3.1 Prediction scheme

Figure 4.3 presents the *spatial scaling stereo-MDC* (SS-MDC) scheme. This scheme exploits spatial scaling of one view [98]. In *Description 1*, left frames are encoded separately, and right frames are predicted using both motion and disparity compensation. Left frames are encoded with the original resolution; right frames are downsampled prior to encoding. *Description 2* is encoded symmetrically to *Description 1*. In *Description 2*, right frames are encoded with the original resolution and left frames are downsampled before compression.

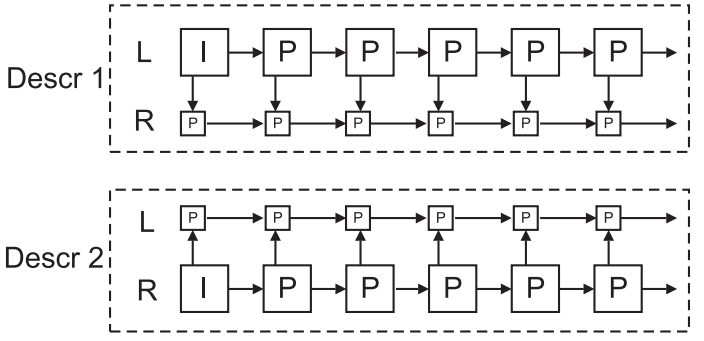


Figure 4.3: Spatial scaling stereo-MDC (SS-MDC).

When both descriptions are received, left and right frames are recon-

structed in full resolution. If one description is lost due to channel failures, the decoder reconstructs the stereoscopic video pair where one view is low-pass filtered. A stereo-pair where one view has the original resolution and another view is low-pass filtered provides acceptable stereoscopic perception. After the channel starts working again, the decoding process returns to the central reconstruction (with both views having full resolution) when the IDR picture is received.

The proposed scheme can be easily done standard compatible. If each description is coded with standard compatible mode of MMRG coder [16] then the standard H.264 decoder can decode the original resolution sequence from each description. The proposed scheme produces balanced descriptions because the left and right frames usually have similar characteristics and are encoded with the same bitrate and visual quality. The proposed SS-MDC scheme is drift-free, i.e. it does not introduce any mismatch between the states of the encoder and decoder in case of description loss.

4.3.2 Downsampling

Downsampling is done by low-passed filtering the frame followed by decimation. The following filters are used:

13-tap downsampling filter:
$\{0, 2, 0, -4, -3, 5, 19, 26, 19, 5, -3, -4, 0, 2, 0\}/64$
11-tap upsampling filter:
$\{1, 0, -5, 0, 20, 32, 20, 0, -5, 0, 1\}/64$

Filters are applied to luma and both chroma components in both horizontal and vertical directions, and picture boundaries are padded by repeating the edge samples. These filters are used in *scalable video coding* (SVC) extension of H.264 [117] and explained in [129]. Downsampling is done for factors of 2 in both dimensions. In the downsampled sequence, the disparity estimation exploits downsampled reference frames to ensure proper estimation.

4.3.3 Redundancy of SS-MDC

The bitrate generated by the SS-MDC coder is $R = R^* + \rho_{sim} + \rho_d$, where R^* is the bitrate obtained with the single description coding scheme providing

the best compression, ρ_{sim} denotes the redundancy caused by using simulcast coding instead of joint coding, and ρ_d is the bitrate spent on coding the downscaled sequences. Thus, the redundancy $\rho = \rho_{sim} + \rho_d$ of the proposed method is low-bounded by the redundancy of the simulcast coding ρ_{sim}. The redundancy of the simulcast coding ρ_{sim} depends on characteristics of the video sequence and varies from one sequence to another. The redundancy ρ_d of coding two downsampled sequences can be adjusted to control the total redundancy ρ by changing the scaling factor and the quantization parameter QP of the downscaled sequence.

4.4 Multi-state stereo-MDC

The *multi-state stereo-MDC* (MS-MDC) scheme is shown in Figure 4.4 [98]. Stereoscopic video sequence is split into two descriptions. Odd frames of both views belong to *Description 1*, and even frames of both views belong to *Description 2*. Motion compensated prediction is done separately in each description. In *Description 1*, left frames are predicted from the preceding left frames of *Description 1*, and right frames are predicted from the preceding right frames of *Description 1* or from the left frames corresponding to the same time moment. The idea of this scheme is similar to video redundancy coding (VRC) [160] and multi-state coding [7], which are described in Subsection 3.1.3.

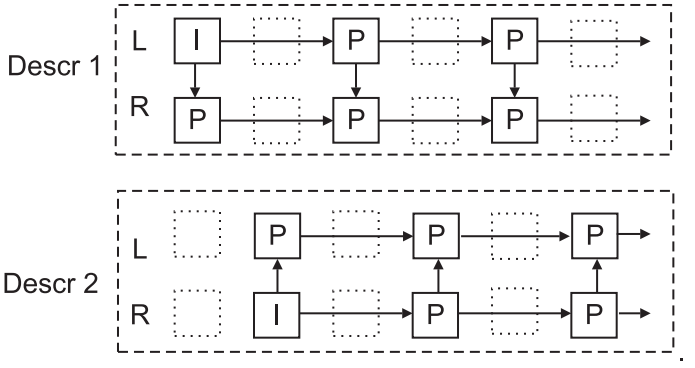


Figure 4.4: Multi-state stereo MDC.

If the decoder receives both descriptions, the original sequence is reconstructed with the same frame rate. If one description is lost, stereoscopic video

is reconstructed at half of the original frame rate. Another possibility is to employ a frame concealment technique for the lost frames. As one can see from Figure 4.4, a missed frame (e.g. the odd frame) can be concealed by employing motion vectors of the next (even) frame, which uses the previous even frame as a reference in motion-compensated prediction.

In this MDC scheme, the redundancy cannot be adjusted and is determined by characteristics of the video sequence. However, for some video sequences, this prediction scheme is able to reach bitrates lower than the bitrate of simulcast coding $R_{sim} = R^* + \rho_s$. Another two advantages of MS-MDC are the following. This scheme can easily be generalized for more than two descriptions. MS-MDC also does not introduce any mismatch between the states of the encoder and decoder in case of description loss.

4.5 Simulation results

In the experiments, we compare side reconstruction performance of the proposed MDC schemes. The results are provided for four stereoscopic video pairs: *Train and tunnel* (720 × 576, 25 fps, moderate motion, separate cameras), *Fun-fair* (360 × 288, 25 fps, high motion, separate cameras), *Garden* (960 × 540, 25 fps, low motion, close cameras) and *Xmas* (640 × 480, 15 fps, low motion, close cameras). Both algorithms are applied to these videos. In all the experiments, I-frames are inserted every 25 frames.

The reconstruction quality measure is PSNR. PSNR value of the stereopair is calculated according to the following formula, where D_L and D_R represent the distortions in the left and right frames [19, 161]:

$$PSNR_{pair} = 10 \log_{10} \frac{255^2}{(D_L + D_R)/2}.$$

In the experiments, average $PSNR_{pair}$ is calculated over the sequence. Redundancy is calculated as the percentage of additional bitrate over the encoding with the minimal bitrate R^*, i.e. the bitrate of the *joint* coding scheme.

To examine characteristics of the video sequences, we code them by *joint* coder and *simulcast* coder for the same PSNR. The results are shown in Table 4.1. One can see that *Train and tunnel* and *Fun-fair* sequences show low inter-view correlation, and sequences *Garden* and *Xmas* show high inter-view correlation. Thus, *Garden* and *Xmas* have high redundancy of simulcast cod-

ing ρ_{sim}, which is the lower bound for the redundancy of SS-MDC coding scheme.

4.5.1 Comparison of the proposed schemes

The SS-MDC scheme is tested for downsampling factors 2 and 4 in both vertical and horizontal directions. For each downscaling factor, we change quantization parameter (QP) of the downscaled sequence to achieve different levels of redundancy.

The results for the second scheme (MS-MDC) are given only for one level of redundancy. The reason is that this method does not allow to adjust redundancy since the coding structure is fixed as in Figure 4.4. The redundancy of MS-MDC method takes only one value and is determined by characteristics of the video sequence.

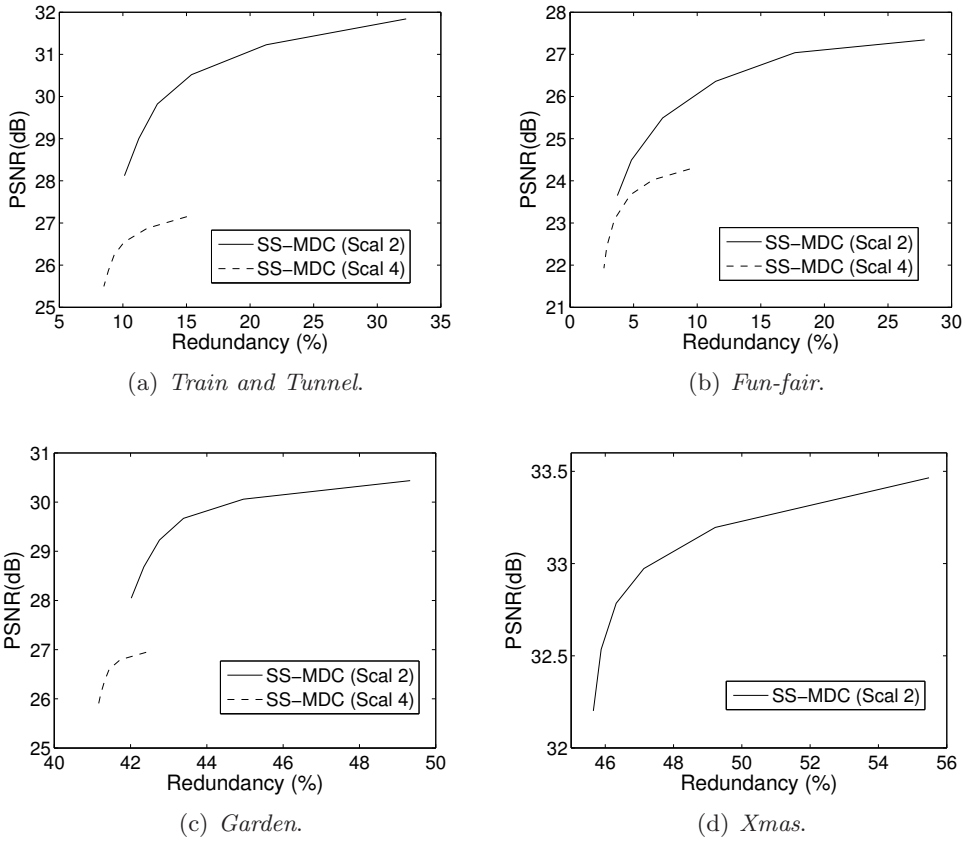
Figure 4.5 shows the redundancy-rate distortion (RRD) curves [106] for SS-MDC. The results are presented as PSNR of the side reconstruction (D_1) vs redundancy ρ. The results for SS-MDC are given for scaling factors 2 and 4. The central reconstruction distortions D_0 can be found in Table 4.1. For *Xmas* sequence, simulation results for scaling factor 4 are not shown, as PSNR is much lower than for scaling factor 2.

Figure 4.6 visually compares the performance of SS-MDC and MS-MDC for the same experimental data. One can see from the figure, that SS-MDC performs better than MS-MDC for the sequences *Train and tunnel* and *Fun-fair*. On the contrary, MS-MDC performs better than SS-MDC for the sequences *Garden* and *Xmas*.

The simulation results show that reconstruction from one description can provide acceptable video quality. The SS-MDC method can work in a wide range of redundancies. Downscaling with factor 2 provides good visual quality with acceptable redundancy. However, the performance of SS-MDC de-

Sequence	D_0, dB	$R^* = R_{joint}$, Kbps	R_{sim}, Kbps	ρ_{sim}, %
Train and tunnel	35.9	3624	3904	7.7
Fun-fair	34.6	3597	3674	2.2
Garden	35.6	5444	7660	40.7
Xmas	38.7	1534	2202	43.5

Table 4.1: Joint and simulcast coding.



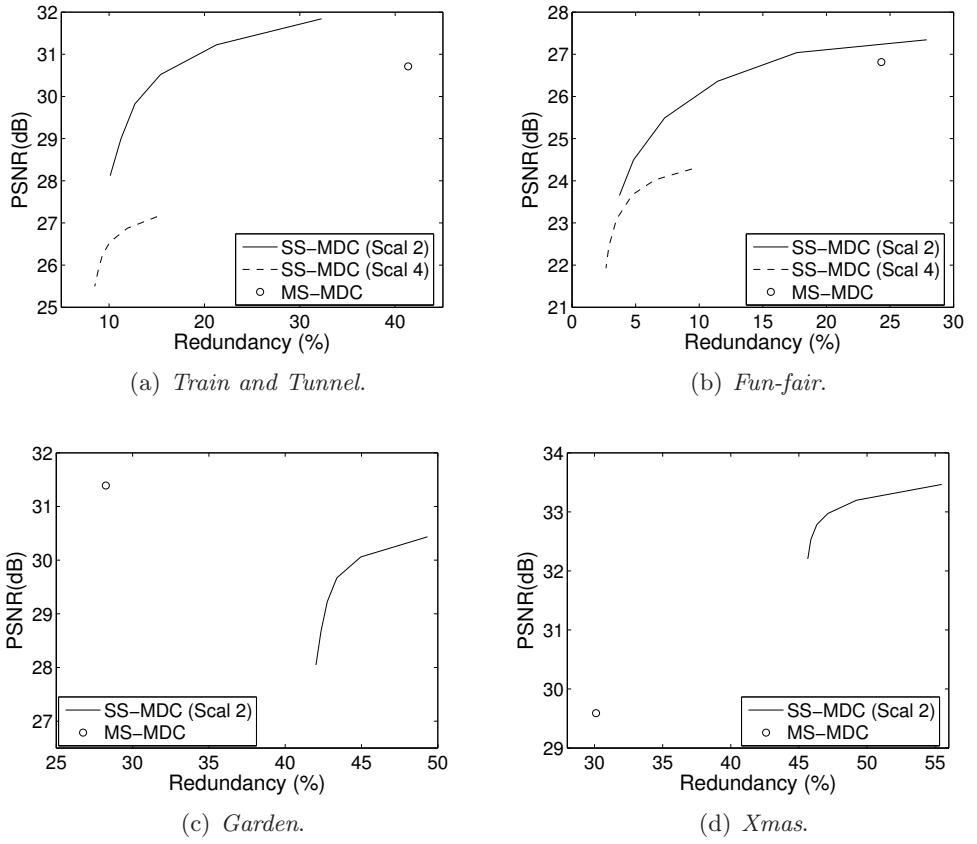
(a) *Train and Tunnel.* (b) *Fun-fair.*

(c) *Garden.* (d) *Xmas.*

Figure 4.5: Side-reconstruction distortion D_1 vs redundancy ρ for SS-MDC.

pends to a great extent on characteristics of the sequence. This method can achieve very low redundancy (less than 10%) for sequences with lower inter-view correlation (*Train and tunnel*, *Fun-fair*). However, it has relatively high redundancy when applied to stereoscopic video sequences with higher inter-view correlation (*Xmas*, *Garden*). The perception performance of SS-MDC is quite good as the stereo-pair perception is mostly determined by quality of the high-resolution picture.

