TK
7881.4
.S78
2007

LAYERED MULTIPLE DESCRIPTION CODING OF MULTI-CHANNEL AUDIO SIGNALS

by

Katarina Stojadinović
Dipl. Ing., University of Belgrade, 1999

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2007

©Katarina Stojadinović 2007

PROPERTY OF
RYERSON UNIVERSITY LIBRARY

UMI Number: EC53578

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform EC53578

Copyright 2009 by ProQuest LLC

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

LAYERED MULTIPLE DESCRIPTION CODING OF MULTI-CHANNEL AUDIO SIGNALS

Katarina Stojadinović

Master of Applied Science

Department of Electrical and Computer Engineering

Ryerson University, Toronto, 2007

Abstract

In this study, we investigate efficient coding of multi-channel audio signals for transmission over packet networks. The techniques studied and developed as part of this research are based on redundancy coding and aim to achieve robustness with respect to packet losses. The resulting algorithm also addresses the needs of network clients with varying access bandwidths; the algorithm generates multi-layer encoded data streams which can range from basic mono to full multi-channel surround audio. Loss mitigation is achieved by applying multiple description coding technique based on the priority encoding transmission packetization scheme. The hierarchy of the transmitted data is derived from a statistical analysis of the multi-channel audio signal. Inter-channel correlations form the basis for estimating the multi-channel audio signal from the received descriptions at the decoder.

Acknowledgment

I would like to thank my respectable supervisor, Prof. Mehmet Zeytinoglu, for the privilege to work with him.

Contents

1	Introduction	1
1.1	Packet Networks	2
1.1.1	Connectionless and Connection-Oriented Transport	4
1.1.2	Delay and Loss in Packet Transmission	5
1.2	Problem Description and Motivation	6
1.3	Summary of Contributions	9
1.4	Thesis Outline	10
2	Background	12
2.1	Multiple Description Coding: A Primer	12
2.2	Three Examples of MDC Realizations	15
2.2.1	MD Scalar Quantizers	15
2.2.2	MD Lattice Vector Quantizers	17
2.2.3	MDC with Correlating Transforms	19
2.3	Applications	21
2.4	Multiple Description Audio Coding	22
2.4.1	Perceptual Audio Coders	22
2.4.2	Related Work	24
2.5	Layered Multiple Description Coding	26
2.6	Priority Encoding Transmission	27
3	Layered Multiple Description Coding of Multi-Channel Audio Signals	31

3.1	Multi-Channel Audio Signals	31
3.2	Inter-Channel Correlations	34
3.3	Layer Descriptions	37
3.4	Encoder	43
3.4.1	Segmentation	44
3.4.2	Layering	44
3.4.3	Normalization	45
3.4.4	Audio Coding	45
3.4.5	Packetization	47
3.5	Decoder	50
3.5.1	Unpacketization	50
3.5.2	Audio Decoding	53
3.5.3	Estimation	53
3.5.4	Synthesis	54
4	Results	56
4.1	Block Size M	56
4.1.1	Distortion vs. Number of Descriptions Received	56
4.1.2	Distortion vs. Block Size	59
4.2	Effects of Packet Loss Probability	60
4.3	Listening Tests	62
4.3.1	Test Procedures	62
4.3.2	Results	64
4.4	Discussion	66
4.4.1	Distortion	67
4.4.2	Subjective Sound Quality	68
5	Conclusions and Future Work	71
5.1	Conclusions	71
5.2	Future Work	72

Appendices	74
A Reed-Solomon Codes	74
A.1 Galois Fields in the Construction of $\mathcal{RS}$ Codes	75
A.2 $\mathcal{RS}$ Encoding	76
A.3 $\mathcal{RS}$ Decoding with Focus on Erasures	77
B Elements of H	78
B.1 Case $l = 1$	78
B.2 Case $l = 2$	79
B.3 Case $l = 3$	81
B.4 Case $l = 4$	82
B.5 Case $l = 5$	84
C Sound Test Database	85
Bibliography	87

List of Tables

1.1	Digital multi-channel audio signal parameters [21].	8
1.2	Standard audio formats on DVD Video disc [23, 24].	8
2.1	Mapping of the original quantization level index onto the quantization indices of the MD quantizer for $K = 16$	17
3.1	Default channel ordering in a multi-channel audio file.	34
4.1	Normalized distortion values averaged over all sound files.	58
4.2	Peak and average normalized distortion values determined with 10 sound files of 105 second duration with $M = 3000$ samples.	61
4.3	Grading scales, [25].	62
4.4	Listening tests results: (a) average MOS grades and (b) absolute grades as a function of the number of descriptions received.	65
4.5	Listening tests results: (a) average MOS grades and (b) absolute grades as a function of the packet loss probability.	66
4.6	Listening tests results: (a) average MOS grades and (b) absolute grades for various sound files with different number of descriptions received and packet loss probabilities.	67
B.1	Dependence of $\mathcal{G}_1$ and $\mathcal{G}_2$ estimates on received layers.	80
B.2	Layers used in the estimation of $\widehat{\mathbf{L}}\mathbf{s}_n$ and $\widehat{\mathbf{R}}\mathbf{s}_n$	82
B.3	Layers used in the estimation of $\widehat{\mathbf{L}}\mathbf{s}_n$ and $\widehat{\mathbf{R}}\mathbf{s}_n$	83

List of Figures

1.1	Packet switching network, [3].	3
1.2	Loss model [8].	6
2.1	Two-transmission channel, three-receiver MD scenario [2].	14
2.2	Multiple Description Lattice Vector Quantization [14].	18
2.3	An example of cascade MD correlating transform [19].	20
2.4	Generic perceptual audio coder [25].	23
2.5	Hierarchical correlating transform [29].	25
2.6	A sample PET packetization.	30
3.1	5.1 multi-channel audio configuration.	33
3.2	Statistical measures of the absolute values of cross-correlation coefficients corresponding to pairs of audio channel signals extracted from the audio database.	36
3.3	Transmission of multi-channel signal $\mathbf{x}$	38
3.4	Encoder block diagram.	43
3.5	An example of mp3 frames [21].	46
3.6	The packetization process.	48
3.7	Decoder structure for the case $\mathcal{Q} = \{P_2, P_3\}$	50
3.8	Recovered codeword data structure for the case $\mathcal{Q} = \{P_2, P_3\}$	51
4.1	Normalized distortions values as a function of the number of descriptions received calculated over all sound files in the database.	58

4.2	Peak and average of normalized distortion values as a function of the blocksize at a packet loss probability of 10^{-1}	59
4.3	Peak and average normalized distortion values determined with 10 sound files of 105 second duration with $M = 3000$ samples.	61
4.4	Listening test interface, [39].	63

List of Symbols and Notation Used

$\mathbf{u}, \mathbf{v}, \dots$	vectors.
$\mathbf{U}, \mathbf{V}, \dots$	matrices.
$\lceil \beta \rceil$	the smallest integer greater than or equal to β .
$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^T \mathbf{u}$	represents the inner product of the column vectors $\mathbf{u}$ and $\mathbf{v}$.
$\hat{u}$	estimate of u .
$\mathbf{u}^T$	transpose of the vector $\mathbf{u}$.
$\text{diag}[u_1, \dots, u_N]$	$N \times N$ diagonal matrix with the diagonal elements $u_1, \dots, u_N$.
$E[\mathbf{X}]$	expected value of the random element/variable $\mathbf{X}$.
$\text{trace}[\mathbf{V}]$	the “trace” operator applied to an $N \times N$ square matrix $\mathbf{V}$ with elements v_{ij} is: $\sum_{i=1}^N v_{ii}$.
$\mathbf{I}$	identity matrix.
v_{mn}	the element of the matrix $\mathbf{V}$ at position (m, n) .
$\mathbf{V}_{:j}$	j th column of matrix $\mathbf{V}$.
$\mathbf{V}_{i:}$	i th row of matrix $\mathbf{V}$.
α_n	the maximum sample magnitude determined from the n th layer signal vector $\mathbf{y}_n = [y_{1,n}, \dots, y_{5,n}]^T$.
$\mathbf{C}$	front-center audio channel signal sequence.
$\mathbf{C}[k]$	k th element of $\mathbf{C}$.
$\mathbf{C}_n$	n th block (segment) of $\mathbf{C}$.
$\hat{\mathbf{C}}$	estimate of $\mathbf{C}$.

$\hat{\mathbf{C}}_n$	estimate of $\mathbf{C}_n$.
$\mathcal{C}$	a 5×5 diagonal matrix with diagonal elements that are either “0” or “1” representing the communication network. The presence of a “1” (“0”) on the main diagonal indicates that the corresponding layer has been received (lost).
$\mathcal{D}$	distortion.
$\mathbf{G}$	linear transformation matrix that defines the layer descriptions such that $\mathbf{y} = \mathbf{G}\mathbf{x}$.
$\mathbf{H}$	linear estimator such that $\hat{\mathbf{x}} = \mathbf{H}\mathbf{z}$.
h_{ij}	the element of $\mathbf{H}$ at position (i, j) .
$\mathbf{H}(l)$	linear estimator matrix with optimized elements for the case when only the first l layer descriptions are available.
$\mathbf{L}$	front-left audio channel signal sequence.
$\mathbf{L}[k]$	k th element of $\mathbf{L}$.
$\mathbf{L}_n$	n th block (segment) of $\mathbf{L}$.
$\hat{\mathbf{L}}$	estimate of $\mathbf{L}$.
$\hat{\mathbf{L}}_n$	estimate of $\mathbf{L}_n$.
$\mathbf{Ls}$	left-surround audio channel signal sequence.
$\mathbf{Ls}[k]$	k th element of $\mathbf{Ls}$.
$\mathbf{Ls}_n$	n th block (segment) of $\mathbf{Ls}$.
$\hat{\mathbf{Ls}}$	estimate of $\mathbf{Ls}$.
$\hat{\mathbf{Ls}}_n$	estimate of $\mathbf{Ls}_n$.
$\mathcal{L}_{i,n}$	n th block of the i th normalized layer description.
$\mathcal{L}_{i,n}^c$	mp3-compressed version of $\mathcal{L}_{i,n}$.
$\mathcal{L}_{i,n}^d$	mp3-decoded version of $\mathcal{L}_{i,n}^c$.
$\mathcal{L}_n^d$	mp3-decoded layer description matrix with i th row $\mathcal{L}_{i,n}^d$.

$m(i, k)$	k th symbol of the $\mathcal{L}_{i,n}^c$ block.
M	number of samples in each block.
N	size of $\mathcal{L}_{i,n}^c$ in words, where each word is 16 bits long.
$p(i, q, r)$	r th parity symbol in the q th codeword associated with $\mathcal{L}_{i,n}^c$.
P_i	i th transmitted packet.
$\mathcal{P}$	set of transmitted packets, $\mathcal{P} = \{P_1, \dots, P_5\}$.
$\mathcal{Q}$	subset of $\mathcal{P}$ representing the set of packets that arrive at the destination.
$\mathbf{R}$	front-right audio channel signal sequence.
$\mathbf{R}[k]$	k th element of $\mathbf{R}$.
$\mathbf{R}_n$	n th block (segment) of $\mathbf{R}$.
$\widehat{\mathbf{R}}$	estimate of $\mathbf{R}$.
$\widehat{\mathbf{R}}_n$	estimate of $\mathbf{R}_n$.
$\mathbf{R}_s$	right-surround audio channel signal sequence.
$\mathbf{R}_s[k]$	k th element of $\mathbf{R}_s$.
$\mathbf{R}_{s_n}$	n th block (segment) of $\mathbf{R}_s$.
$\widehat{\mathbf{R}}_s$	estimate of $\mathbf{R}_s$.
$\widehat{\mathbf{R}}_{s_n}$	estimate of $\mathbf{R}_{s_n}$.
$\mathbf{R}_{xx}$	autocorrelation matrix of $\mathbf{x}$.
$\mathbf{R}_{xx,n}$	autocorrelation matrix of $\mathbf{x}_n$.
$\mathcal{RS}(k, n)$	Reed-Solomon code that encodes a message of size k symbols into a codeword of size n symbols.
$\mathbf{W}$	low frequency audio channel sequence.
$\mathbf{W}[k]$	k th element of $\mathbf{W}$.

$\mathbf{W}_i$	n th block (segment) of $\mathbf{W}$.
$\widehat{\mathbf{W}}$	estimate of $\mathbf{W}$.
$\widehat{\mathbf{W}}_n$	estimate of $\mathbf{R}_n$.
$\mathbf{x}$	multi-channel audio signal $[\mathbf{L}^T \mathbf{R}^T \mathbf{C}^T \mathbf{L}\mathbf{s}^T \mathbf{R}\mathbf{s}^T]^T$.
$\widehat{\mathbf{x}}$	estimate of $\mathbf{x}$.
$\mathbf{x}_n$	n th block (segment) of $\mathbf{x}$, where each block is M -samples long.
$\widehat{\mathbf{x}}_n$	estimate of $\mathbf{x}_n$.
$\mathbf{y}_i$	i th layer description vector/sequence.
$\mathbf{y}$	layer description matrix with $\mathbf{y}_i = \mathbf{y}_i$.
$\mathbf{y}_{i,n}$	n th block of $\mathbf{y}_i$.
$\mathbf{y}$	layer description matrix $[\mathbf{y}_1^T \cdots \mathbf{y}_5^T]^T$.
$\mathbf{y}_n$	n th block of $\mathbf{y}$.
$\mathbf{z}$	vector of received layer descriptions $\mathbf{C}\mathbf{y}$.

List of Abbreviations

AC-3	Dolby Adaptive Transform Coder-3, 8
ATM	Asynchronous Transfer Mode network, 4
BCH	Bose-Chaudhuri-Hocquenghem codes, 74
CI	Confidence Interval, 35
CRC	Cyclic Redundancy Check, 46
DTS	Digital Theater System, 8
DVD	Digital Versatile Disc or Digital Video Disc, 7
GF	Galois Field, 75
ITU-R	International Telecommunication Union - Radiocommunication Sector, 32
LC	Layered Coding, 26
LFE	Low Frequency Enhancement, 32
LMDC	Layered Multiple Description Coding, 27
MDC	Multiple Description Coding, 9
MDLVQ	Multiple Description Lattice Vector Quantization, 17

MDSQ	Multiple Description Scalar Quantization, 16
MOS	Mean Opinion Score, 62
MP2	MPEG-1 Audio Layer II, 8
mp3	MPEG-1 Audio Layer III, 45
MPEG	Moving Picture Experts Group, 8
MPEG-2 AAC	Second MPEG standard; Advanced Audio Coding, 8
MPEG-2 BC	Second MPEG standard; Backward Compatible (with MPEG-1 bitstream), 8
QoS	Quality of Service, 5
SDG	Subjective Difference Grade, 63

Chapter 1

Introduction

MULTI-CHANNEL audio signals are an essential component of multimedia content such as digitally encoded movies on digital video disks. High-fidelity reproduction of the multi-channel audio signals contributes significantly to the enjoyment of such multimedia content. Therefore, efficient encoding, transmission and decoding of multi-channel audio signals are integral to all multimedia delivery systems. Data rates required to represent multi-channel, high-fidelity digital audio signals can be significantly higher than the coding of speech or even stereo audio signals. While compressing the audio data before storage or transmission is an acceptable (and universally employed) option, care should be paid to minimize the distortion introduced at the encoder. In multi-channel audio signal processing this issue becomes even more pronounced due to high level of interdependency among the various audio channel signals.