The MS-MDC coder performs usually with 30-50% redundancy and can provide acceptable side reconstruction even without error concealment algorithm (just by copying the previous frame instead of the lost frame). MS-MDC should be used for sequences with higher inter-view correlation, where SS-MDC exhibits high redundancy.



(a) *Train and Tunnel.* (b) *Fun-fair.*

(c) *Garden.* (d) *Xmas.*

Figure 4.6: Comparison of SS-MDC and MS-MDC for various test sequences. Side-reconstruction distortion D_1 vs redundancy ρ.

4.5.2 Criterion for switching the schemes

The encoder can decide which scheme to use by collecting the encoding statistics. Table 4.2 shows the statistics of motion vectors (MVs) prediction for the *joint* coding mode, SS-MDC, and MS-MDC. The statistics are collected for P-frames of the right sequence. Values in Table 4.2 show the average value c_i, which is the fraction of motion vectors in the total number of motion and disparity vectors in the right sequence frame. That is,

$$c_i = \frac{m_i}{m_i + d_i}, \tag{4.2}$$

where m_i is the number of motion vectors for frame i, and d_i is the number of disparity vectors in frame i. One can see that the average value of c_i correlates

Sequence	Joint	SS-MDC	MS-MDC
Train and tunnel	0.94	0.78	0.90
Fun-fair	0.92	0.80	0.85
Garden	0.65	0.60	0.63
Xmas	0.66	0.56	0.61

Table 4.2: Ratio of MVs to sum of MVs and DVs in right sequence, c_i, Eq. (4.2).

with the redundancy of *simulcast* coding ρ_{sim} given in Table 4.1. The value c_i tells the decoder when to switch from SS-MDC to MS-MDC and vice versa.

Thus, the encoder operates as follows. Once the encoding mode has been chosen depending on c_i, the encoding process starts, and the statistics are being collected. Before encoding the IDR picture, the encoder compares the values $c_{i-N}, \ldots, c_i$ of the last $N+1$ right frames with the threshold $T = 0.7$ and decides whether to switch to a different mode or not. Thus, the encoder adaptively chooses between the SS-MDC or MS-MDC mode depending on characteristics of the video sequence.

4.6 Chapter conclusions

Two MDC approaches for stereoscopic video have been introduced. The approach called SS-MDC exploits spatial downsampling of one view while the MS-MDC approach is based on temporal subsampling. Both approaches produce balanced descriptions and provide stereoscopic reconstruction with acceptable quality in case of one channel failure for the price of moderate redundancy (in the range of 10-50%). Both approaches provide drift-free reconstruction in case of description loss. The performance of these approaches depends on characteristics of stereoscopic video sequence. The SS-MDC approach performs better for the sequences with lower inter-view correlation while MS-MDC performs better for the sequences with higher inter-view correlation. The criterion for switching between the approaches is used at the encoder to choose the approach that provides better performance for this sequence.

Chapter 5

Multiple Description Coding of 3D Geometry

Nowadays, the most common representation for 3D objects is a surface-based triangle mesh [6]. Achieving high quality of 3D models requires large number of triangles. Therefore, compressing mesh data is necessary due to storage space and bandwidth limitations. A great number of 3D mesh compression schemes have been proposed [110]. In multimedia applications such as virtual presence, internet computer games, e-commerce, tele-medicine, object-based video compression, 3DTV, 3D meshes need to be communicated in a networked environment. However, in a typical network, packets may be lost or delayed because of congestions and buffers overflow. For scenarios, in which the mesh data is transmitted over the error-prone channel, error resilience can be achieved by MDC of 3D geometry.

This chapter is organized as follows. Section 5.1 gives a short survey of 3D mesh compression approaches. Sub-section 5.1.3 surveys the approaches to MDC of 3D meshes. In Section 5.2, we propose a novel method for MDC of 3D geometry. Section 5.3 describes the optimization of the proposed method. Section 5.4 introduces MD-FEC of 3D geometry. Section 5.5 discusses the computational complexity issues of the proposed methods and shows how the complexity can be reduced by introducing D-R curve modeling. Finally, Section 5.6 discusses experimental results and Section 5.7 concludes the chapter.

5.1 Survey of 3D mesh coding

Among numerous types of geometry representation, triangular mesh is widely used in graphic rendering hardware and for many simulation algorithms. For this reason, a lot of research has been done on triangular 3D mesh compression. A typical 3D mesh is determined by its geometry and connectivity information. The geometry information includes the coordinates of the mesh vertices in the 3D space, while connectivity information includes the incidences between the mesh vertices and is determined by the mesh topology. A typical mesh compression algorithm compresses the connectivity and geometry data separately. While the geometry information can be compressed in the lossy mode, connectivity data describing the adjacency information between vertices is usually coded in the lossless mode. In connectivity-driven compression, mesh connectivity is compressed without regarding the geometry, and coding of the geometry data is determined by underlying connectivity.

Earlier works on mesh compression considered single-rate coding of 3D meshes. These works were originally intended to save bandwidth when transmitting the geometry information from the CPU to the graphic card [110]. In single-rate coding, a 3D mesh can be rendered when the whole bitstream has been received. Later works considered progressive mesh encoding, which has been driven by popularity of the Internet. In progressive compression, a mesh can be continuously reconstructed from the coarse to fine quality with different level of details.

5.1.1 Single-rate compression

Connectivity compression

Taubin and Rossignac have proposed a *topological surgery* algorithm [140]. In this algorithm, the mesh is cut along the set of cut edges in a way that the resulting "flattened" mesh has the form of a planar polygon. In this way, the mesh connectivity is presented in the form of two spanning trees: the *vertex spanning tree* which represents the set of cut-edges, and the *triangle spanning tree*, representing the planar polygon. Then, both spanning trees are run-length encoded. In order to achieve better compression, mesh cutting into the spanning tree is done in such a way that the resulting tree contains longer runs. Vertex coordinates are encoded with a simple first order predictor. The

prediction is done in the order provided by connectivity compression. Prediction residuals are entropy-encoded. This method is applicable to triangular manifold meshes. This algorithm is also a part of MPEG-4 standard [71].

The approach to triangle mesh coding proposed by Gumhold and Strasser is called a *cut-border machine* (CBM) [63]. It is a *triangle conquest* approach, in which the algorithm inserts a new triangle in the "conquered" area closed by the cut-border. The insertion of a new triangle is done by using one of the following building operations: *new vertex*, *connect backward*, *connect forward*, *border*, *split cut-border*, *close cut-border*, and *cut-border union*. Another example of the triangle conquest approach is the *Edgebreaker* algorithm introduced by Rossignac [120]. The main difference between the CBM and Edgebreaker is how they code the data associated with split operation. The operations used by Edgebreaker are *loop extension* (C), *left* (L), *right* (R), *end* (E), and *split* (S). The Edgebreaker algorithm gives the worst-case bound on connectivity compression bitrate, which is 4 bits per vertex (bpv). However, the Edgebreaker requires two decoding passes for decompression and has $O(v^2)$ computational complexity, where v is the number of vertices. The CBM cannot meet this worst-case bound but allows single-pass coding and decoding. This makes it suitable for coding the large data sets. The Edgebreaker performance and complexity have also been improved in a number of publications.

A valence driven approach has been proposed by Touma and Gotsman [143] and is known as a TG coder. The TG algorithm is considered state-of-the-art in single-rate 3D mesh compression. The algorithm is applicable to orientable manifold triangle meshes. It is based on the fact that in the orientable manifold, all the vertices incident to the chosen vertex may be ordered. Thus, mesh connectivity can be encoded as a list of vertex valences in a special order. The algorithm starts with the triangle, pushing its vertices to the *active list*. Then, the algorithm pops up the vertex and tries to expand the active list by conquering edges in the counter-clockwise direction. When all the edges are exhausted, the algorithm proceeds to the next vertex in the active list. The algorithm outputs the valences of the visited vertices. Since the distribution of valences in a typical triangle mesh is around 6, the output can be efficiently compressed by arithmetic coding. The resulting bitrates are on average less than 1.5 bpv for connectivity information [110]. In [5], Alliez and Desbrun have further improved the performance of TG algorithm. Their coder performs better than TG coder, especially for irregular meshes.

Geometry compression

Unlike the connectivity, which is usually encoded losslessly, the geometry is coded using lossy compression methods. Although the geometry requires more data in the compressed bitstream than the connectivity, most efforts in the past have been applied to connectivity coding. Therefore, in most approaches geometry coding follows connectivity coding. To exploit correlation between the vertices, most geometry compression schemes use a three-step approach: pre-quantization of vertex positions, prediction of quantized positions, entropy-coding of prediction residuals [110]. The quantization techniques include scalar and vector quantization. Quantization can be globally uniform or different quantization resolution can be applied to different parts of the mesh. Typically, geometry coding quantizes each coordinate uniformly at the precision of 8 to 16 bits per coordinate.

Prediction of vertex coordinates exploits the correlation between the adjacent vector coordinates. Usually, the prediction schemes are the special cases of the linear prediction scheme [110]. Among the proposed prediction methods are delta prediction [37], linear prediction [140], and parallelogram prediction [143].

5.1.2 Progressive mesh compression

When transmitting complex meshes over the network, a progressive mesh compression is beneficial. First, the coarse mesh is transmitted and rendered. When the refinement data are transmitted, mesh can be rendered at a higher resolution. In this case, the user does not need to wait until all the mesh data are downloaded and may even cancel the transmission if the *level of detail* (LOD) is high enough. To achieve progressive compression, the original mesh is simplified step-by-step until the coarse mesh is obtained. If the inverse operation to mesh simplification exists, it is possible to obtain the original mesh from the coarse mesh and several levels of details. However, having several levels of details usually decreases the compression efficiency.

The progressive mesh compression approaches can be split into two groups: connectivity-driven techniques, which usually provide lossless mesh compression, and geometry-driven techniques which are usually lossy. In the latter case, mesh connectivity can even be changed to provide better compression of the original geometry.

Lossless compression

The concept of *progressive mesh* (PM) coding has been first introduced by Hoppe [67]. In this method, the *coarse mesh* is obtained from the original mesh by a sequence of *edge collapses*. The edge is collapsed by merging its endpoints into one point. The two triangles which share this edge are removed and all vertices connected to the previous vertices are reconnected to the merged vertex. The vertex split is the inverse operation to the edge collapse. Thus, the original mesh can be represented as a coarse mesh and a sequence of vertex split operations. The order of mesh collapses is important. Hoppe uses the energy function which is determined by the distortion caused by edge collapse operation. The energy values are assigned to each edge, with edges carrying smaller energy values corresponding to lower distortion. The edges with smaller energy values are collapsed first. The PM method is not very efficient from the compression viewpoint.

The PM approach [67] has two restriction. First, it is applicable only to orientable manifold meshes. Second, edge collapse operation always preserve the original mesh topology. Therefore, Popovic and Hoppe [111] have proposed a method based on more general mesh simplification operation called a *progressive simplicial complex* (PSC). The basic operation merges two vertices, which do not have to share an edge. Four different vertex configurations are possible. The inverse operation is the generalized vertex split. PSC is capable of dealing with arbitrary mesh topology. However, this approach may demand more bits than PM for connectivity coding. Both PM and PSC use delta prediction for geometry coding.

The *progressive forest split* (PFS) technique [139] enhances the PM's technique. In this approach, Taubin et al. have achieved the improved compression efficiency at the expense of worse granularity. In this approach, the difference between two successive levels of detail consists of a group of vertex splits that is realized by PFS operation. The PFS operation assigns the mesh edges to the forest of trees. Then, the trees are cut through the edges and the resulting crevices are filled by the triangulation process. A single PFS operation can double the number of triangle in the mesh. Each PFS operation encodes the forest structure, triangulation information and subsequent vertex translations. The forest trees can be encoded as it is done in topological surgery approach [140]. Minimizing the number of LOD's, one can achieve higher

compression ratios. PFS has been adopted by MPEG-4 [71].

Pajarola and Rossignac [108] proposed a *compressed progressive mesh* (CPM) approach, which also improves the performance of PM scheme [67] by the cost of decreased granularity. The edge collapses are grouped into a series of independent sets. The location of each vertex in the set is marked by two-coloring the mesh, which results in the overhead of 1 bpv per set. The prediction of the new vertex positions is done in a way similar to the butterfly subdivision scheme [42]. The CPM approach requires about half of the bitrate of PFS.

A valence-driven approach has been proposed by Alliez and Desburn [4]. The key observation used in this method is the following. The entropy of mesh connectivity depends on the distribution of mesh valences. Removing the vertex with valence more than six increases the connectivity entropy. In contrast, removing the vertex with valence less than three, decreases the entropy. Therefore, a valence-driven decimation is applied, which is done in two passes. In the decimation pass, the mesh is divided into patches around the vertices and vertices with valence less or equal six are removed. Vertex removal is followed by the retriangulation procedure, which fills the patches. During the cleaning pass, only the vertices with valence three created in the retriangulation process, are removed . The algorithm outputs a sequence of vertex valences that is compressed by the arithmetic coder. Geometry is encoded with barycentric prediction combined with a local coordinate frame. The approach has average compression ratio about 1.1 time larger than TG coder [143] even though it is fully progressive. This makes it one of the state-of-the-art lossless compression schemes.

Progressive Geometry Compression (PGC)

The approaches described above encode the connectivity and geometry information separately; connectivity is usually compressed losslessly. However, a mesh is just an approximation of the original geometry. The mesh represents the original geometry with some fidelity depending on sampling and discretization errors E_s and E_d. One can consider a 3D mesh as consisting of the *geometry*, *connectivity*, and *parameter* information [83]. Consider a vertex in a highly sampled mesh. Moving the vertex in the direction normal to the surface will result in changing the geometry. However, slightly moving this vertex within the surface will not change the geometry; only the parameter

information will change. The most important idea of the Khodakovsky's [83] work is to treat the geometry and parameter information separately. High compression ratios can be achieved by eliminating the parameter and connectivity information while compressing the geometry information.