Today, multimedia content is increasingly being delivered in real-time over packet networks. These distribution models introduce their own challenges such as having to deal with packet losses, or excess delay due to traffic congestion in the network. While the increase of available transmission bandwidth provides a partial solution, we recognize that enhanced network throughput and reliability also increase users' expectations on the quality of reproduced multimedia content. An encoder can compress multi-channel audio data and encode it in a format that is suitable for transmission

over packet networks. A properly designed encoder can attempt to achieve these objectives while minimizing the side effects of heavy signal processing such as processing power, encoder and decoder complexity, end-to-end delay, and sensitivity to packet losses [1].

In this study, we investigate efficient coding of multi-channel audio signals for transmission over packet networks. The techniques studied and developed as part of this research are based on redundancy coding and aim to achieve robustness with respect to packet losses. The resulting algorithm also addresses the needs of network clients with varying access bandwidths; the algorithm generates multi-layer encoded data streams which can range from basic mono to full multi-channel surround audio.

Section 1.1 provides a brief introduction to packet networks as a background for the specific problem addressed in this work. The information contained in this section is mainly based on [3].

Section 1.2 describes the focus of this study: real-time transmission of multi-channel audio signals over packet networks. We first discuss issues associated with the packet networks such as delay and limitations on packet retransmission. We then present how concealment techniques allow the development of algorithms which can achieve robustness with respect to packet losses. Section 1.3 summarizes the main contributions of this thesis, and Section 1.4 presents the thesis outline.

1.1 Packet Networks

Communication networks, such as telephone networks, television broadcast networks, computer networks, or the Internet, provide the means for transferring information between distant users. Communication services supported by a network determine the network design, and differ in how the information is formed and transmitted. Circuit switching networks, and packet switching networks are examples of different network design. In this section, we discuss the transmission of data packets over

packet networks, problems encountered (delay, loss), and techniques applied in response to situations when packets are delayed, or lost.

In *circuit switching* networks, such as telephone networks, a dedicated path (or circuit) is assigned to each transmission. The connection between the source and the destination must be set up before data transmission can take place. Setting up a connection requires signaling and the allocation of necessary resources along the path. This type of transmission is *connection-oriented*, because of the initial setting up of a dedicated line for the flow of data.

In *packet switching* networks, such as computer networks, information in the form of messages is transferred between computers connected to the network. Because of the expectation that the transfer time is to be short, the messages must be of limited size. A unit of information transmitted over packet switching network is called a *packet*. A packet is a block of data of limited length. Longer messages are segmented into several packets. Figure 1.1 represents a packet switching network. A network

Figure 1.1: Packet switching network, [3].

consists of transmission lines interconnected by packet switches spread over a geographic area, providing transport of packets between users. A packet switch performs routing and forwarding functions. In the routing function, the switch selects the path

along which to send data, and updates the routing table. A routing algorithm determines feasible routes. The criterion for a routing algorithm may be the minimization of the number of hops, or the end-to-end delay, or finding a path that offers the greatest bandwidth. The routing table serves as a means to maintain a record of the best routes toward destinations. In the forwarding function, the switch transfers the packet from the input port to the output port as specified by the routing algorithm.

1.1.1 Connectionless and Connection-Oriented Transport

Packet switching may also be connection-oriented. Such *virtual-circuit packet switching* techniques establish a virtual circuit or a fixed path, before transmission of packets takes place. Asynchronous Transfer Mode (ATM) networks represent such an example. In order to set up a virtual circuit, switches along the determined path communicate by exchanging connect-request and connect-confirm signals. Failure to receive a connect-confirm signal from any switch results in the failure of the setup process.

Transmission of packets does not have to be connection-oriented. In *connectionless* transmission systems packets travel between packet switches, where they are forwarded based on routing decisions. Each packet is routed independently. At the destination, messages are recovered or reassembled from packets.

Both the connectionless and connection-oriented services in a packet switching network rely on packet switches. While under both services packets share the same transmission lines, these services exhibit different characteristics. Connection-oriented packet switching requires additional time for setting up virtual circuits, thus increasing the end-to-end delay. However, in the case when extensive data transfer will take place, the overhead associated with setting up a virtual circuit may be acceptable. During the set-up phase connection-oriented services may allocate resources, such as buffers at packet switches and bandwidth assigned to links, to deliver packets within requested performance bounds. However, if a failure happens at a switch, all connections involving that particular switch will fail. Connectionless services cannot

guarantee the correct order of the received packets, because the packets traveling from the same source to the same destination do not necessarily follow the same route.

1.1.2 Delay and Loss in Packet Transmission

Real time transfer of multimedia content imposes strict conditions on the performance of communication networks. In particular, the feasibility of real time transfer of multimedia content is limited by the delay of data packets during transmission. Delays are the result of packet creation, and buffering at packet switches. Best-effort service treats all packets equally. Therefore, a best-effort service may not be suitable for real-time transmission of packets due to excessive delay experienced by such data streams. On the other hand, differentiated and guaranteed service classes provide end-to-end quality of service (QoS) and therefore can potentially support real-time multimedia content transmission over the Internet.

In the differentiated service, each packet is tagged with information that indicates how it should be treated at packet switches. In many data transmission applications a packet must be delivered within a certain time window. To achieve this goal, the guaranteed service class provides resource reservations in the switches for all packets belonging to a given flow.

The delay experienced by packets transversing a network is a function of the network and service characteristics, and the traffic conditions. Applications characteristics in turn determine the maximum allowable tolerance to packet delays. One possible approach to mitigate the effects of delay jitter, a measure of variations in the packet arrival times, is to buffer received packets at the destination. This approach would allow data to be extracted from the receiver buffer at a constant rate thus masking the effects of delay jitter.

Data packets simultaneously arriving at a packet switch are buffered in a queue to wait until the transmission line becomes available. When a packet arrives at a queue with full buffers, the packet is discarded, i.e., lost. Figure 1.2 illustrates the loss

model. Let B_{in} be the aggregate input bandwidth from N_{in} links. Packets arriving

Figure 1.2: Loss model [8].

from the N_{in} links will pass through the router with a finite buffer capacity over the output link with bandwidth B_{out} , such that $B_{out} < B_{in}$. Packet are lost when the transmission link experiences a congestion and the queue associated with that particular link becomes full. Packet loss rate can be defined either with a focus on the router, or with a focus on a particular flow [8]. In the former, the loss rate is the ratio of the number of packets dropped by the router to the total number of packets that can be transmitted over the router's output link over a certain time interval. In the latter definition, only packets that belong to a specific flow are taken into consideration.

In this study we will use the packet loss probability as a system parameter that describes the end-to-end performance by representing the probability that the packet is lost at any point along the transmission path. Dynamic changes in a network environment affect the packet loss probability. Measurements and modeling of packet loss in the Internet have been the subject of numerous studies [6, 7, 8]. The resources that monitor Internet statistics [9, 10] consider packet loss values less than 5% to be unnoticeable.

1.2 Problem Description and Motivation

In this study we investigate the problem of real-time transmission of audio signals over packet networks. Each data packet generated at the source coder contains compressed audio samples representing the audio signal over a short time segment. These packets

enter the network and are forwarded to the destination. At the receiver, the destination, packets are reassembled and the audio signal is recovered from compressed audio samples. If packets are lost during transmission, the receiver may request retransmission of lost packets. However, in real-time applications retransmission may not be a feasible option as retransmission will introduce further delay, and can potentially contribute to network congestion. As an alternative, the receiver may estimate the data contained in the lost packets from the samples contained in previously received packets. Depending on the number of lost packets, signal characteristics and the estimation techniques used, this approach may introduce unacceptable level of audible distortion.