The *progressive geometry compression* (PGC) scheme [83] accepts arbitrary 2-manifold triangular meshes. The scheme consists of three main blocks: remeshing, a wavelet transform, and a zero-tree coding algorithm. In the first step, a smooth global parametrization is computed with the MAPS algorithm [90]. This makes it possible to compute successive approximation of the original mesh with semi-regular connectivity. The sequence of approximations at different levels of detail builds a multiresolution hierarchy. The approximations are obtained from the irregular coarse mesh by the face subdivision process (see Figure 5.1). The connectivity information in a semi-regular mesh depends only on the base mesh and the number of subdivisions. Hence, connectivity information can be compressed efficiently. Although the parametrization error E_r can be made arbitrarily small, it does not make sense to make it smaller than the discretization error E_d. Otherwise, the semi-regular mesh will be just a better approximation of the input mesh and not necessarily of the original geometry. The equality $E_r = E_d$ usually holds when triangles of the semi-regular mesh are approximately of the same size as the triangles of the input mesh [83].

The subsequent wavelet transform replaces the original mesh with the coarsest mesh and a sequence of wavelet coefficients, which represent the difference between the successive levels. Khodakovsky et al. use Loop wavelet transform in their scheme [91]. The coarsest level connectivity is irregular; it is coded by the TG coder [143]. The distribution of wavelet coefficients is centered about zero and their magnitudes decay at finer levels. This justifies using the zero-tree coder for compressing the wavelet coefficients.

Wavelet coefficient trees are coded with SPIHT algorithm [123]. The main difference from the SPIHT for images is how the coefficient trees are formed. In images, coefficients are associated with quadrilateral faces. Thus, it is easy to form quad trees. In meshes, vertices do not have the tree structure, but the edges do. Each edge is a parent of four edges of the same orientation in the finer mesh (see Figure 5.1). Every edge of the base domain forms a root of the wavelet coefficient tree. This grouping is consistent for an arbitrary semi-regular mesh [83]. Thus, usual SPIHT can be applied to the coefficients.

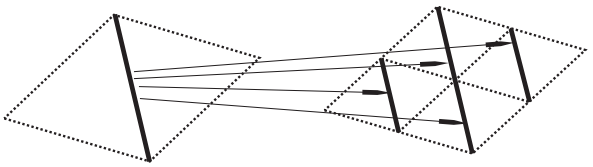


Figure 5.1: Coarse edge (left) is parent to four finer edges (right) [83].

The wavelet coefficients are vector-valued. The x, y and z components exhibit high degree of correlation. To reduce the correlation between the components, a local frame induced by the surface tangent plane is used [167]. In a smooth semi-regular mesh, the normal component is much larger than the tangential component. The former is also more important for the geometry. All vector components are quantized and coded with SPIHT separately, with quantization cells for the tangential component four times larger than for the normal component [83]. The output of the SPIHT coder can be compressed even further with arithmetic coding, although SPIHT itself provides good compression.

For better progressivity, a predetermined number of bit-planes of the coarsest level geometry can be transmitted initially with the coarsest level connectivity. The refinement bit-planes can be transmitted as the SPIHT coder descends a given number of wavelet coefficient bit-planes [83]. The approach reportedly provides a 12 dB better quality than CPM for the same bitrate.

Another wavelet coder has been proposed by Khodakovsky and Guskov [82]. This coder is based on *normal meshes* [64]. The normal meshes represent wavelet details as a scalar offset in the normal direction relative to the coordinate frame. Consequently, wavelet coefficient is represented with a scalar value instead of three-dimensional vectors like in [83]. Therefore, improvement of 2-5 dB can be achieved compared to [83]. The unlifted butterfly scheme [42] has been used as a wavelet predictor as it is also used in normal remeshing [64].

5.1.3 MDC of 3D meshes

Several MDC methods have been proposed for 3D mesh coding. The first approach, called *partitioning vertex geometry* is based on the idea that the geometry information may be coded in a lossy manner while the connectiv-

ity information is of a vital importance and should be coded in a lossless manner [77]. In this approach, 3D mesh vertices are split between different descriptions while the connectivity information is included in all descriptions. Loosing a description causes loss of geometry coordinates of some vertices. In this case, the locations of missing vertices can be estimated from the available vertices. The connectivity information is always available and undistorted thus helping to estimate missing vertices. Duplicating the connectivity information in the descriptions is not costly since the size of compressed connectivity is much smaller than the size of the compressed geometry [77]. The connectivity is compressed with the topological surgery approach [140]. The geometry compression uses prediction of the vertex coordinates followed by arithmetic coding of prediction residuals. In the description, the vertices are predicted from the preceding vertices of the same description in order to ensure successful reconstruction when one description is lost. In order to minimize the loss in compression efficiency due to this prediction strategy, a surface-based prediction scheme is used [77].

Multiple description scalar quantization (MDSQ) [145] has been described in Sub-section 2.1.1. The 3D-geometry coding approach using MDSQ has been proposed in [14]. This method is based on Khodakovsky's PGC scheme [83], which has been described in Sub-section 5.1.2. After the mesh geometry is transformed into the wavelet domain, the MDSQ is applied to each wavelet coefficient producing two sets of MDSQ indices. Then, each set of indices is compressed with SPIHT. Each description contains a set of MDSQ indices compressed with SPIHT and the TG-coded coarsest-level mesh. Since the size of the coarsest-level mesh is small, the redundancy introduced by duplicating the coarsest-level mesh does not significantly affect the overall compression performance. The redundancy in this method can be adjusted by changing the number of diagonals in the MDSQ index assignment matrix.

5.2 Tree-based mesh MDC (TM-MDC)

In this section we present a *tree-based mesh MDC* (TM-MDC) scheme [99]. The proposed scheme is based on PGC [83], which is a progressive compression technique for arbitrary topology, highly detailed and densely sampled meshes arising from geometry scanning. PGC is described in Sub-section 5.1.2 and consists of remeshing, wavelet transform, and SPIHT coding of wavelet coefficients.

To obtain multiple descriptions from wavelet coefficients, we adapted the ideas from MD image coding [78, 94]. Suppose that the coder generates N descriptions. In our coder (Figure 5.2), wavelet coefficient trees are split into several sets W_i, $i = 1 \dots N$ and coded by SPIHT algorithm at high bitrate. These sets are included in the descriptions in the following way. Each description contains M copies of different tree sets ($M \leq N$). Namely, *Description* i contains a set W_i coded at rate $R_{i,0}$ and $M-1$ sets of redundant trees W_j, $j \neq i$. These $M-1$ tree sets represent coding redundancy and are coded at the rates lower than $R_{i,0}$. That is $R_{i,0} > R_{i,1} > \dots R_{i,M-1}$. Rates $R_{i,j}$ are obtained as a result of optimization explained in Section 5.3.

If all the descriptions are received, the decoder uses for reconstruction only the tree copies coded with rates $R_{i,0}$. The redundant copies are used for the trees which higher-rate copies are lost. Thus, if some descriptions are lost, the lost trees are recovered from their lower-rate copies from the received descriptions. The compressed coarsest mesh C coded with the rate R_C is included in every description. Thus, the decoder can perform the inverse wavelet transform even if only one description is received. Duplicating coarsest mesh C in all descriptions also contributes to coding redundancy.

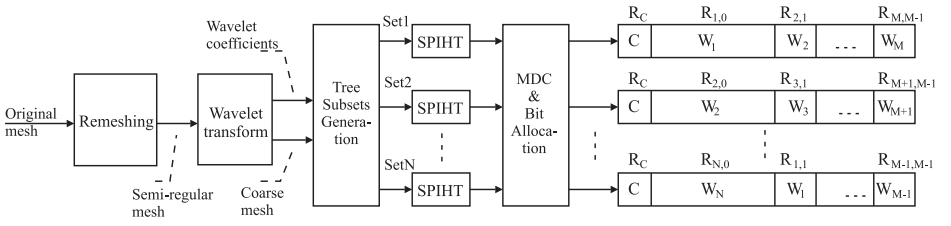
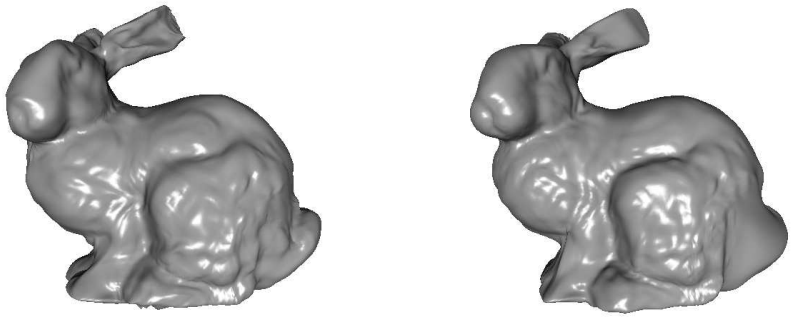


Figure 5.2: TM-MDC encoder scheme.

The manner of grouping the coefficient trees into sets is particularly important, since different sets are reconstructed with different quality in the case



(a) PSNR= 50.73 dB (b) PSNR= 48.51 dB

Figure 5.3: Reconstruction from one description for different types of tree grouping. (a) Grouping spatially disperse trees; PSNR = 50.73 dB. (b) Grouping spatially close trees; PSNR = 48.51 dB. Group size is 10.

of description loss. Therefore, 3D mesh locations corresponding to different tree sets will have different quality. To perform grouping of the trees into sets, ordering of the coarsest mesh vertices is performed, as proposed in [17, 80]. It provides ordering of the vertices that has good locality and continuity properties. Then, the desired type of grouping the wavelet trees is obtained by sampling the one-dimensional array, the output of the algorithm from [17, 80].

Two types of grouping the trees have been compared: 1) grouping the closely located trees together; 2) grouping spatially disperse trees. The spatially close grouping is obtained by assigning successive vertices from the array to the same group. The disperse grouping is obtained by sampling the array in a round-robin fashion. It has been observed that the latter case yields annoying artifacts when only one description is received and that the former case results in better visual perception of the reconstructed mesh. However, the latter case gives better objective quality (PSNR).

This is illustrated in Figure 5.3. Model Bunny is encoded into four descriptions and optimized for the description loss rate $P = 15\%$. Two grouping strategies are compared: grouping spatially disperse trees and grouping close trees with group size equal to 10. The model is reconstructed from the *Description 1* alone. One can see that although grouping disperse trees achieves lower objective distortion than grouping close trees, it produces annoying visual artifacts. Therefore, the remaining results of this work have been obtained by grouping spatially close trees with group size 10.

5.3 Optimizing bit allocation for TM-MDC

The redundancy of the proposed algorithm is determined by the number of redundant tree copies, their rates, and the coarsest mesh size. Bit allocation has to minimize the expected distortion at the decoder subject to packet (description) loss conditions on the network and the target bit budget.

Suppose that N descriptions are generated. Then, coefficient trees are split into N sets, and M copies of tree sets are included in one description ($M \leq N$). Assume that the P_j, $j = 0, \ldots M$ are the probabilities that j-th copy of the tree is used for reconstruction. In this case, P_0 is the probability of getting the full-rate copy of the tree set, and P_M is the probability of not receiving any copy of the tree set. We have to minimize the expected distortion

$$E[D] = \sum_{i=1}^{N} \sum_{j=0}^{M} P_j D_{ij}(R_{ij}) \tag{5.1}$$

where D_{ij} is the distortion incurred by using j-th copy of a tree set i and R_{ij} represents bits spent for j-th copy of i-th tree set. Optimization is performed under the following bitrate constraints

$$\sum_{i=1}^{N} \sum_{j=0}^{M} R_{ij} + NR_C \leq R \tag{5.2}$$

where R is the target bitrate and NR_C is the rate of the coarsest mesh. The rate of the coarsest mesh is constant with the geometry information quantized to 14 bitplanes.

Probabilities P_j can be found from the packet loss model and the packetization strategy. In our experiments, we use a simple model of the memoryless channel where probability of the description loss P is the same for each description and is independent of the previous channel events. In the following, we assume for simplicity that one packet corresponds to one description. If the description has to be fragmented into different packets, probability of the description loss P can easily be found from the *packet loss rate* (PLR). For this model, the probability P_0 of receiving the main copy of the description is obviously

$$P_0 = 1 - P. \tag{5.3}$$

Then, the probability that the j-th copy is used for reconstruction is

$$P_j = (1 - P)P^j \tag{5.4}$$

and the probability that all the copies of the tree set are lost is

$$P_M = P^M. \tag{5.5}$$

Optimizing bit allocation requires knowledge of $D(R)$ relation for every allocation step. Calculation of $D(R)$ is a computationally expensive operation. Each tree set contributes to total distortion D. Since each set of coefficient trees corresponds to some separate location on the mesh surface (defined by the root edges), the distortions corresponding to separate trees may be considered approximately additive. Therefore, the operational distortion-rate curve $D_i(R_i)$ for the tree set i is obtained in advance. Calculations of $D_i(R_i)$ are performed only once, before the optimization algorithm is used for the first time. Then, D-R curves are saved and can be used every time to perform bit allocation for the new values of R and P.

Optimization is performed with generalized Breiman, Friedman, Olshen, and Stone (BFOS) algorithm [118]. BFOS algorithm first allocates high rate for each copy of the tree set. In each iteration, the algorithm consequently deallocates bits from the set where the $D(R)$ curve shows the lowest decay at the allocated bitrate. This process stops when the bit budget constraints are satisfied. In case the optimization brings zero rates for some redundant copies, these copies are not included in the descriptions. Simulations show that the bit-allocation algorithm exhibits the desired behavior. The higher is the packet loss rate, the more bits are allocated to redundant copies providing more robustness to description losses.

5.4 MD-FEC for 3D mesh coding

The second method that we have used for error-resilient coding of 3D-meshes is based on MD-FEC [15]. The MD-FEC for images is described in [95] and has been already described in Sub-section 2.1.3. A principal idea of this algorithm is to assign unequal amounts of FEC symbols to different parts of the progressive bitstream according to their contribution to reconstruction quality. If the

compression algorithm generates the embedded bitstream (e.g. SPIHT [123]), the importance of the compressed data symbols corresponds to the order of byte in the compressed bitstream.

A PGC-coded bitstream starts with the compressed coarsest mesh connectivity which is the most important part since the whole mesh connectivity depends on the coarsest level connectivity due to the semi-regular mesh structure. The next part of the bitstream contains a predetermined number of the coarsest level geometry bit-planes. The rest of the bitstream consists of the output of the SPIHT algorithm and remaining bitplanes of the coarsest level geometry, which are inserted at the end of each refinement pass of the SPIHT algorithm.

Then, the bit allocation algorithm assigns the optimum set of FEC symbols in order to minimize the expected distortion subject to the packet loss probability model. The channel is simulated by a simple packet (description) loss model. Each description (packet) has the same probability of loss P, which is independent of the previous channel events. To optimize bit allocation, an approximately optimal assignment algorithm [96] has been used. The first stage of this algorithm finds the convex hull of the $D(R)$ curve of the source. At the second stage, a globally optimal assignment of FEC codes is found. This stage assumes that the RS codes are allocated fractionally. The algorithms gets a set of (D, R) pairs, which form a convex hull of operational $D(R)$ curve, and the probability mass function (PMF) $p_X(x)$ of the number of descriptions X received out of N generated descriptions. The algorithm starts with zero rate allocation and moves from one (D, R) point to another in the order of increasing R. At each iteration, the algorithm calculates the ratios $\Delta D_k/\Delta R_k$ which correspond to changes in the distortion because of protecting the new part of bitstream with (N, k) codes. These values are compared to assignment of stronger codes to the earlier parts of the bitstream, and the assignment resulting in the fastest decrease of distortion is chosen. The algorithm terminates when it fills the total bit budget. Then, the last stage removes the fractional RS codes assumption and converts the bit allocation to the approximately optimal one. For the details of the algorithm, the reader is referred to [96].

Table 5.1 shows the redundancies of MD-FEC obtained by bit allocation algorithm for different $D(R)$ curves and packet loss rates.

	Redundancy (%) for different P					
Description loss rate $P =$	1%	3%	5%	10%	15%	20%
L^2 distance	13	23	32	43	47	53
Approx. L^2 distance	12	24	33	43	47	51
Weibull model	16	27	34	44	50	55

Table 5.1: MD-FEC. Model *Bunny*. Redundancy (%) obtained by different bit allocation algorithms for different P.

5.5 Complexity issues and distortion-rate function modeling

One of the most popular and robust objective distortion metric in 3D meshes is L^2 distance between two surfaces. L^2 distance between the surfaces X and Y is defined as

$$d(X,Y) = \left(\frac{1}{area(X)} \int_{x \in X} d(x,Y)^2 dx \right)^{1/2}. \tag{5.6}$$

Since the distance is not symmetric, the metric is symmetrized by taking the maximum of $d(X,Y)$ and $d(Y,X)$. Metro tool [33] approximates this distance by sampling vertices, edges, and triangles and taking root mean square value of the shortest distances from points in the surface X to the surface Y. The inverse wavelet transform has also to be taken before calculating the distance between the surfaces. Therefore, long off-line computations are required to obtain densely sampled operational D-R curves which allow bit allocation with fine granularity.

To speed-up calculation of the D-R curves, we use D-R function modeling. In our experiments, we use a Weibull model [27]. We have found that it can efficiently approximate a D-R curve of the wavelet coefficient tree sets or D-R curve of the whole compressed mesh. The Weibull model is

$$D(R) = a - be^{-cR^d} \tag{5.7}$$

where real numbers a, b, c, and d are parameters, which depend on the D-R characteristics of the source and the bitstream. As there are four parameters in the model (5.7), the parameters can be found by using at minimum four points. This significantly decreases the amount of computations. The model

can approximate both L^2 and PSNR curves. We use nonlinear least-squares regression to fit the model to D-R samples.

Figure 5.4 compares operational D-R curves in TM-MDC approach and their Weibull models. The Weibull models are $D(R) = 0.000169 - 0.002578e^{-0.8134R^{0.52805}}$ and $D(R) = 75.48 + 24.264e^{0.0153R^{0.675}}$ for L^2-distance and PSNR respectively.

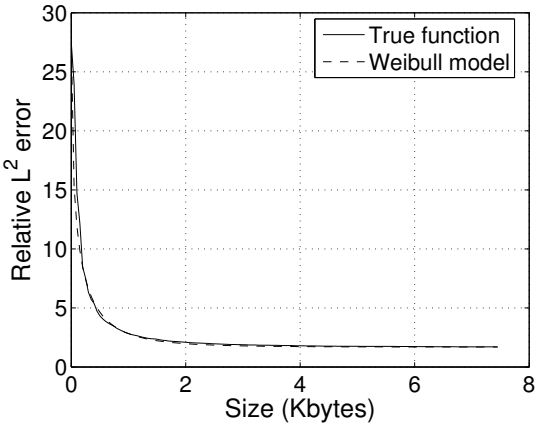
Figure 5.5 compares true operational $D(R)$ curves of PGC bitstream and their Weibull models. The models are $D(R) = -640.48 + 640.48e^{0.413R^{-1.597}}$ and $D(R) = 88.95 - 553.93e^{-1.101R^{0.125}}$ for L^2-distance and PSNR.

One can see that the Weibull model closely approximates the real data. Moreover, the Weibull model has a nice feature of convexity, which is desirable for bit allocation algorithms [96, 118]. As the model needs only four samples to approximate the D-R curve, only $4N$ D-R samples are required to generate D-R curves in case of TM-MDC. MD-FEC requires computation of only 4 samples to approximate the D-R curve. However, the number of samples can be increased to obtain the model which approximates the D-R curve more precisely.

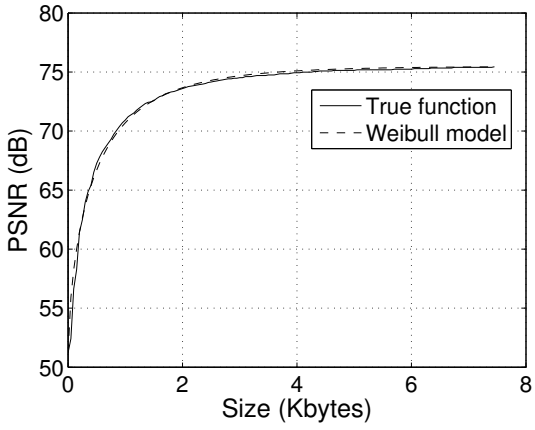
5.6 Simulation results

This section presents the simulation results for test models *Bunny* and *Venus head*. The reconstruction distortion is relative L^2 error, which is calculated with Metro tool [33]. Relative error is calculated by dividing L^2 distance to the original mesh bounding box diagonal. The error is shown in the figures in units of 10^{-4}. We also provide the same numbers in PSNR scale where $\text{PSNR} = 20\log_{10} peak/d$, *peak* is the bounding box diagonal, and d is the L^2 error. When all descriptions are lost and no reconstruction is possible, we calculate the distortion as the L^2 distance between the surface of the original mesh and a single point with coordinates $(0, 0, 0)$.

Four different coders have been compared. The first coder is the tree-based mesh MDC (*TM-MDC)*. The second coder is the one based on *MD-FEC*. The third coder is a *simple MDC* coder where the 3-D model is coded into four descriptions. In this coder, each description contains the coarsest mesh and one set of wavelet coefficient trees. This coder is identical to TM-MDC optimized for $P = 0$ (for $P = 0$, no redundant trees are included in the descriptions).

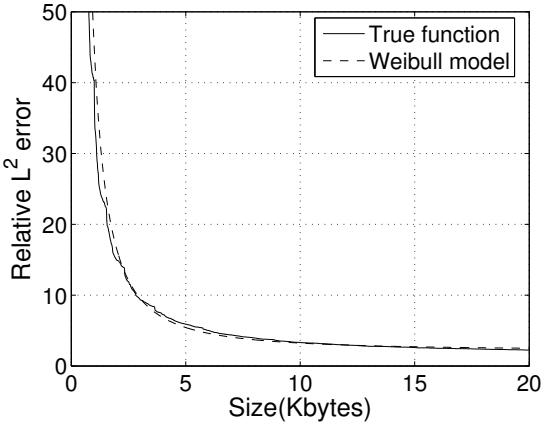


(a)

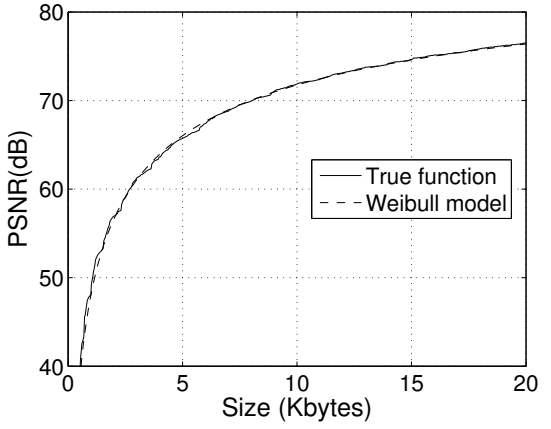


(b)

Figure 5.4: TM-MDC. Comparison between the Weibull model (10 samples) and operational D-R curve (relative L^2 error) of the *Set 1* of wavelet coefficient trees for *Bunny* model. (a) Relative L^2 error in units of 10^4; (b) PSNR.



(a)



(b)

Figure 5.5: PGC. Comparison between the Weibull model (4 samples) and operational $D(R)$ curve (L^2 distance) for *Bunny* model. (a) Relative L^2 error; (b) PSNR.

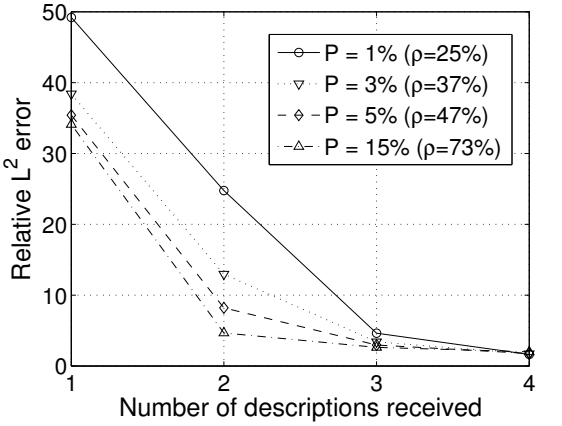
The sets of coefficient trees in both TM-MDC and simple MDC are formed by spatially close groups of trees of size 10. The last coder is *unprotected PGC* [83]. The distortion of unprotected PGC is calculated in the following way. The output bitstream of PGC coder is divided into N parts of equal size, where N is the number of descriptions in the MD coder that unprotected PGC is compared to. PGC produces the embedded bitstream. Thus, the received part can be used for reconstruction if all the packets containing earlier parts of the bitstream have been received. For example, if parts one, two, and four are received, only parts one and two are used for reconstruction. If part one is lost, the reconstruction is not possible.

In the figures, the label L^2 *dist.* in brackets after TM-MDC and MD-FEC corresponds to bit allocation using the L^2 distance obtained by the Metro tool [33], and the label *Weibull* corresponds to using the $D(R)$ curve obtained by D-R curve modeling. If none of the labels is sown in the figure, the D-R curve modeling has been used.

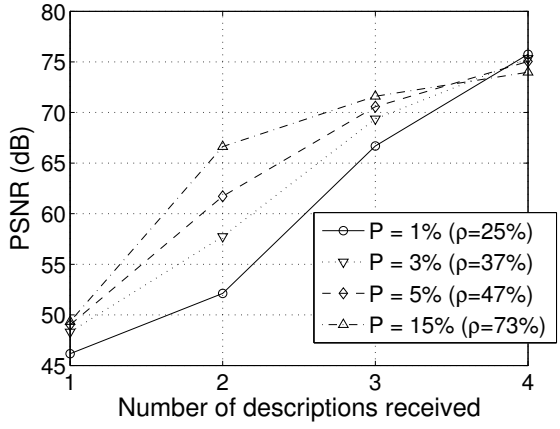
5.6.1 Reconstruction from different number of descriptions

Figures 5.6 and 5.7 show the average distortion for reconstruction with TM-MDC from different number of descriptions for models *Bunny* and *Venus head*. Model *Bunny* is coded at 22972 Bytes (5743 Bytes per each description) and model *Venus head* at 24404 Bytes (6101 Bytes per each description). Bit allocation is performed for each value of P. The corresponding value of redundancy ρ is given in brackets. One can see that the coder optimized for $P = 1\%$ shows the best performance when all the descriptions are received while the coder optimized for $P = 15\%$ shows the best results for reconstruction from one, two, and three descriptions.

Figure 5.8 compares reconstruction from different number of descriptions of TM-MDC and MD-FEC algorithms. The results are obtained for model *Bunny*, coded at 22972 bytes. Both MDC coders are optimized for $P = 5\%$. As one can see, both MD coders outperform unprotected PGC with the exception of one case, when all the descriptions are received. The TM-MDC achieves higher PSNR than MD-FEC for reconstruction from one description, but lower PSNR for reconstruction from three descriptions. This is probably connected with the fact that each description in TM-MDC includes the whole coarsest level geometry while in case of reconstruction from one description

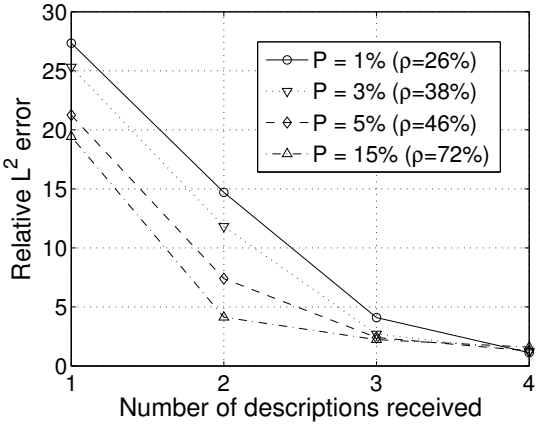


(a)

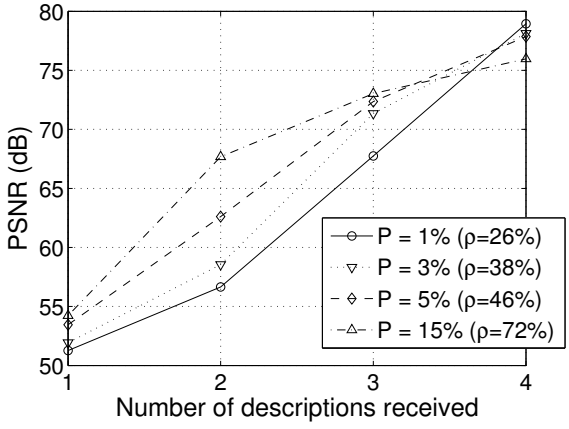


(b)

Figure 5.6: TM-MDC. Reconstruction of *Bunny* model from different number of descriptions. Results are given for bit allocations for different values of P) Redundancy ρ is shown in brackets. (a) Relative L^2 error in units of 10^4; (b) PSNR.



(a)



(b)

Figure 5.7: TM-MDC. Reconstruction of *Venus head* model from different number of descriptions. Results are given for bit allocations for different values of P). Redundancy ρ is shown in brackets. (a) Relative L^2 error in units of 10^4; (b) PSNR.