Transmission of multi-channel audio content over packet networks may command high data rates. The total data rate for an uncompressed high fidelity stereo digital audio signal, sampled at a frequency of 44.1 kHz, and uniformly quantized at 16 bits per sample, is 1.4112 Mb/s. The data rate resulting from a 5.1 multi-channel high fidelity signal is approximately 3.6 Mb/s (the aggregate data rate for a 5.1 multi-channel high fidelity signal takes into account the bandwidth of all channels and the sampling frequencies commensurate with channel bandwidths). Furthermore, new source coders employ higher sampling frequencies up to 192 kHz and higher precision quantization up to 24 bits per sample. These emerging coding standards impose even more serious demands on bandwidth, capacity for delivery and storage. Table 1.1 lists values for sampling frequencies and sample precisions for different technologies [21]. A storage media such as a DVD Video disc that contains multimedia content with both video and audio portions (for example, a movie) has a limited capacity. The total data rate, the sum of data rates for video and audio portions, and the content length determines the required storage capacity for a given movie. The maximum total data rate is 9.8 Mb/s [24]. The capacity of a single sided, single layer DVD disc (DVD-5) is 4.7 GB. The uncompressed audio portion of a movie which is an hour and a half (5400 seconds) long and contains 5.1 channels of audio sampled at 48 kHz, and quantized at 16 bits per sample, would require 2.64 GB. The remaining capacity would be used

	Sampling frequency [kHz]	Sample precision [b/sample]	Number of audio channels	Data rate [kb/s]
MPEG-2 BC	32, 44.1, 48	16-24	1-5.1	32-1130
AC-3	32, 44.1, 48	16-24	1-5.1	32-640
MPEG-2 AAC	8-96	16-24	1-48	Up to 576 per channel
DVD Audio	44.1, 48, 88.2, 96, 176.4, 192	16-20-24	1-6	9600

Table 1.1: Digital multi-channel audio signal parameters [21].

for the video portion and other data. As expected the remaining storage capacity on the DVD disc would present a serious challenge to code the video portion at an acceptable quality. The format of audio data on a DVD disc is standardized. Table 1.2 lists the standard audio formats on a DVD Video [23, 24].

	Sampling frequency [kHz]	Sample precision [b/sample]	Number of audio channels	Data rate [kb/s]
PCM	48, 96	16, 24	2-6	Up to 6144
AC-3	48		1-5.1	Up to 448
DTS	48, 96		2-6.1	768, 1536
MP2	48		1-7.1	Up to 912

Table 1.2: Standard audio formats on DVD Video disc [23, 24].

Packet losses are inevitable in a packet based network. Internet as a large and complex set of networks represents an environment where changes are unpredictable, and therefore packet delivery to the destination may not be guaranteed. Traditionally, the system responds by retransmission of the lost packets. The retransmission of the lost packets requires additional time. For real-time applications an additional delay in

the transport of data is not desirable. Concealment techniques represent an alternate approach to mitigate the effects of packet losses. Such techniques do not rely on the retransmission of lost packets. Instead, they estimate any missing data to mask the distortion resulting from lost packets. The decoder may estimate the missing data in several different ways; estimation techniques based on interpolation or replacing the missing data with its conditional expectation, conditional on the data contained in the received packets, represent some of the feasible options.

Multiple Description Coding (MDC) is one of the techniques that offers loss mitigation by employing redundancy. A multiple description encoder creates several, at least two, descriptions of the source signal. The descriptions are sent over different paths in the packet network. The redundancy inherent in multiple descriptions traversing the network over different paths would allow the receiver to decode the data at a reduced quality even if some of the packets representing a description are lost in the network. MDC also utilizes the fact that different users can have varying bandwidth requirements and/or access rates.

MDC introduces redundancies to achieve robustness with respect to packet losses. The robustness of MDC comes at the expense of higher data rates for transmission. However, a properly structured multiple description coder may keep the decoding delay at the receiver within acceptable limits. In Internet applications involving real-time transport of digital multimedia content, such as video conferencing, audio streaming, or Voice over IP, where high delay and packet losses are not tolerable, MDC techniques may prove to be useful.

1.3 Summary of Contributions

The main contributions of this thesis can be summarized as follows.

- **Statistical analysis of multi-channel audio signals:** We conducted an extensive analysis of multi-channel audio signals and determined statistical measures of inter-channel correlations. The results of this analysis can be used as a

basis for various multi-channel audio coding algorithms.

- **Layering strategies for multi-channel audio signals:** We developed layer descriptions based on the perceptual significance and inter-channel correlation of multi-channel audio signals.
- **Extension of priority encoded transmission to multi-channel audio signals:** We applied the priority encoded transmission technique to multi-channel audio signals for transmission over packet networks. We empirically demonstrated the robustness of the resulting multiple description coded data with respect to packet losses.
- **Derivation of simple decoder structures:** We developed simple decoder structures which provide good sound quality and are suitable for real-time applications.

1.4 Thesis Outline

The research presented in this thesis addresses transport of multi-channel audio signals over packet networks such as the Internet. The organization of the thesis is as follows.

Chapter 2 introduces MDC, presents some practical applications of the technique, and provides a comparison of MDC with layered coding. The chapter also reviews the literature on MDC and presents the current state of the art in the use of multiple description techniques. The chapter concludes with the presentation of Priority Encoding Transmission technique.

Chapter 3 introduces multi-channel audio signals, presents the audio database, and analyses inter-channel correlations in the test signals. The chapter delineates the algorithms used in this study, and presents the encoder and decoder structures. As we are not aware of any work which explicitly addresses coding of multi-channel audio signals within an MDC framework, Chapter 3 discusses how this question may

be answered. The algorithm is an extension of a more conventional MDC [31]; it utilizes inter-channel correlations in a multi-channel audio signal to design the multiple descriptions, i.e., layers. This approach is suitable for users with heterogeneous bandwidth requirements as it provides a balance between the fidelity of the reproduced audio at the receiver and the number of received descriptions.

Chapter 4 presents the results obtained by applying the algorithm to multi-channel audio signals in the test database.

Chapter 5 summarizes the main contributions of this thesis, and points out directions for the future research.

Chapter 2

Background

MULTIPLE Description Coding (MDC) is a coding technique which provides a suitable framework for developing robust audio coding algorithms. In Section 2.1 we introduce the MDC. Section 2.2 presents three examples of MDC, whereas Section 2.3 discusses practical applications which demonstrate the versatility of the MDC technique.

We then proceed to describe the compression and transmission of audio samples in the context of packet based processing. Section 2.4 reviews applications of MDC to audio coding. The material presented in Section 2.4 provides the motivation for multiple description coding of multi-channel audio signals which is the focus of this thesis.

Section 2.5 introduces Layered Multiple Description Coding. The chapter concludes with Section 2.6 which describes the Priority Encoding Transmission (PET) based packetization scheme.

2.1 Multiple Description Coding: A Primer

Multiple Description Coding is a coding technique which addresses packet losses as coded data streams are packetized and transmitted over packet networks. MDC techniques achieve this objective by introducing redundancy into coded data streams.

This approach avoids the necessity for retransmitting lost packets and is therefore particularly suitable for low delay applications.

The idea of multiple description coding was born in the 1970s at Bell Laboratories. The motivation was to solve the problem of link outages when transmitting speech signals in the telephone system. The nature of telephone communication requires continuous service, and any standby links that provide continuous service increase cost. Channel splitting of a single call was proposed in 1979. In this approach, the signal is split into two streams and then sent over separate channels. When one transmission channel fails, the signal is still recovered from the data stream received at the other channel albeit at a reduced quality. In September 1979 at the Shannon Theory Workshop the channel splitting problem became the *multiple description problem* when Gersho, Ozarow, Witsenhausen, Wolf, Wyner, and Ziv posed the question: “If an information source is described by two separate descriptions, what are the concurrent limitations on qualities of these descriptions taken separately and jointly?”

Today, MDC is a source coding technique that models an information source as a set of data streams, i.e., *descriptions*. All descriptions will not necessarily reach the receiver, and the source data will be recovered from the arrived descriptions which are a subset of the transmitted descriptions. This approach allows uninterrupted delivery of multimedia content at the price of possible degraded quality and coding efficiency.