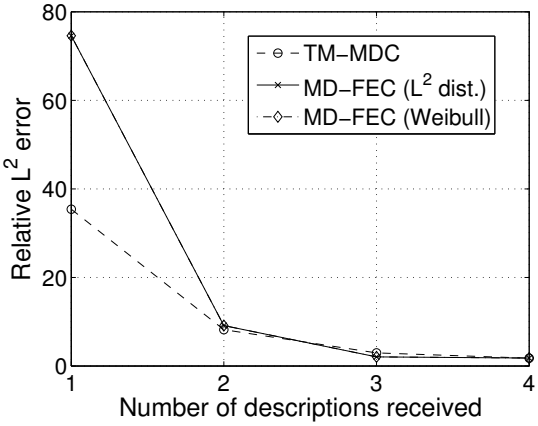
in MD-FEC, not all the bit-planes of the coarsest level geometry are present. Different bit allocation algorithms used in TM-MDC and MD-FEC can also be the reason for this difference in reconstruction PSNR. Another observation from Figure 5.8 is that the results of L^2 *distance* and *Weibull model* methods are indistinguishable. This proves the success of the modeling proposed in Section 5.5.

Figure 5.9 shows the average distortion for reconstruction from different number of descriptions for model *Bunny* encoded into 8 descriptions at total 25944 Bytes. The curves are generated for TM-MDC coder with bit allocation for $P = 5, 10\%$. The results for unprotected PGC are also shown. One can see that unprotected PGC achieves higher PSNR than TM-MDC just in one case: when all the descriptions are received. When some descriptions are lost, TM-MDC achieves much higher PSNR.

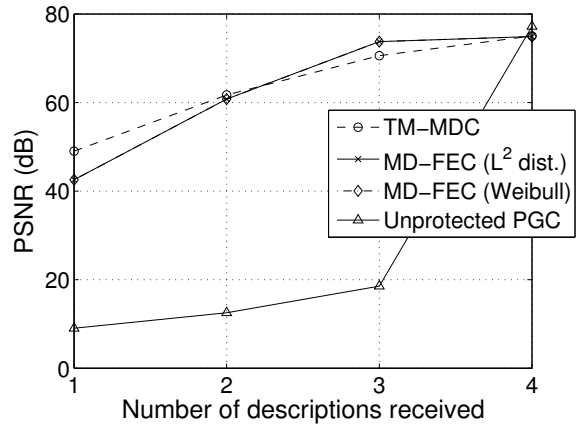
Figure 5.10 shows the trade-off between the central distortion D_0 and mean-side distortion $(D_1 + D_2)/2$. The plot is presented for model *Bunny* coded into two descriptions at total 21486, 16486, and 11486 Bytes.

5.6.2 Network performance of the proposed metods

Figures 5.11 and 5.12 compare the network performance of TM-MDC with the simple MD coder and unprotected PGC. The results are calculated for $P = 0, 1, 3, 5, 10, 15, 20\%$. The TM-MDC performs bit allocation for each value of P. For the simple MD coder, the value of redundancy is always fixed. These figures also compare the performance of TM-MDC using "true" operational D-R curves for bit allocation and TM-MDC using D-R curve modeling. For each P, we have performed 100000 simulations of packet loss patterns and calculated the average distortion. For $P = 0$, TM-MDC and simple MD coder show the same performance because the TM-MDC coder optimized for $P = 0$ is in fact the same as a simple MD coder. Unprotected PGC shows slightly better performance than MD coders in the error-free environment ($P = 0$). However, in the network with packet losses, performance of simple MDC coder and unprotected PGC dramatically decreases while the reconstruction quality of TM-MDC shows only mild degradation. For $P = 20\%$, the PSNR of the optimized TM-MDC coder is 10 dB higher than PSNR of the simple MD coder and 30 to 40 dB higher than PSNR of unprotected PGC. Because of the great difference in distortion values between the MD coders and unprotected



(a)



(b)

Figure 5.8: Model *Bunny*. Comparison of TM-MDC and MD-FEC in reconstruction from different number of descriptions. (a) Relative L^2 error; (b) PSNR.

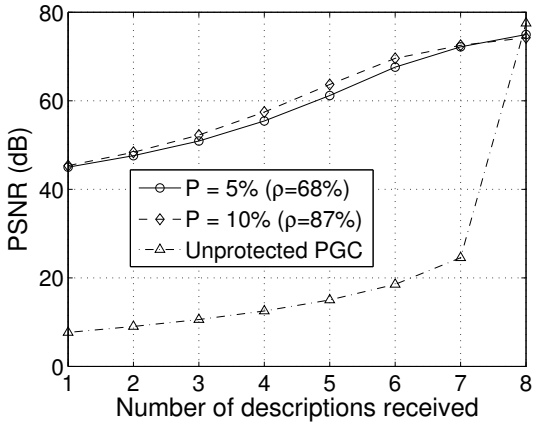


Figure 5.9: TM-MDC. Model *Bunny* encoded into 8 descriptions at total 25944 Bytes. Reconstruction from different number of descriptions. Compared to unprotected PGC.

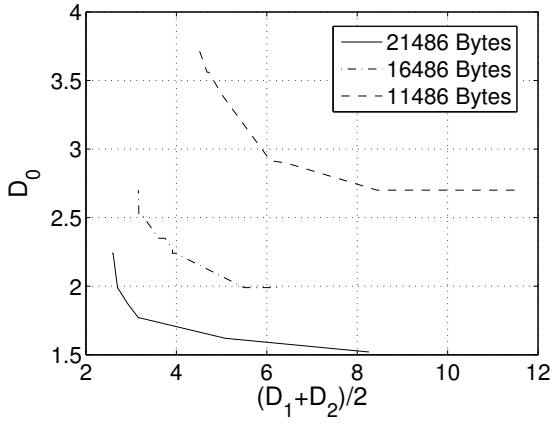


Figure 5.10: TM-MDC. Model *Bunny* encoded into two descriptions at total 21486, 16486, and 11486 Bytes. Central vs mean-side distortion.

SPIHT, the results for the latter are shown only in PSNR scale. TM-MDC using Weibull model for D-R function shows very close performance to TM-MDC using the real operational D-R curves. This again proves the validity of D-R modeling. Therefore, D-R curve modeling can be used to decrease the amount of computations at the preparatory stage without significant decrease in network performance.

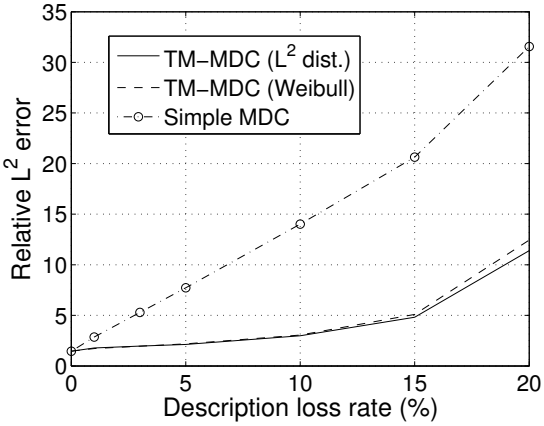
Figure 5.13 compares the network performance of TM-MDC, MD-FEC (Weibull model), and unprotected PGC for model *Bunny*. As one can see from Figure 5.13, MD-FEC slightly outperforms TM-MDC, and they both significantly outperform unprotected PGC. MD-FEC slightly outperforms TM-MDC for the following reason. In the memoryless channel model, loss patterns with one description lost are more frequent than those with three descriptions lost; and for the formers, MD-FEC achieves better reconstruction. The difference in bit allocation of TM-MDC and MD-FEC may be connected with the fact that only a part of coarsest mesh geometry bit-planes is transmitted in MD-FEC in the beginning of bitstream. On the contrary, in TM-MDC, all the coarsest mesh information is transmitted in the beginning of the bitstream.

5.6.3 Visual illustrations

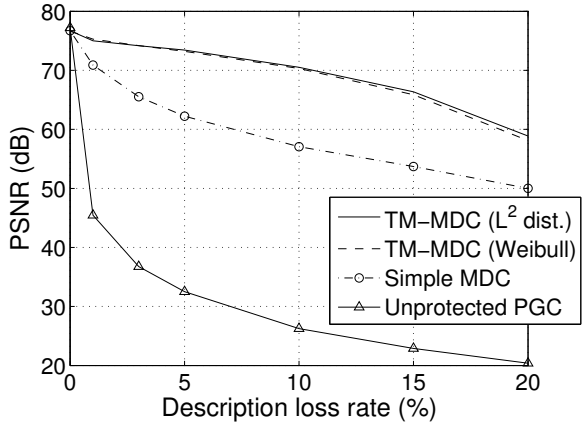
Figures 5.14 and 5.15 show visual reconstruction results for TM-MDC. In Figure 5.14, test model *Bunny* is encoded into four descriptions with the redundancy $\rho = 63\%$ at total 22972 Bytes and reconstructed from one, two, tree, and four descriptions. One can see from the figure that even the reconstruction from one description provides acceptable visual quality. Figure 5.15 shows the visual reconstruction results for *Venus head* model, encoded into four descriptions with redundancy $\rho = 53\%$ at total 24404 Bytes. Both models are encoded with spatially close trees grouping with group size 10.

5.7 Chapter conclusions and discussion

We have proposed two algorithms for coding 3D meshes. Both MD coders are based on PGC scheme [83]. In PGC, wavelet transform is applied to the semi-regular remeshed model, and wavelet coefficient trees are coded with SPIHT algorithm. TM-MDC exploits the redundancy in the form of redundant partially-coded trees of wavelet coefficients and the duplicated coarsest level

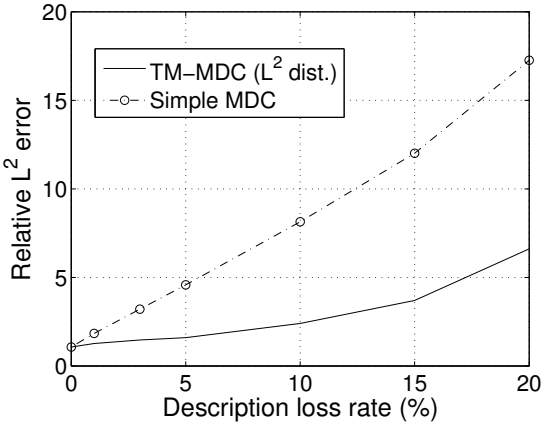


(a)

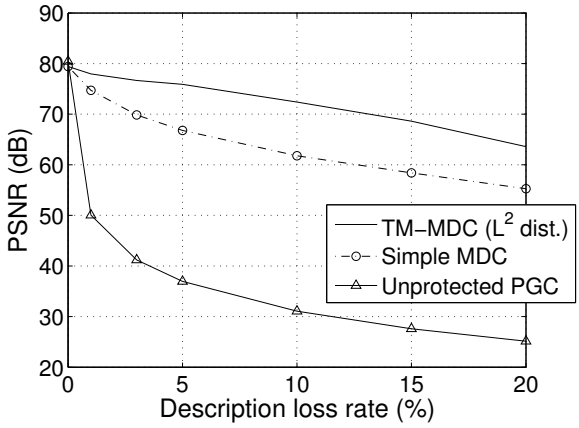


(b)

Figure 5.11: Model *Bunny*. Comparison of network performance of TM-MDC with simple MDC scheme and unprotected PGC. The results for TM-MDC with D-R curve modeling are given as TM-MDC (Weibull). (a) Relative L^2 error in units of 10^4; (b) PSNR.



(a)



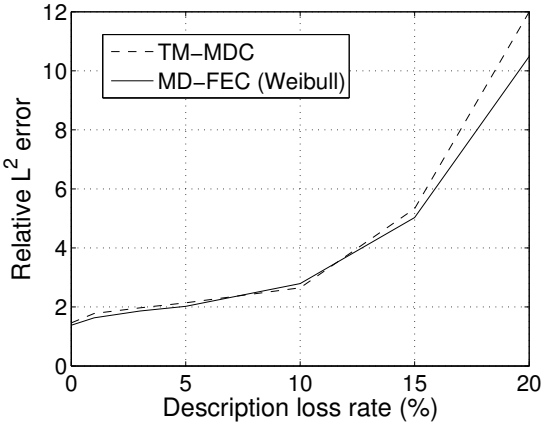
(b)

Figure 5.12: Model *Venus head*. Comparison of network performance of TM-MDC with simple MDC scheme and unprotected PGC. (a) Relative L^2 error in units of 10^4; (b) PSNR.

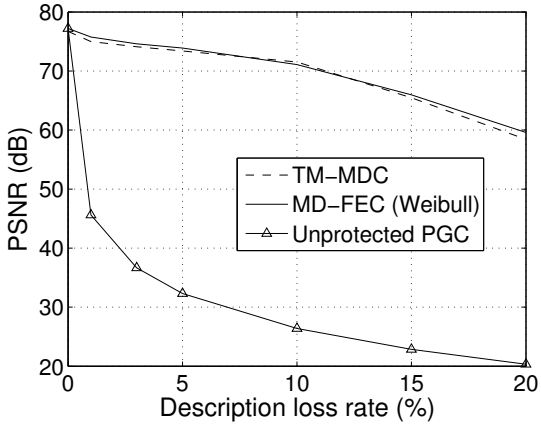
mesh. Two strategies of grouping the trees were studied, and spatially close grouping has been chosen as the one providing better visual quality. The second proposed algorithm is MDC-FEC for 3D mesh coding. Stronger codes are allocated to the beginning of SPIHT stream, while less or no FEC bits are allocated to the end of bitstream.

Two methods have been proposed to obtain $D(R)$ curves for bit allocation. From these two methods, L^2 distance turns out to give slightly better results in terms of the expected distortion. Modeling D-R curves performs just slightly worse than using the real L^2 distance. However, D-R curve modeling significantly decreases the amount of computations at the preparatory stage.

We have observed that TM-MDC and MD-FEC demonstrate similar performance in the network environment with losses. However, both methods have their advantages and disadvantages with respect to each other. For example, MD-FEC needs only one compressed bitstream to perform bit allocation for any number of descriptions while TM-MDC needs to generate different compressed bitstreams in case the number of descriptions has changed. The TM-MDC coder includes the whole coarsest level geometry in each description whereas MD-FEC spreads bit-planes of the coarsest level geometry according to their importance in the compressed bitstream. The TM-MDC can potentially provide region of interest (ROI) coding of 3D meshes. In this case, more important trees can be included in all the descriptions with higher bitrate. Moreover, one has to take into account that the results for TM-MDC have been obtained with grouping spatially close trees. However, grouping spatially disperse trees increases the PSNR on approximately 2 dB (see Figure 5.3). Therefore, various grouping strategies may be investigated.



(a)



(b)

Figure 5.13: Model *Bunny*. Comparison of the network performance of the MD-FEC and TM-MDC. (a) Relative L^2 error; (b) PSNR.