The biggest challenge faced by MDC techniques is the design of descriptions. Each description on its own should contain sufficient information to reproduce the source signal with acceptable fidelity even when all other descriptions are lost during transmission. Highest quality and lowest distortion is achieved when all the descriptions are received. There are two extreme designs. In the first approach, all descriptions are identical. Loss of any number of the descriptions would not endanger the signal reconstruction as long as at least one description is received. However, this design is extreme in the redundancy it introduces. In cases when more than one or even all descriptions arrive at the destination, the additional descriptions do not improve the quality of the reconstruction of the source signal. In general, if the descriptions

are too similar, receiving all of them does not yield any significant improvement in recovering the source signal. Such an approach however exhibits high redundancy. The other extreme design would arbitrarily divide the bits representing the source signal into disjoint sets and then create descriptions of those sets. The reconstruction of the source signal at the receiver will be satisfactory only if all the descriptions are successfully received. If a loss happens, the descriptions which arrived to the destination are practically useless. A more preferable approach would be a design that makes use of any subset of the descriptions in the reconstruction process, with a quality commensurable to the number of the received descriptions. The challenge is to design descriptions which are not too similar, separately sufficient and yet capable of good joint description of the source signal with minimal distortion. Another important characteristic of MDC is that in an MD coder all descriptions are of equal importance, i.e., they have the same priority.

Figure 2.1 depicts an MDC scenario with two descriptions of one source sent over two transmission channels which is to be received by one of the three receivers. The

Figure 2.1: Two-transmission channel, three-receiver MD scenario [2].

source signal sequence represented in the figure by $\{x\}$ is coded in the encoder, which creates two descriptions of this signal. They are sent over two different transmission channels. If they both reach the receiving end, their combination in the decoder D_0

allows the reconstruction of the signal $\{x\}$ as $\{\hat{x}_0\}$. If any of the descriptions is lost during the transmission, the description received will be decoded by one of the side decoders, D_1 or D_2 . The decoder's output will be one of the reconstructions $\{\hat{x}_1\}$ or $\{\hat{x}_2\}$, depending on the decoder. In order to achieve the objective of MDC that increasing number of received descriptions refines the quality, $\{\hat{x}_0\}$ should be the best possible reconstruction. If the two descriptions were identical, receiving them both, would not provide any improvement in the quality compared to the case when only one description is received and then used in the reconstruction of the original signal.

In general, if the source was represented by K descriptions, there would be $2^K - 1$ decoders. Different decoders may represent different classes of users, or users operating under different conditions. A good MDC design would tend to minimize the distortion at all decoders. In the context of a multi-channel, multi-decoder MDC scenario as shown in Figure 2.1 the main question is the joint optimization of distortion values at each decoder as a function of channel transmission rates. The transmission channels depicted in Figure 2.1 do not necessarily correspond to physically independent paths, rather they may only represent different descriptions.

2.2 Three Examples of MDC Realizations

This section presents three MDC examples: MD Scalar Quantizers, MD Lattice Vector Quantizers, and MDC with Correlating Transforms. A longer list would include additional examples such as MDC with Frame Expansion, Spatial and Temporal Down-Sampling, Matching Pursuits Algorithms. These examples are by no means exhaustive, however, they are representative of the diverse applications of MDC.

2.2.1 MD Scalar Quantizers

Two quantizers offset from each other: A conventional uniform scalar quantizer maps each input sample value onto a codeword. Suppose that the goal is to generate two descriptions. Splitting the bits of the codewords into two streams in an attempt

to form descriptions will not give satisfactory result if only one of them is received after transmission. In that case, one could only conclude the set of intervals the input values belong to. This information can be used for the estimation of the input sample values if the previous sample is known and correlated to the current one. Another idea produced by the works of Goodman and Quirk in 1970s [5] was combining the most and least significant bits from even- and odd-numbered sample streams for channel splitting.

Consider the scenario when the channels use independent quantizers such that the quantization intervals of the first quantizer are the intervals of the second quantizer shifted by some offset. In this case the codeword generated by the first quantizer defines an interval of possible values for the transmitted data, and the codeword generated by the second quantizer defines another but overlapping interval of possible values. The intersection of the interval includes the transmitted value, which makes the reconstruction more precise. The resulting distortion when both descriptions are received is smaller than the distortion when only one description is received. If one description is lost, the value of the transmitted sample can still be reconstructed with less precision. However, this approach significantly increases the total data rate.

Quantizers with disconnected cells [5]: As in the previous case, the channels use independent quantizers, but now the quantization cells used by each quantizer are unions of disjoint (disconnected) intervals. When both descriptions are received, the quantized input value belongs to the intersection of quantization intervals from both quantizers. For a given aggregate data rate, this approach minimizes the quantization error when both descriptions are received.

Multiple description scalar quantization (MDSQ) [11]: The MDSQ encoder processes incoming samples by first uniformly quantizing them. This operation provides an index to the quantization level for each sample. The encoder then generates two indices from each index and transmits the resulting indices to the decoders in the receiver. These indices define the quantization levels of the separate quantizers as discussed in the previous paragraph.

		Index i_2 (second channel)					
		0	1	2	3	4	5
Index i_1 (first channel)	0	0	1				
	1	2	3	5			
	2		4	6	7		
	3			8	9	11	
	4				10	12	13
	5					14	15

Table 2.1: Mapping of the original quantization level index onto the quantization indices of the MD quantizer for $K = 16$.

For a simple example [5], consider the K -level quantization of input samples. The first quantizer uniformly quantizes each incoming sample and generates the quantization level index $j \in \{0, \dots, K - 1\}$. The encoder then maps each index j to a pair of indices i_1 and i_2 which represent the descriptions generated by the two quantizers. The indices i_1 and i_2 are then sent over the first and second channels, respectively. Table 2.1 illustrates how the index j is mapped onto the i_1 and i_2 indices for $K = 16$. For example, when the indices $i_1 = 2$ and $i_2 = 3$ are received from the first and the second channel, respectively, the decoder determines that the original sample has been quantized to a value corresponding the in quantization level index $j = 7$. If the second channel data i_2 is lost, the decoder approximates the sample value by a value from union of the intervals mapped to indices $j = 4, 6$ and 7 in the original uniform quantizer. As expected, decoding based on i_1 only results in higher distortion.

2.2.2 MD Lattice Vector Quantizers

The MD techniques discussed in the previous section all rely on scalar quantizers to create different descriptions for each source sample. A similar approach applied to vectors of source samples, requires some order established in the set of codewords [12]. The symmetry of a lattice structure suitably offers that order.

In lattice vector quantization, each data vector is mapped to a lattice point. Let Λ be the lattice as shown in Figure 2.2. Let the data vector may be a pair of consecutive source samples as indicated by the point ϵ in the figure. The lattice vector quantizer maps ϵ to the nearest lattice point π represented by its label l_π . In the MDLVQ approach [13, 14, 15], the original fine lattice point π is represented by a unique ordered pair of points (π_1, π_2) which belong to the sublattice Λ' . The sublattice Λ' can be obtained from Λ by scaling and rotation and is therefore geometrically similar to Λ . For further details of this example the reader is referred to [14]. Details of the algorithm for uniquely determining π_1 and π_2 is presented in [13].

Figure 2.2: Multiple Description Lattice Vector Quantization [14].

The indices l'_{π_1} , and l'_{π_2} of the pair of sublattice points are the two descriptions. If only one of the descriptions is received, an estimate of the data vector is determined from the sublattice point associated with that description. If both descriptions are received, the initial lattice point π is decoded uniquely, since the mapping of the label l_π onto (l'_{π_1}, l'_{π_2}) represents a one-to-one relationship. The design of optimal mapping is the main challenge of the MDLVQ technique.

The MDLVQ technique was first presented in [13]. The best performance was always achieved when the decoder receives both descriptions. In [14], the authors modified the technique to address the performance in cases when a description is lost, by changing the shape of the Voronoi cells while controlling a design parameter. The

Voronoi cell assigned to a lattice point is the set of all points for which this lattice point is the nearest of all lattice points. Figure 2.2 shows the Voronoi cell of point π .

In [15], the MDLVQ technique is extended to more than two descriptions by using normalized modified discrete cosine transform coefficients as the K descriptions which represent the source signal. In this case, there is a central lattice Λ , and K sublattices Λ'_i , $i = 0, \dots, K - 1$. The quantization process first maps the source point, i.e., the data vector, onto the nearest point π in lattice Λ . Point π is then uniquely mapped to K points, one in each of the sublattices. This mapping is one-to-one, and its inverse is applied to obtain point π , if all K descriptions arrive at the receiver.

2.2.3 MDC with Correlating Transforms

Transform coding [16] is a technique which applies a linear transform to the source data to create uncorrelated transform coefficients. The transform coefficients are in turn quantized, entropy coded and transmitted. Uncorrelated transform coefficients avoid redundancy, and enhance compression performance.

A modification of this technique introduces controlled correlation among transform coefficients, thus potentially creating conditions for generating multiple descriptions. If only a proper subset of the multiple descriptions, i.e., correlated transform coefficients, arrive at the receiver, then the decoder can utilize the correlation among transform coefficients to estimate the lost transform coefficients.

This approach was first presented in [17]. The authors proposed mapping of two independent, zero-mean, Gaussian random variables X_1 and X_2 into descriptions Y_1 and Y_2 , called correlated transform coefficients as follows.

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \underbrace{\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\text{correlating transform } T} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}. \quad (2.1)$$

The so-called Pairwise Correlating Transform T generates the multiple descriptions Y_1 and Y_2 as the two correlated transform coefficients. By doing so, T also intro-

duces redundancy to the descriptions. In [18], the authors extended the initial work by taking into account the quantization of the source data before the correlating transform.

In [18] the authors addressed a 2-description scenario. In [19] Goyal and Kovačević extended this approach to M -tuple source data which is transformed into $K \leq M$ descriptions. The authors minimized the average distortion while maintaining a constant total bit rate. The difference between constant total bit rate and the minimum average rate is the redundancy. In the $K = 2$ case, the minimal distortion depends only on the probabilities associated with the four different scenarios of receiving the two descriptions, and on the redundancy introduced.

Let us consider a scenario when the source consists of $M > 2$ components and the MD correlating transform procedure generates $K \leq M$ descriptions. First, the M -component source data is quantized with a uniform scalar quantizer, and then transformed into the M -component output vector $\mathbf{y}$. The transform coefficients, i.e., the components of $\mathbf{y}$, are distributed into K descriptions. Of particular interest is the scenario when the coefficients are placed into more than two descriptions. The complexity of design of a single transform usually leads to a cascade structure with several simpler transforms. Figure 2.3 shows an example of a cascade structure where the source $\mathbf{x}$ consists of four components, and the system produces four MD coefficients y_k , $k = 1, \dots, 4$ for transmission over four channels, [19]. After the

Figure 2.3: An example of cascade MD correlating transform [19].

first stage of the transforms represented by T_1 and T_2 , transform coefficients $z_k^{(i,j)}$ carry information on the source components x_i and x_j . After the second stage of transform implemented via T_3 , MD coefficients $\{y_1, y_2, y_3, y_4\}$ carry information on all four source components $\{x_1, x_2, x_3, x_4\}$. If some of the MD coefficients are lost during transmission, the decoder may estimate the missing coefficients from the MD coefficients that were received as all MD coefficients carry partial information on all source components. This cascade structure may be extended to more than two rows of transforms [20]. In such a case, each of the transform coefficients after the k th stage carry information on 2^k source components. To limit the complexity of implementation, a transform block operates on pairs of coefficients. The extension of the system shown in Figure 2.3 to 2^M -input, 2^M -output case would require M transform stages. In [20], this cascade system is implemented by using Hadamard transform based on Hadamard matrices.

2.3 Applications

The analogy between the transmission of multiple descriptions and packet delivery in a network have inspired many researchers to focus on MD techniques. The necessity of dropping packets in order to respond to dynamic changes in a network environment is conventionally handled by (re)transmission protocols. However, if the receiver does not provide feedback, retransmission of packets may also contribute to congestion. Also, the delay resulting from retransmitted packets (which themselves may also be dropped by the network, particularly if network congestion continues) may not be acceptable. Examples of interactive applications which can benefit from MDC include video conferencing, video-on-demand, voice-over-IP and audio streaming services.

Distributed storage, as in peer-driven architecture systems, represents another example which can potentially benefit from MD techniques. Content can be stored at many locations, and for fast access data may be retrieved from the local storage or a peer server. For better quality the content descriptions may be retrieved from several locations, both local and remote, and then combined.

In wireless communication systems with multiple antennas, multiple description source coding can generate bit stream descriptions of the source, which are then independently coded and transmitted using multiple antennas.

The usage of MD coding is justified wherever the possibility of some descriptions loss is present. The underlying philosophy of MD coding implies that different levels of quality of the reconstructed data must be tolerated in the given application, so the precise numerical or textual information do not qualify. However, applications involving images, video, speech, and audio may use MD coding.

2.4 Multiple Description Audio Coding

2.4.1 Perceptual Audio Coders

The design of contemporary audio coders is largely based on psychoacoustic principles, since the final receiver of the coded audio signal is the human ear [21]. Psychoacoustics is the science which focuses on subjective perceptions of sounds, and individual's response to sound stimuli. The objective of an audio coder is to reduce the output data rate without introducing perceptible changes between the original and the reproduced sound. The audio coder strives to achieve this objective using perceptual coding techniques which are based psychoacoustic models and exploit the characteristics of human hearing [25]. Beside perceptual irrelevancies, an audio coder may also exploit statistical redundancies. Figure 2.4 represents a generic perceptual audio coder. The *Time/Frequency Analysis* block converts the audio samples into frequency domain. Its output are blocks of frequency domain coefficients obtained through a frequency transformation such as modified discrete cosine transform. The necessity of the time-to-frequency domain transformation is dictated by the fact that most of the psychoacoustic properties of the auditory system are frequency dependent and therefore can be best determined by analyzing transform coefficients. The *Psychoacoustic Analysis* block computes the masking thresholds which determine maximal level of distortion allowed at each frequency bin. This information determines the

Figure 2.4: Generic perceptual audio coder [25].

bit allocations used by the *Quantizer* block which re-quantizes the transform coefficients. The *Quantizer* may also exploit statistical redundancies using techniques such as differential pulse code modulation. The *Noiseless Coding* block compresses the re-quantized transform coefficients using lossless data reduction techniques such as Huffman coding. And finally, the *Multiplexer* block forms the output bitstream by multiplexing the re-quantized and coded data with accompanying side information needed at the decoder for reconstruction.

Threshold in quiet—also known as the hearing threshold—is a frequency dependent measure which represents the minimum audible sound level at a given frequency [21]. A digital audio coder first determines parts of the audio signal spectrum (if any) which fall below the threshold-in-quiet curve. As audio signal components below the curve will not be perceptible, the coder simply discards them. Audio signal components which are above the threshold curve can be re-quantized with possibly fewer bits as long as the quantization noise level remains below the hearing threshold.

Digital audio coders also rely heavily on the masking properties of audio signals to determine parts of the signal that can be discarded or coarsely re-quantized. There are two types of masking phenomenon: simultaneous or frequency masking and temporal masking. The former happens when a louder sound (the “masker”) occurs concurrently with a soft sound (the “maskee”), rendering the maskee inaudible. The latter happen when the masker and the maskee do not occur at the same time. As

the auditory system requires some time to perceive a stimulus, an audio signal in a particular time frame may temporally mask audio signals in subsequent frames. The louder the sound in the first frame is, the longer will be the duration of its temporal masking.

In the presence of the masker, another stimulus or noise will be audible only if it is not below the masking threshold at the given frequency. Therefore, the coder utilizes the masking threshold information gathered from analyzing a block of audio samples to determine which signal components are masked and the level of masking. The calculated masking thresholds are in turn used by the *Quantizer* block to discard and re-quantize the audio samples in that block.

As perceptual audio coders operate on blocks of audio samples at a time, they make particularly suitable candidates for packet processing techniques such as MDC.