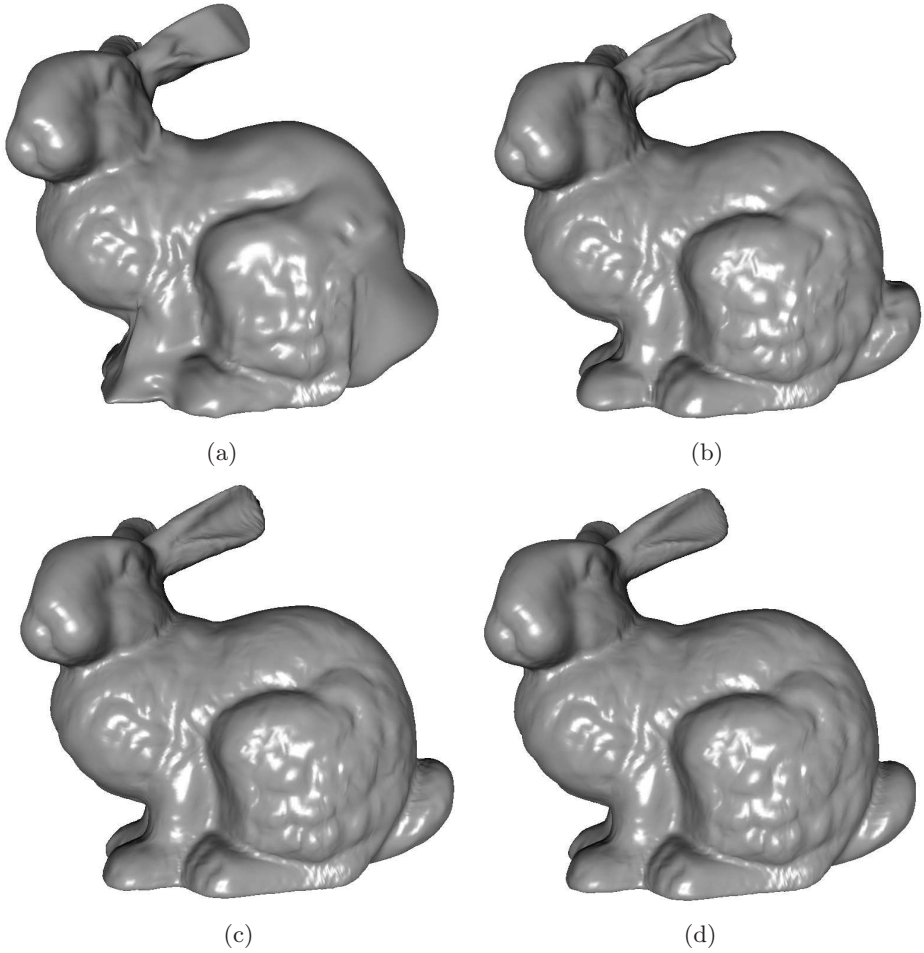


Figure 5.14: TM-MDC. Reconstruction of *Bunny* model from (a) one description (48.36 dB), (b) two descriptions (63.60 dB), (c) three descriptions (71.44 dB), (d) four descriptions (74.33 dB).

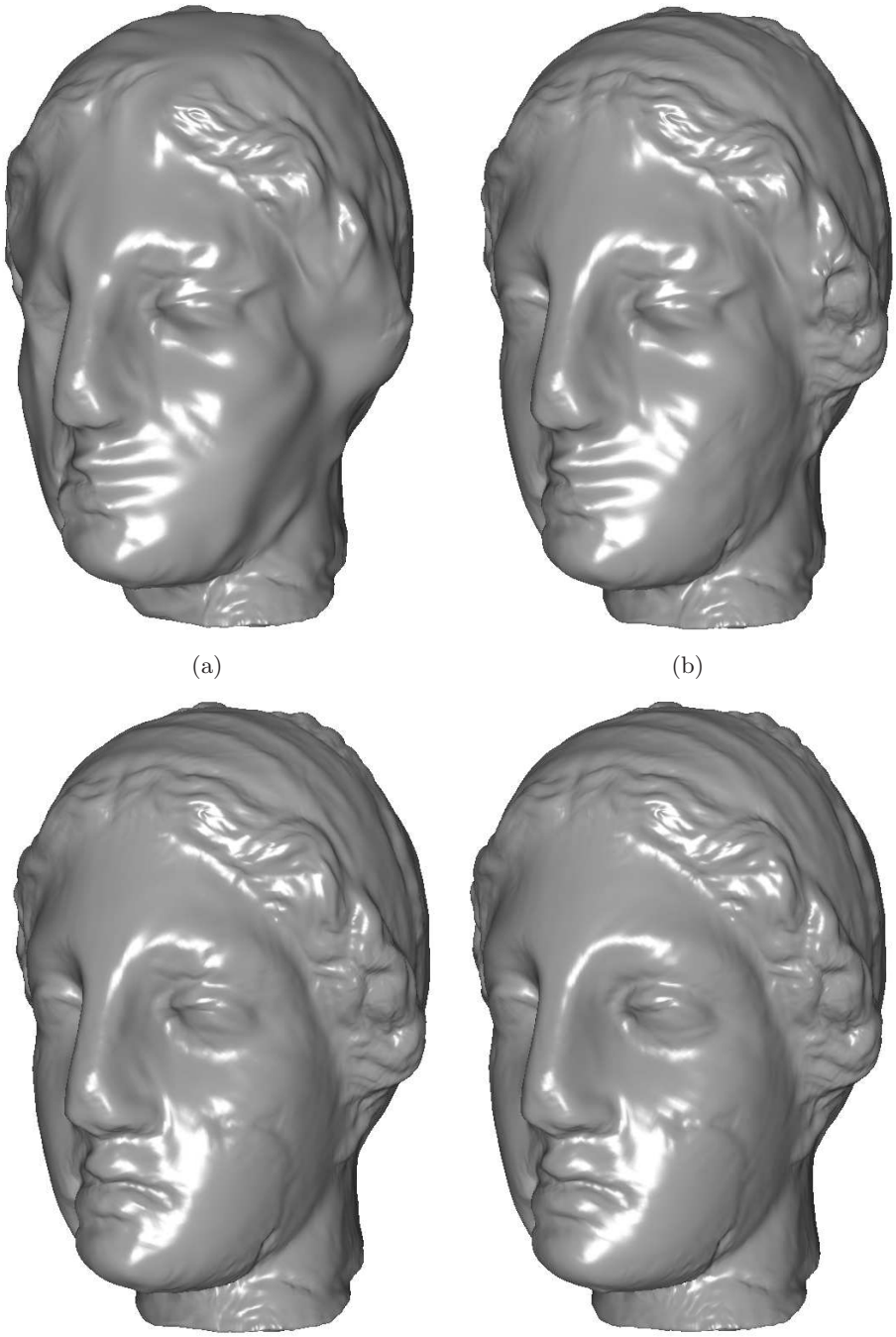


Figure 5.15: TM-MDC. Reconstruction of *Venus Head* model from (a) one description (53.97 dB), (b) two descriptions (65.18 dB), (c) three descriptions (72.51 dB), (d) four descriptions (77.08 dB).

Chapter 6

Conclusions

This thesis has introduced new MDC schemes for images, video, stereoscopic video, and 3D meshes. The motivation for using multiple description coding arises from the vulnerability of the compressed bitstream to channel errors and erasures when transmitting multimedia data over error-prone channels. The goals of this thesis were to prove the validity of MDC as an error resilience approach and demonstrate the application of MDC to different types of visual data, from images to 3D meshes.

In Chapter 2, a two-stage approach has been developed for image coding. The first stage of the method employs B-spline interpolation to obtain a coarse image approximation (shaper) which fits as much information as possible in the limited bit budget. The shaper is compressed and included in both descriptions. Then, properly interpolated, the quantized coarse image is subtracted from the original image, yielding a residual, which contains image details. A checkerboard subsampling of block transform coefficients is applied to divide the residual image between two descriptions.

Two block transforms have been compared for coding the residual image. The DCT-based coder shows higher PSNR while the LOT-based coder produces the picture with less annoying artifacts when reconstructing it from one description. The postprocessing algorithm in the DCT-based coder reduces blocking artifacts and increases the quality of side reconstruction. The proposed bit allocation optimizes the coder performance with respect to the target bitrate and probability of description loss. The proposed MDC method outperforms two other MD coders based on JPEG [29, 156].

In Chapter 3, we have proposed an MDC scheme for video coding which

does not use motion-compensated prediction. The coder is targeted for mobile devices. This scheme adapts the 2-stage image coding method to coding of video. The proposed coder exploits 3D transforms to remove correlation in the video sequence. Coding is done in two stages: the first stage produces a coarse approximation (shaper) of the video sequence. The second stage encodes the residual sequence. The shaper is obtained by pruned 3D-DCT and the residual signal is compressed by 3D-DCT or a hybrid 3D block transform. The redundancy is introduced by including the shaper in both descriptions. The amount of redundancy is easily controlled by the shaper quantization step. A simple bit allocation algorithm has been derived and a closed-form solution of the optimization problem has been found. Thus, bit allocation algorithm can be used in the encoder without a significant increase in the encoding complexity.

The complexity of the proposed coder has been estimated. The results state that the encoder is up to two times less complex than the optimized H.263. As the proposed MD video coder has low computational complexity, it can be used in mobile devices with low computational power and limited battery life. The coder has been shown to outperform the MDTC video coder [116] and two simple MD coders based on H.263+. The coder performs especially well in the low-redundancy region.

Chapter 4 has addressed MDC of stereoscopic video. Two drift-free MDC approaches have been proposed. Both approaches produce balanced descriptions and provide stereoscopic reconstruction with acceptable quality in case of one channel failure for the price of moderate, in the range of 10-50%, redundancy. The performance of these approaches depends on the characteristics of the stereoscopic video sequence. The first approach, called SS-MDC, performs better for sequences with lower inter-view correlation, while the MS-MDC approach performs better for sequences with higher inter-view correlation. A criterion for switching between the approaches has been found and it is used by the encoder to choose the approach that provides better performance for the particular sequence or the part of the sequence.

Future research on this topic may include developing algorithms for the estimation of the lost frames in MS-MDC, optimization of the proposed schemes subject to probability of packet loss and adapting the MDC algorithms to multi-view video coding. The MS-MDC approach can be applied to multi-view video. However, the inter-view prediction in multi-view video is less

efficient compared to stereoscopic video due to the greater distance between the cameras. This should be taken into account when designing MDC schemes for multi-view video.

In Chapter 5, two MDC techniques for 3D meshes have been presented. The proposed coders are based on progressive geometry compression, which is efficient for highly detailed 3D meshes. The first approach, TM-MDC, exploits the redundancy in the form of partially coded trees of wavelet coefficients and the duplicated coarsest-level mesh. Two strategies of grouping the trees were investigated, and spatially close grouping has been chosen as the one providing better visual quality. The second algorithm is MD-FEC for coding 3D meshes. Both algorithms have been shown capable of providing a flexible number of descriptions and can be adjusted for the varying packet (description) loss rate. Graceful degradation of reconstruction quality is observed when increasing the description loss rate.

The D-R function modeling has been introduces for both 3D mesh coding methods. The proposed modeling drastically decreases the amount of computations at the preparatory stage with the price of only slight decrease in the averaged reconstruction distortion.

The plans for future research on TM-MDC include optimization of bit allocation of the coarsest level mesh, investigating the possibilities for partitioning the coarsest mesh geometry and searching for optimal patterns for grouping wavelet coefficient trees.

This thesis has studied MDC of different types of visual information. Several practical MDC schemes have been developed, which provide efficient error resilience in the lossy network environment. The proposed methods can be used to facilitate error-resilient transmission of visual information over wireless networks and the Internet.

Bibliography

[1] A. Aksay, C. Bilen, E. Kurutepe, T. Ozcelebi, G. Bozdagi Akar, R. Civanlar, and M. Tekalp, *Temporal and spatial scaling for stereoscopic video compression*, in Proc. EUSIPCO'06 (Sept. 4-8, Florence, Italy), 2006.

[2] E. Akyol, A. M. Tekalp, and M. R. Civanlar, *Scalable multiple description video coding with flexible number of descriptions*, in Proc. IEEE Int. Conf. Image Processing, vol. 3, Sept. 2005, pp. 712–715.

[3] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, *Priority encoding transmission*, IEEE Trans. Information Theory **42** (1996), 1737 – 1744.

[4] P. Alliez and M. Desbrun, *Progressive encoding for lossess transmission of triangle meshes*, in Proc. ACM SIGGRAPH 2001, 2001, pp. 198–205.

[5] ———, *Valence-driven connectivity encoding for 3D meshes*, in Proc. EUROGRAPHICS 2001, vol. 20(3), 2001, pp. 480–489.

[6] P. Alliez and C. Gotsman, *Recent advances in compression of 3D meshes*, in Proc. Symposium on Multiresolution in Geometric Modeling, 2003.

[7] J. Apostolopoulos, *Error-resilient video compression through the use of multiple states*, in Proc. Int. Conf. Image Processing, vol. 3, Sept. 2000, pp. 352–355.

[8] J. Apostolopoulos, W. Tan, S. Wee, and G. Wornell, *Modelling path diversity for multiple description video communication*, in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol. 3, May 2002, pp. 2161–2164.

[9] J. Apostolopoulos and S. Wee, *Unbalanced multiple description video communication using path diversity*, in Proc. Int. Conf. Image Processing, vol. 1, Oct. 2001, pp. 966–969.

[10] I. Bajic and J. Woods, *Domain-based multiple description coding of images video*, IEEE Trans. Image Process. **12** (2003), 1211–1225.

[11] M. Bakr and A. Salama, *Implementation of 3D-DCT based video encoder/decoder system*, in Proc. IEEE MWSCAS-2002, vol. 2, Aug. 2002, pp. II–13–16.

[12] B. Balasubramaniyam, E. Edirisinghe, and H. Bez, *Extended H.264 codec for stereoscopic video coding*, in Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems XII, vol. 5664, 2005.

[13] ———, *A parallel encoding approach to H.264 stereo video*, in Proc. IET Int. Conf. Visual Information Engineering, Sept 2006, pp. 195–200.

[14] M. O. Bici and G. Bozdagi Akar, *Multiple description scalar quantization based 3D mesh coding*, in Proc. IEEE Int. Conf. Image Processing (Atlanta, US), Oct. 2006.

[15] M. O. Bici, A. Norkin, G. Bozdagi Akar, A. Gotchev, and J. Astola, *Multiple description coding of 3D geometry with forward error correction codes*, in Proc. 3DTV-CON2007 (Kos, Greece), May 2007.

[16] C. Bilen, A. Aksay, and G. Bozdagi Akar, *A multi-view video codec based on H.264*, in Proc. IEEE Conf. Image Processing (ICIP) (Oct. 8-11, Atlanta, USA), 2006.

[17] A. Bogomjakov and C. Gotsman, *Universal rendering sequences for transparent vertex caching of progressive meshes*, Computer Graphics Forum **21** (2002), 137–148.

[18] N. Boulgouris, K. Zachariadis, A. Leontaris, and M. Strintzis, *Drift-free multiple description coding of video*, in Proc. IEEE Int. Workshop Multimedia Signal Processing, vol. 1, 2001, pp. 105–110.

[19] N. V. Boulgouris and M. G. Strintzis, *A family of wavelet-based stereo image coders*, IEEE Trans. on Cirquits Syst. Video Technol. **12** (2002), no. 10, 898–903.

[20] S. Boussakta and H. Alshibami, *Fast algorithm for the 3D-DCT*, IEEE Trans. Signal Processing **52** (2004), 992–1001.

[21] N. Bozinovic and J. Konrad, *Motion analysis in 3D DCT domain and its application to video coding*, IEEE Trans. Circuits Syst. Video Technol. **20** (2005), 510–528.

[22] A. Bruchstein, M. Elad, and R. Kimmel, *Down-scaling for better transform compression*, IEEE Trans. Image Processing **12** (2003), no. 9, 1132–1144.

[23] A. Burg, R. Keller, J. Wassner, N. Felber, and W. Fichtner, *A 3D-DCT real-time video compression system for low complexity single-chip VLSI implementation*, in Proc. of the MoMuC2000 (Tokyo), 2000, pp. 1B–5–1.

[24] P. Burt and E. Adelson, *The laplacian pyramyd as a compact image code*, IEEE Trans. Communications **31** (1983), 532–540.

[25] C.-S.Kim, R.-C.Kim, and S.-U. Lee, *Matching pursuits multiple description coding for wireless video*, IEEE Trans. Circuits Syst. Video Technol. **11** (2001), 1011–1021.

[26] R. Chan and M. Lee, *3D-DCT quantization as a compression technique for video sequences*, in Proc. IEEE Conf. Virtual Systems and Multimedia (VSMM'97), Sept. 1997, pp. 188–196.

[27] Y. Charfi, R. Hamzaoui, and D. Saupe, *Model-based real-time progressive transmission of images over noisy channel*, in Proc. WCNC'03 (New Orleans, LA), Mar. 2003, pp. 347–354.

[28] J. Y. Chen, Z. Liwei, and S. Q. Ding, *The effect of JPEG coding scheme on the perceived quality of 3D images*, SID symposium, vol. 29, 1998, pp. 1211–1214.

[29] K.-P. Choi and K.-Y. Lee, *An efficient multiple description coding using whitening transform*, IEICE Trans. Fundamentals **E86-A** (2003), no. 6, 1382–1389.

[30] D.-M. Chung and Y. Wang, *Multiple description image coding based on lapped orthogonal transforms*, in Proc. IEEE Int. Conf. Image Processing (ICIP'98), vol. 1, Oct 1998, pp. 664–668.

[31] ———, *Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms*, IEEE Trans. Circuits Syst. Video Technol. **9** (1999), 895–908.

[32] ———, *Lapped orthogonal transforms designed for error-resilient image coding*, IEEE Trans. Circuits Syst. Video Technol. **12** (2002), 752–764.

[33] P. Cignoni, C. Rocchini, and R. Scopigno, *Metro: Measuring error on simplified surfaces*, Computer Graphics Forum **17** (1998), 167–174.

[34] D. Comas, R. Singh, and A. Ortega, *Rate-distortion optimization in a robust video transmission based on unbalanced multiple description coding*, in Proc. IEEE Int. Workshop Multimedia Signal Processing, Oct. 2001, pp. 581–586.

[35] D. Comas, R. Singh, A. Ortega, and F. Marques, *Unbalanced multiple-description video coding with rate-distortion optimization*, EURASIP Journal Applied Signal Processing (2003), no. 1, 81–90.

[36] G. Cote, B. Erol, M. Gallant, and F. Kossentini, *H.263+: video coding at low bitrates*, IEEE Trans. Circuits Syst. Video Technol. **8** (1998), 849–866.

[37] M. Deering, *Geometry compression*, in Proc. ACM SIGGRAPH'95 Conf., 1995, pp. 13–20.

[38] S. Diggavi, N. Sloane, and V. Vaishampayan, *Asymmetric multiple description lattice vector quantizers*, IEEE Trans. Inform. Theory **48** (2002), no. 1, 174–191.

[39] L.-F. Ding, S.-Y. Chien, Y.-W. Huang, Y.-L. Chang, and L.-G. Chen, *Stereo video coding system with hybrid coding based on joint prediction scheme*, in Proc. IEEE ISCAS'05, vol. 6, May 2005, pp. 6082–6085.

[40] I. Dinstein, M. G. Kim, A. Henik, and J. Tzelgov, *Compression of stereo images using subsampling transform coding*, Optical Engineering **30** (1991), no. 9, 1359–1364.

[41] I. Dinstein, M.G. Kim, J. Tzelgov, and A. Henik, *Compression of stereo image and the evaluation of its effects on 3-D perception*, in Proc. SPIE, vol. 1153, Stereoscopic Displays and Virtual Reality Systems XII, 1989.

[42] N. Dyn, D. Levin, and J. Gregory, *A butterfly subdivision scheme for surface interpolation with tension control*, ACN Trans. on Graphics **9** (1990), no. 2, 160–169.

[43] J. Ellinas and M. Sangriotis, *Stereo video coding based on interpolated motion and disparity estimation*, in Proc. IEEE ISPA03, vol. 1, Sept. 2003, pp. 301–306.

[44] W. Equitz and T. Cover, *Successive refinement of information*, IEEE Trans. Inform. Theory **37** (1991), no. 2, 269–275.

[45] C. Fehn, *Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV*, in Proc. SPIE Stereoscopic Displays and Virtual Reality Systems XI (San Jose, CA, USA,), Jan. 2004.

[46] C. Fehn, P. Kauff, M. Op de Beeck, F. Ernst, W. IJsselsteijn, M. Pollefeys, L. Van Gool, E. Ofek, and I. Sexton, *An evolutionary and optimised approach on 3D-TV*, in Proc. Int. Broadcast Conf. (Amsterdam, The Netherlands), Sept. 2002, pp. 357–365.

[47] M. Fleming and M. Effros, *Generalized multiple description vector quantization*, in Proc. Data Compression Conference, Mar. 1999, pp. 3–12.

[48] N. Franchi, M. Fumagalli, R. Lancini, and S. Tubaro, *A space domain approach for multiple description video coding*, in Proc. IEEE Int. Conf. Image Processing, vol. 3, Sept. 2003, pp. 253–256.

[49] N. Franci, M. Fumagalli, and R. Lancini, *Flexible redundancy insertion in a polyphase down sampling multiple description image coding*, in Proc. IEEE Int. Conf. Multimedia Expo, vol. 2, Aug. 2002, pp. 605–608.

[50] M. Fumagalli, D. Sagetong, and A. Ortega, *Estimation of erased data in a H.263 coded stream by using unbalanced multiple description coding*, in Proc. Int. Conf. Multimedia Expo, July 2003, pp. 13–16.

[51] J. Goshi, R. Ladner, A. Mohr, E. Riskin, and A. Lippman, *Unequal loss protection for H.263 compressed video*, in Proc. IEEE Data Compression Conf., Mar. 2003, pp. 73 – 82.

[52] A. Gotchev, *Spline and wavelet based methods for signal and image processing*, Doctor of Technology thesis, Tampere University of Technology, Sept. 2003.

[53] A. Gotchev, K. Egiazarian, G. Marchokov, and T. Saramäki, *A near least squares method for image decimation*, in Proc. Int. Conf. Image Processing (ICIP'03) (Barcelona, Spain), Sept. 2003.

[54] V. Goyal, *Multiple description coding: compression meets the network*, IEEE Signal Processing Mag. **18** (2001), 74–93.

[55] V. Goyal, J. Kelner, and J. Kovacevic, *Multiple description vector quantization with a coarse lattice*, IEEE Trans. Inform. Theory **48** (2002), no. 3, 781–788.

[56] V. Goyal and J. Kovacevic, *Optimal multiple description transform coding of gaussian vectors*, in Proc. IEEE Data Compression Conf., Mar. 1998, pp. 388–397.

[57] ———, *Generalized multiple description coding with correlating transforms*, IEEE Trans. Inform. Theory **47** (2001), no. 6, 2199–2224.

[58] V. Goyal, J. Kovacevic, R. Arean, and M. Vetterli, *Multiple description transform coding of images*, in Proc. Int. Conf. Image Processing, vol. 1, Oct. 1998, pp. 674–678.

[59] V. Goyal, J. Kovacevic, and M. Vetterli, *Multiple description transform coding: Robustness to erasures using tight frame expansions*, in Proc. IEEE Int. Symp. Inform. Theory (Cambridge, MA), Aug. 1998, p. 408.

[60] ———, *Quantized frame expansions as source-channel codes for erasure channels*, in Proc. IEEE Int. Conf. Data Compression, Mar. 1998, pp. 326–335.

[61] V. Goyal, M. Vetterli, and N. Thao, *Quantized overcomplete expansions in $\mathbb{R}^n$: Analysis, synthesis, and algorithms*, IEEE Trans. Inform. Theory **44** (1998), no. 1, 16–31.

[62] R. Gray and D. Neuhoff, *Quantization*, IEEE Trans. Information Theory **44** (1998), no. 6, 2325–2383.

[63] S. Gumhold and W. Strasser, *Real time compression of triangle mesh connectivity*, in Proc. SIGGRAPH'98 Conf., 1998, pp. 133–140.

[64] I. Guskov, K. Vidimce, W. Sweldens, and P. Schroeder, *Normal meshes*, in Proc. 27th annual conference on Computer graphics and interactive techniques, July 2000, pp. 95–102.

[65] H.264/AVC reference software, *http://iphome.hhi.de/suerring/tml/*.

[66] S. S. Hemami, *Reconstruction-optimized lapped orthogonal transforms for robust image transmission*, IEEE Trans. Circuits Syst. Video Technol. **6** (1996), 168–181.

[67] H. Hoppe, *Progressive meshes*, in Proc. ACM SIGGRAPH'96, 1996, pp. 99–108.

[68] ISO/IEC, *Information technology - open systems interconnection - basic reference model: the basic model*, ISO/IEC 7498-1: 1994(E), Nov. 1994.

[69] ISO/IEC JTC1/SC29/WG11, *Text of ISO/IEC FDIS 23002-3 representation of auxiliary video and supplemental information*, Doc. N8768, Marrakech, Morocco, January 2007.

[70] ———, *Text of ISO/IEC 13818-1:2003/FDAM2 carriage of auxiliary data*, Doc. N8799, Marrakech, Morocco, January 2007.

[71] ISO/IEC JTC/SC29/WG11/N2502a, *MPEG-4 video verfification model v.9.1*, Oct. 1998.

[72] ITU-T, *Video coding for low bit rate communication*, ITU-T Rec. H.263; version 1, Nov. 1995; version 2, Jan. 1998; version 3, Nov. 2000.

[73] ITU-T and ISO/IEC JTC 1, *Generic coding of moving pictures and associated audio information part 2: Video*, ITU-T Rec. H.222.0; ISO/IEC 13818-1 (MPEG 2 Systems), Nov. 1994.

[74] ———, *Advanced video coding for generic audiovisual services*, ITU-T Rec. H.264; ISO/IEC 14496-10 AVC, 2003, most recent version: 2005.

[75] ITU-T and ISO/IEC, *Information technology JPEG 2000 image coding system part 1: Core coding system*, ITU-T Rec. T.800; ISO/IEC 15444-1.

[76] H. Jafarkhani and V. Tarokh, *Multiple description trellis coded quantization*, in Proc. IEEE Int. Conf. Image Processing (ICIP98), vol. 1, Oct. 1998, pp. 669–673.

[77] P. Jaromersky, X. Wu, Y. Chiang, and N. Memon, *Multiple-description geometry compression for networked interactive 3D graphics*, in Proc. ICIG'2004, Dec. 2004, pp. 468–471.

[78] W. Jiang and A. Ortega, *Multiple description coding via scaling-rotation transform*, in Proc. Int. Conf. Acoustics Speech Signal Processing, vol. 5, Mar. 1999, pp. 2419–2422.

[79] M. Karczewicz and R. Kurceren, *The SP- and SI-frames design for H.264/AVC*, IEEE Trans. Circuits Syst. Video Technol. **13** (2003), 637–644.

[80] Z. Karni, A. Bogomjakov, and C. Gotsman, *Efficient compression and rendering of multi-resolution meshes*, in Proc. IEEE Int. Conf. Visualization (Boston, US), Oct. 2002.

[81] J. Kelner, V. Goyal, and J. Kovacevic, *Multiple description lattice vector quantization: variations and extension*, in Proc. IEEE Data Compression Conf. (Snowbird, UT), Mar. 2000, pp. 480–489.

[82] A. Khodakovsky and I. Guskov, *Normal mesh compression*, submitted for publication, http://www.multires.caltech.edu/pubs/compression.pdf.

[83] A. Khodakovsky, P. Schröder, and W. Sweldens, *Progressive geometry compression*, in Proc. Computer Graphics, Siggraph, 2000, pp. 271–278.

[84] C.-S. Kim and S.-U. Lee, *Multiple description motion coding algorithm for robust video transmission*, in Proc. ISCAS 2000, vol. 4, May 2000, pp. 717 – 720.

[85] ———, *Multiple description coding of motion fields for robust video transmission*, IEEE Trans. Circuits Syst. Video Technol. **11** (2001), no. 9, 999–1010.

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

[87] S. D. Kim, J. Yi, H. M. Kim, and J. B. Ra, *A deblocking filter with two separate modes in block-based video coding*, IEEE Trans. Circuits Syst. Video Technol. **9** (1999), 156–160.

[88] J. Koivusaari and J. Takala, *Simplified three-dimensional discrete cosine transform based video codec*, *SPIE-IS&T Electronic Imaging, Multimedia on Mobile Devices* (San Jose, CA), vol. 5684, Jan. 2005, pp. 11–20.

[89] V. Lappalainen, A. Hallapuro, and T.D. Hamalainen, *Optimization of emerging H.26L video encoder*, in Proc. IEEE Workshop Signal Processing Systems, 2001, pp. 406–415.

[90] A. Lee, W. Sweldens, P. Shroeder, L. Cowsar, and D. Dobkin, *MAPS: multiresolution adaptive parametrization of surfaces*, in Proc. SIGGRAPH 98, 1998, pp. 95–104.

[91] C. Loop, *Smooth subdivision surfaces based on triangles*, Master's thesis, University of Utah, Department of Mathematics, 1987.

[92] H. S. Malvar and D. H. Staelin, *The LOT: transform coding without blocking effects*, IEEE Trans. Acoustics, Speech, and Signal Processing **37** (1989), 553–559.

[93] H. Man, R. Queiroz, and M. Smith, *Three-dimensional subband coding techniques for wireless video communications*, IEEE Trans. Circuits Syst. Video Technol. **12** (2002), 386–397.