2.4.2 Related Work

While the application of MDC techniques to audio coding seems to be only natural, most of the published work appear to focus on mono or stereo audio signals [26, 14]. More recent work extend MD audio coding to more than two descriptions [15, 29].

The work presented in [26] modifies a standard perceptual audio coder by incorporating an *MD with Correlating Transform* block between the quantizer and noiseless coder. Pairs of quantized coefficients are transformed into two descriptions by the correlating transform. The decoder utilizes an inverse transform in between the noiseless decoder and the inverse quantizer to recover the original coefficient pair. In the case when one of the descriptions is lost, the inverse transform estimates the lost MD correlated transform coefficient from the description available at the decoder. The main contribution of this work has been the development of an appropriate transform which would optimize redundancy and the average distortion for each coefficient pair. The authors also demonstrate the improved robustness of the perceptual audio coder with respect to lost coefficients.

MD with Correlating Transform was the basis for the MDC approach developed in [29]. In this work, the hierarchical (cascade) pairwise correlating transform generates more than two descriptions. Figure 2.5 depicts one such transform block. The transform block operates on modified discrete cosine transform (MDCT) coefficients and introduces distributed correlation among the MD coefficients by processing the input block in a 3-stage transform operation. First, the MDCT coefficients are pairwise

Figure 2.5: Hierarchical correlating transform [29].

grouped. The correlating transform blocks T at the first stage introduce controlled correlation to the MDCT coefficients connected to their respective inputs. Transform blocks are then cross-linked and fed to the inputs of the stage-2 transforms. This process is repeated at the third and final stage such that each of the eight MD coefficients carry some information from each of the eight MDCT coefficients. The indices of the MDCT coefficients which contribute to a particular output of a transform block are listed beside each output.

In [14] the authors use the Multiple Description Vector Quantization technique presented in Section 2.2.2 for MDC of audio signals. In particular, the authors develop an algorithm which represents each source point as a 2-element vector and transmit the indices of an ordered pair of sublattice points—rather than the index of the

nearest lattice point—as the two descriptions. If during transmission any one of the descriptions is lost, the decoder reconstructs the source point from the index of the sublattice point that has arrived at the destination. If both descriptions arrive at the destination, then the decoder uniquely reconstructs the lattice point. Algorithms for the optimal mapping of a lattice point to sublattice points have been the subject of several works [13, 27, 28].

The algorithm presented in [14] pre-filters the audio signal prior to MD lattice vector quantization. The pre-filter normalizes the signal to its masking threshold. The pre-filtering operation also allows the MDC distortion measure to reflect the audible distortion that may be perceptible in the reconstructed signal.

In [15], the authors use an M-channel MD lattice vector quantizer as the basis for an MDC coder with more than two descriptions. A time-frequency analysis block based on MDCT produces normalized transform coefficients, which are then further processed by the MD quantizer. K sublattices and a central lattice allow the MD quantizer to form K descriptions as described in Section 2.2.2.

2.5 Layered Multiple Description Coding

Multiple Description Coding deals with packet losses by generating and transmitting multiple descriptions of the signal. Each description is capable of reproducing the source signal at some level of fidelity and independent of other descriptions. If multiple descriptions are received, the MD decoder combines the individual descriptions to refine the quality of the reproduced signal. With an increasing number of descriptions available, the quality of the reproduced signal steadily improves. In MDC all descriptions are of equal importance.

In Layered Coding (LC) multiple descriptions of different importance, i.e., *layers*, represent the source signal. There is a required base layer and multiple enhancement layers. Increasing the number of enhancement layers available at the decoder enhances the quality of the reproduced signal. Enhancement layers cannot be used on their

own and require the presence of the base layer for reconstructing the source signal. As a consequence of this characteristic, LC is a technique most suitable for creating a coding structure for Internet clients with different bandwidth requirements.

Layered Multiple Description Coding (LMDC) [31] combines the MDC and LC techniques within the same coding framework. In LMDC, each layer is represented by multiple descriptions. All clients are targeted to receive the base layer descriptions, whereas the enhancement layer descriptions are available only to high bandwidth clients. The scenario where the low and high bandwidth clients are isolated, resembles MDC as the quality of reproduction from the base layer improves with the increasing number of the base layer descriptions received. Similarly, the quality of the reproduced signal at the high bandwidth clients depends on the numbers of descriptions from the base and the enhancement layers.

2.6 Priority Encoding Transmission

Albanese et al. introduced the Priority Encoding Transmission (PET) packetization technique in [33] as a method for sending multimedia content over lossy packet based networks. The PET packetization technique first partitions the source signals into layers and assigns a priority level to each layer. Based on their respective priority levels, the samples in each layer are independently encoded into codewords that form packets for transmission.

The PET Packetization Algorithm: Let $\mathbf{S} = \{s[n], n = 1, \dots, N_s\}$ be the N_s -sample long source signal, where $s[n]$ represents the n th sample from of the source signal. Assume that $\mathbf{S}$ is partitioned into K layers $\mathcal{L}_k$, $k = 1, \dots, K$ of decreasing priority and let l_k , $k = 1, \dots, K$ be the number of samples in layer $\mathcal{L}_k$. To simplify the discussion, assume that samples in $\mathbf{S}$ are already sorted in the order of the layer they belong to, i.e., the samples in the highest priority $\mathcal{L}_1$ layer are at the beginning of $\mathbf{S}$, followed by the samples in $\mathcal{L}_2$ and so on. The algorithm reorganizes the samples in each layer into N_k sample long segments such that each N_k -sample long segment

represents a message to be coded. In [31], Reed-Solomon $\mathcal{RS}(N, N_k)$ code has been used, where each segment of length N_k is coded into a codeword of length N . Without loss of generality in [31], we set $N = K$ and $N_k = k$. Once all the segments from all K -layers are coded into N -sample long codewords, the packetization algorithm forms the priority encoded packets by assigning the k th sample from each codeword to the k th-priority packet P_k , $k = 1, \dots, K$.

Figure 2.6 shows an example of PET packetization with $M = 60$, $K = 5$, $l_k = 12$ and $N_k = k$ for $k = 1, \dots, K$, and $N = K = 5$. The black colored samples in the figure extend the total number of samples in layer $\mathcal{L}_5$ to a number which is integer divisible by 5. This extension is necessary for proper segmentation. The dashed rectangles represent the parity symbols added to each segment by the Reed-Solomon coder such that each N_k -sample long segment is mapped onto a N -sample long codeword. The graphical depiction presented in Figure 2.6 clearly shows that the redundancy added to the segments (represented by the parity symbols) in each layer is a function of the priority level of that layer. For example, each segment in the highest priority layer $\mathcal{L}_1$ receives 4 parity symbols whereas segments in the lowest priority layer $\mathcal{L}_5$ do not receive any parity symbols.

The key property of a PET system is that if any N_k packets out of N transmitted are received, then all layers $\mathcal{L}_l$ with $l \leq k$ are completely recovered. In the PET packetization algorithm all packets are of equal significance, which makes the algorithm suitable for designing an MDC system, since packets of equal significance may serve as descriptions. Receiving a larger number of descriptions defined as above would allow the recovery of more layers, thus increasing the quality of the recovered source sequence $\mathbf{S}$.

The key property of the PET system is a direct consequence of the Reed-Solomon codes. An $\mathcal{RS}(N, N_k)$ code encodes a message of length N_k into codeword of N symbols including $N - N_k$ parity symbols. The main feature of the Reed-Solomon codes is that the code is able to correct any $N - N_k$ erasures, with no constraint that the erasures must be among the parity symbols. In other words, as long as at least

N_k symbols out of the N symbols in the codeword are not erased, it is possible to decode the N_k -symbol long coded message. This property of the Reed-Solomon codes imply that all symbols in a codeword are equally important. Appendix A provides a brief description of Reed-Solomon codes.

Chapter 3

Layered Multiple Description Coding of Multi-Channel Audio Signals

IN THIS chapter we present the design of a layered multiple description coder for multi-channel audio signals. We first introduce multi-channel audio signals, present their characteristics and then proceed to develop the layered MDC algorithm that exploits these characteristics.

3.1 Multi-Channel Audio Signals

Technical innovations introduced since the 1990's such as DVD, HDTV broadcasting, and the emerging trend that movie watchers are increasingly preferring home movie theaters over cinema halls have motivated the advancement of multi-channel audio. Audio channels correspond to specific representations of the audio signal that allow enveloping impression and spatial perception of sound. A multi-channel audio signal played back through a set of speakers in a correct spatial arrangement, can provide a listener with a sound experience which is significantly more realistic than it is possible only with monophonic or stereophonic sound.

This research and in particular the signal processing algorithms developed in this Chapter are based on multi-channel audio in 5.1 channel configuration. This configuration was originally developed for movie applications [36] with the objective to enhance film soundtracks through multi-channel audio such that the reproduced sound environment would resemble the sound experienced on a movie stage. The MPEG-2 Audio standard has been among the first audio coding standards supporting the 5.1 channel configuration. The Dolby AC-3 standard also known as Dolby Digital, is another audio coding standard which supports multi-channel audio for several applications including North American HDTV and DVD Video.

In the 5.1 configuration [21], the multi-channel audio signal consists of five 20 kHz-bandwidth channels and a low frequency enhancement channel (LFE), which covers frequencies below 200 Hz.¹ Often, the notation 3/2/.1 is used to denote the arrangement with three speakers placed in front of the listener, and two in the back. Figure 3.1 depicts the spatial speaker arrangement corresponding to the 5.1 configuration. International Telecommunication Union - Radiocommunication Sector (ITU-R) recommends the locations of the speakers and their placement above ear level to maximize the surround sound effect. Specifically, the five full-bandwidth speakers are placed around the listener who is at the center of a circle determined by their positions. The center channel (**C**) is directly in front of the listener, whereas the left (**L**) and right (**R**) channels are at $\pm 30^\circ$ angles from the listener-center axis. Left-surround (**Ls**) and right-surround (**Rs**) channels are at $\pm 100^\circ$ – 120° from the same axis. The precise location of the LFE speaker is not specified, but it is often placed in the front. In 5.1 music applications such as DVD Audio, the main channels may include low frequency components. Under such circumstances the low frequency content may potentially overload the main speakers and decrease the overall dynamic range of the signals played back through the main channels. When present, the LFE

¹When the LFE channel was first introduced in the early 1990's, it covered the frequency band from 20 to 120 Hz [22]. Although the bandwidth of the LFE channel corresponds to only 0.005 of the 20 kHz audio bandwidth, the system with 5 full bandwidth channels and the LFE channel was nevertheless named "5.1".

Figure 3.1: 5.1 multi-channel audio configuration.

channel preserves the compatibility with home video format.

A multi-channel audio signal may consist of more than six channels. For example, the 7.1 configuration includes back-left and back-right channels in addition to all the channels contained in the 5.1 configuration. The 7.1 configuration places the back-left and back-right speakers at angles between 135° and 150° from the listener-center axis. The role of these two channels is to enhance the sense of ambient sound from the direction behind the listener. These additional sound channels effectively address some inherent limitations of the human auditory system which at times fails to correctly locate a sound coming from behind.

The multimedia content delivered as multi-channel audio is streamed as a file consisting of interleaved sound samples from individual channels. Table 3.1 presents the default channel layout order in an audio file [35]. In the case when some of the channels listed in Table 3.1 are not part of a given configuration, the order is still preserved among the present channels.

In this work, we will focus on audio signals in 5.1 configuration which will include the channel signals in the following order: front left, front right, front center, low

Index	Channel
1	Front Left
2	Front Right
3	Front Center
4	Low Frequency
5	Back Left
6	Back Right
7	Front Left of Center
8	Front Right of Center
9	Back Center
10	Side Left
11	Side Right
12	Top Center
13	Top Front Left
14	Top Front Center
15	Top Front Right
16	Top Back Left
17	Top Back Center
18	Top Back Right

Table 3.1: Default channel ordering in a multi-channel audio file.

frequency, left surround and right surround. We will use respectively the labels **L**, **R**, **C**, **W**, **Ls** and **Rs** to refer to these audio channels.

3.2 Inter-Channel Correlations

In order to measure the inter-channel correlation characteristics of multi-channel audio signals we extracted 88 audio files from ten DVD Video disks and one DVD Audio disk and formed a database of audio files in 5.1 configuration. The sampling rate of all the audio signals was 48 kHz, and the duration of the great majority (total of 78)

of the files ranged from 6 to 10 seconds, while the remaining 10 files were 105 seconds long. All files in the database are in full 5.1 configuration with all six audio channels being present. The only exception are files extracted from a DVD disk containing only 5 main channels with no LFE present. Appendix C provides the full discographic information for the source DVD Video and DVD Audio disks.

The development of any audio coding algorithm requires an understanding of the inherent characteristics of the underlying audio signal. The objective of this work is to develop a coding algorithm which is robust with respect to packet losses. Towards this goal we can design a multiple description coding system which exploits the redundancies existing in a multi-channel audio signal. To explore such redundancies we investigate the inter-channel correlations as a measure of similarity. In particular, for each file in the database we extracted the six audio channel signals **L**, **R**, **C**, **W**, **Ls** and **Rs**, and calculated the fifteen cross-correlation values for pairs of audio channel signals. Figure 3.2 presents statistical measures of the absolute values of cross-correlation coefficients from all sound files in the audio database. Figure 3.2 also shows for every pair of audio channel signals: (i) the mean value of the cross-correlation coefficient obtained by averaging over all files in the database, (ii) the 95% confidence interval (CI) for the mean value of the correlation coefficient, and (iii) the minimum and maximum values for the cross-correlation coefficient. The CI is an estimated interval which is likely to include the unknown mean value of a sample correlation coefficient population, with a probability of 0.95.

Analysis of the cross-correlation values indicates that the audio channel signals can be separated into three disjoint groups with Group 1 consisting of the front channel signals **{L, R, C}**, Group 2 consisting of the two surround channels **{Ls, Rs}** and Group 3 consisting of the low frequency signal **{W}**. We observe that signals within each group are mutually correlated whereas signals from different groups exhibit relatively low levels of mutual correlation. The results of the cross-correlation analysis and the corresponding natural groupings of the audio channel signals guide the development of the layered MDC algorithm that we introduce in the next section.

Figure 3.2: Statistical measures of the absolute values of cross-correlation coefficients corresponding to pairs of audio channel signals extracted from the audio database.

3.3 Layer Descriptions

Transformation of channel signals into linear combinations would provide a suitable framework for generating layered descriptions of multi-channel audio signals. Each linear combination of channel signals would contain elements from all channels to which the transform has been applied. Therefore, each linear combination, a *layer*, would be sufficient on its own to reproduce the multi-channel audio signal at some level of fidelity. Alternatively, by utilizing the correlated nature of channel signals, we can estimate the individual channel signals from each linear combination with varying degrees of success.

As a result of the cross-correlation analysis shown in Figure 3.2 we decided to formulate the low frequency channel signal **W** into a separate layer. This decision was motivated by the fact that the low frequency channel is not present in all audio files, and that even when it is present, it exhibits very low levels of cross-correlation with other channel signals.

Our initial efforts were directed to formulate layers based on optimal linear combinations of channel signals. The optimization criteria was the minimization of a distortion measure defined as the mean-square error between the original and reconstructed signal. The number of free parameters involved in the optimization process limited the analysis to a suboptimal solution.

Let $\mathbf{x}$ be a vector of 5 channel signals represented as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{L} \\ \mathbf{R} \\ \mathbf{C} \\ \mathbf{Ls} \\ \mathbf{Rs} \end{bmatrix}. \quad (3.1)$$

Let $\mathbf{y}$ be the vector representing the layered