[94] A. Miguel, A. Mohr, and E. Riskin, *SPIHT for generalized multiple description coding*, in Proc. IEEE Int. Conf. Image Processing, vol. 3, Oct. 1999, pp. 842 – 846.

[95] A. Mohr, E. Riskin, and R. Ladner, *Generalized multiple description coding through unequal loss protection*, in Proc. IEEE Int. Conf. Image Processing, vol. 1, 1999, pp. 411 – 415.

[96] ———, *Approximately optimal assignment for unequal loss protection*, in Proc. ICIP'00, 2000, pp. 367–370.

[97] A. Munos, T. Blu, and M. Unser, *Least squares image resizing using finite differences*, IEEE Trans. Image Processing **10** (2001), 1365–1378.

[98] A. Norkin, A. Aksay, C. Bilen, G. Bozdagi Akar, A. Gotchev, and J. Astola, *Schemes for multiple description coding of stereoscopic video*,

in Proc. MRCS 2006, Istanbul, Turkey. LNCS, vol. 4105, Sept. 2006, pp. 730–737.

[99] A. Norkin, M. O. Bici, G. Bozdagi Akar, A. Gotchev, and J. Astola, *Wavelet-based multiple description coding of 3-D geometry*, in Proc. VCIP'07, Proceedings SPIE (San-Jose, US), vol. 6508, Jan. 2007, pp. 65082I–1 –65082I–10.

[100] A. Norkin, M. O. Bici, A. Aksay, C. Bilen, A. Gotchev, G. B. Akar, K. Egiazarian, and J. Astola, *Three-dimensional television: capture, transmission, and display*, ch. Multiple description coding and its relevance to 3DTV, Springer, Heidelberg, 2007.

[101] A. Norkin, A. Gotchev, K. Egiazarian, and J. Astola, *Multiple description image coders using whitening transforms: analysis and comparative study*, in Proc. TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP'04 (Vienna, Austria), Sept. 2004, pp. 225–232.

[102] _____, *A low-complexity multiple description video coder based on 3D-transforms*, in Proc. EUSIPCO 2006 (Florence, Italy), Sept. 4-8, 2006.

[103] _____, *Two-stage multiple description image coders: analysis and comparative study*, Signal Processing: Image Communication **21/8** (2006), 609–625.

[104] _____, *Low-complexity multiple description coding of video based on 3D block transforms*, EURASIP Journal on Embedded Systems **2007** (2007), Article ID 38631, 11 pages, doi:10.1155/2007/38631.

[105] M. Orchard and G. Sullivan, *Overlapped block motion compensation: an estimation-theoretic approach*, IEEE Trans. Image Processing **3** (1994), no. 5, 693–699.

[106] M. Orchard, Y.Wang, V. Vaishampayan, and A. Reibman, *Redundancy rate distortion analysis of multiple description image coding using pairwise correlating transforms*, in Proc. Int. Conf. Image Processing (Santa Barbara, CA), Oct. 1997, pp. 608–611.

[107] L. Ozarow, *On a source-channel coding problem with two channels and three receivers*, Bell Syst. Tech. J. **59** (1980), no. 10, 1909–1921.

[108] R. Pajarola and J. Rossignac, *Compressed progressive meshes*, IEEE Trans. Visualizations and Computer graphics **5** (2000), no. 1, 79–93.

[109] S.-C. Pei and C.-L. Lai, *Very low bit-rate coding algorithm for stereo video with spatiotemporal HVS model and binary correlation disparity estimator*, IEEE Journal on Selected Areas in Communications **16** (1998), 98 – 107.

[110] J. Peng, C.-S. Kim, and C.-C. J. Kuo, *Technologies for 3D mesh compression: a survey*, Journal of Visual Communication and Image Representation **16** (2005), 688–733.

[111] J. Popovic and H. Hoppe, *Progressive simplicial complexes*, in Proc. ACM SIGGRAPH'97, 1997, pp. 217–224.

[112] K. Rao and R. Yip, *Discrete cosine transform: algorithms, advantages, applications*, Academic Press Limited, 12–28 Oval Road, London, 1990.

[113] S. Regunathan and K. Rose, *Efficient prediction in multiple description video coding*, in Proc. IEEE Int. Conf. Image Processing, vol. 1, Sept. 2000, pp. 1020 – 1023.

[114] A. Reibman, H. Jafarkhani, Y. Wang, and M. Orchard, *Multiple description video using rate-distortion splitting*, in Proc. IEEE Int. Conf. Image Processing (ICIP2001), vol. 1, Oct. 2001, pp. 978–981.

[115] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, and R. Puri, *Multiple description coding for video using motion-compensated prediction*, in Proc. IEEE Int. Conf. Image Processing (ICIP99), vol. 3, Oct. 1999, pp. 837–841.

[116] _____, *Multiple description coding for video using motion-compensated temporal prediction*, IEEE Trans. Circuits Syst. Video Technol. **12** (2002), 193–204.

[117] J. Reichel, H. Schwarz, and M. Wien, *Scalable video coding - working draft 3*, JVT-P201, Poznan, PL, 24-29 July, 2005.

[118] E. Riskin, *Optimum bit allocation via generalized BFOS algorithm*, IEEE Trans. Inform. Theory **37** (1991), 400–4002.

[119] L. Roberts, *TMN 8 (H.263+) encoder/decoder, version 3.0*, Signal Processing and Multimedia Lab., Univ. British Columbia, May 1997.

[120] J. Rossignac, *Edgebreaker: connectivity compression for triangle meshes*, IEEE Trans. Visualizations and Computer graphics **5** (1999), no. 1, 47–61.

[121] D. Rusanovskyy and K. Egiazarian, *Postprocessing for three-dimensional discrete cosine transform based video coding*, in Proc. 7th Int. Conf.

Advanced Concepts for Intelligent Vision Systems, ACIVS (Antwerp., Belgium), Sept. 2005, pp. 618–625.

[122] P. Sagetong and A. Ortega, *Optimal bit allocation for channel-adaptive multiple description coding*, in Proc. VCIP 2000 (San Jose, CA), Jan. 2000, pp. 53–63.

[123] A. Said and W. Pearlman, *A new, fast, and efficient image codec based on set partitioning in hierarchical trees*, IEEE Trans. Circuits Syst. Video Technol. **6** (1996), no. 3, 243–250.

[124] S. Saponara, L. Fanucci, and P. Terreni, *Low-power VLSI architectures for 3D discrete cosine transform (DCT)*, in Proc. IEEE MWSCAS-2003, vol. 3, Dec. 2003, pp. 1567–1570.

[125] M. van der Schaar and D. Turaga, *Multiple description scalable coding using wavelet-based motion compensated temporal filtering*, in Proc. IEEE Int. Conf. Image Processing, vol. 2, Sept. 2003, pp. 489–492.

[126] A. Schertz, *Source coding of stereoscopic television pictures*, in Proc. IEE Inter. Conf. Image Processing and its Applications (Maastricht, The Netherlands), 1992, pp. 462–464.

[127] A. Scodras, *Fast discrete cosine transform pruning*, IEEE Trans. Signal Processing **42** (1994), 1833–1837.

[128] A. Secker and D. Taubman, *Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting*, in Proc. IEEE Int. Conf. Image Processing **2** (2001), 1029 – 1032.

[129] S. A. Segall, *Study upsampling/downsampling for spatial scalability*, JVT-Q083, Nice, FR, PL, 14-21 October, 2005.

[130] S. Servetto, V. Vaishampayan, and N. Sloane, *Multiple description lattice vector quantization*, in Proc. IEEE Data Compression Conf. (Snowbird, UT), Mar. 1999, pp. 13–22.

[131] S. D. Servetto, K. Ramchadran, V. Vaishampayan, and K. Nahrstedt, *Multiple-description wavelet based image coding*, in Proc. IEEE Int. Conf. Image Processing (Chicago, IL), 1998.

[132] _____, *Multiple-description wavelet based image coding*, IEEE Trans. Image Processing **9** (2000), no. 5, 813–826.

[133] R. Singh and A. Ortega, *Erasure recovery in predictive coding environments using multiple description coding*, in Proc. IEEE 3d Workshop Multimedia Signal Processing, Sept. 1999, pp. 333–338.

[134] A. Smolic, K. Mueller, N. Stefanovski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, and A. Koz, *Coding algorithms for 3DTV - a survey*, IEEE Trans. Circuits Syst. Video Technol. **17** (2007), no. 11, 1606–1621.

[135] S. Somasundaram and K. Subbalakshmi, *3-D multiple description video coding for packet switched networks*, Proc. IEEE Int. Conf. Multimedia and Expo (ICME'03), vol. 1, July 2003, pp. I – 589–592.

[136] L. B. Stelmach, W. J. Tam, and D. V. Meegan, *Stereo image quality: effects of spatio-temporal resolution*, IEEE Trans. on Circuits Systems Video Technol. **10** (2000), 188–193.

[137] L. B. Stelmach, W. J. Tam, D. V. Meegan, A. Vincent, and P. Corriveau, *Human perception of mismatched stereoscopic 3D inputs*, in Proc. Int. Conf. Image Processing (Vancouver, Canada), Sept. 2000.

[138] X. Tang and A. Zakhor, *Matching pursuits multiple description coding for wireless video*, IEEE Trans. Circuits Syst. Video Technol. **12** (2002), 566–575.

[139] G. Taubin, A. Gueziec, W. Horn, and F. Lazarus, *Progressive forest split compression*, in Proc. ACM SIGGRAPH'98, 1998, pp. 123–132.

[140] G. Taubin and J. Rossignac, *Geometric compression through topological surgery*, ACM Trans. Graphics **17(2)** (1998), 84 – 115.

[141] P. Thevenaz, T. Blu, and M. Unser, *Interpolation revisited*, IEEE Trans. Medical Imaging **19** (2000), 739–758.

[142] T. Tillo and G. Olmo, *A novel multiple description coding scheme compatible with the JPEG2000 decoder*, IEEE Signal Processing Letters **11** (2004), 908–911.

[143] C. Touma and C. Gotsman, *Triangle mesh compression*, in Proc. Graphics Interface (Vancouver, BC, Canada), Jun. 1998.

[144] D. Tzovaras, N. Grammalidis, and M. G. Strintzis, *Disparity field and depth map coding for multiview 3D image generation*, Signal Processing: Image Communication **11** (1998), 205–230.

[145] V. Vaishampayan, *Design of multiple description scalar quantizers*, IEEE Trans. Inform. Theory **39** (1993), no. 3, 821–834.

[146] ———, *Application of multiple description codes to image and video transmission over lossy networks*, in Proc. 7th Int. Workshop Packet Video (Brisbane, Australia), Mar. 1996, pp. 55–60.

[147] V. Vaishampayan and J.-C. Batllo, *Asymptotic analysis of multiple description quantizers*, IEEE Trans. Inform. Theory **44** (1998), no. 1, 278–284.

[148] V. Vaishampayan and J. Domaszewicz, *Design of entropy-constrained multiple description scalar quantizers*, IEEE Trans. Inform. Theory **40** (1994), no. 1, 245–250.

[149] V. Vaishampayan and S. John, *Balanced interframe multiple description video compression*, in Proc. IEEE Int. Conf. Image Processing (ICIP99), vol. 3, Oct. 1999, pp. 812 – 816.

[150] V. Vaishampayan, N. Sloane, and S. Servetto, *Multiple description vector quantization with lattice codebooks: design and analysis*, IEEE Trans. Inform. Theory **47** (2001), no. 5, 1718–1734.

[151] A. Vetro, Y. Su, H. Kimata, and A. Smolic, *Joint draft 1.0 on multi-view video coding*, Oct. 2006.

[152] Y. Wang and S. Lin, *Error-resilient coding using multiple description motion compensation*, in Proc. IEEE Int. Workshop Multimedia Signal Processing (MMSP01), Oct. 2001, pp. 441–446.

[153] ———, *Error-resilient coding using multiple description motion compensation*, IEEE Trans. Circuits Syst. Video Technol. **12** (2002), no. 6, 438–452.

[154] Y. Wang, M. Orchard, and A. Reibman, *Multiple description image coding for noisy channels by pairing transform coefficients*, in Proc. IEEE First Workshop Multimedia Signal Processing, June 1997, pp. 419–424.

[155] ———, *Optimal pairwise correlating transform for multiple description coding*, in Proc. Int. Conf. Image Processing, vol. 1, Oct. 1998, pp. 679 – 683.

[156] Y. Wang, M. Orchard, V. Vaishampayan, and A. Reibman, *Multiple description coding using pairwise correlating transforms*, IEEE Trans. Image Processing **10** (2001), no. 3, 351–366.

[157] Y. Wang, A. Reibman, and S. Lin, *Multiple description coding for video delivery*, Proceedings of IEEE **93** (2005), 57–70.

[158] Y. Wang, A. Reibman, M. Orchard, and H. Jafarkhani, *An improvement to multiple description transform coding*, IEEE Trans. Image Processing **50** (2002), no. 11, 2843–2854.

[159] Y. Wang and C. Wu, *A mesh-based multiple description coding method for network video*, in Proc. 18th Int. Conf. Advanced Information Networking and Application (AINA), vol. 1, Sept. 2004, pp. 549 – 554.

[160] S. Wenger, G. Knorr, J. Ott, and F. Kossentini, *Error resilience support in H.263+*, IEEE Trans. Circuits Syst. Video Technol. **8** (1998), no. 7, 867–877.

[161] W. Woo and A. Ortega, *Optimal blockwise dependent quantization for stereo image coding*, IEEE Trans. on Cirquits Syst. Video Technol. **9** (1999), 861–867.

[162] L. Yan, Z. Zhaoyang, and A. Ping, *Stereo video coding based on frame estimation and interpolation*, IEEE Trans. on Broadcasting **49** (2003), 14 – 21.

[163] L. Yaroslavsky, A. Happonen, and Y. Katiyi, *Discrete signal sinc-interpolation in DCT domain: fast algorithms*, in Proc. Int. TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP'02 (Toulouse, France), Sept. 2002, pp. 179–185.

[164] B.-L. Yeo and B. Liu, *Volume rendering of DCT-based compressed 3D scalar data*, IEEE Trans. Visualization and Computer Graphics **1** (1995), 29–49.

[165] K. Yu, J. Lv, J. Li, and S. Li, *Practical real-time video codec for mobile devices*, in Proc. IEEE Int. Conf. on Multimedia and Expo (ICME 2003), vol. 3, July 2003, pp. 509–512.

[166] M. Yu, Z. Wenqin, G. Jiang, and Z. Yin, *An approach to 3D scalable multiple description video coding with content delivery networks*, Proc. IEEE Int. Workshop VLSI Design and Video Technology, May, 28-30 2005, pp. 191–194.

[167] D. Zorin, P. Shroeder, and W. Sweldens, *Interactive multiresolution mesh editing*, in Proc. SIGGRAPH 97, 1997, pp. 189–192.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland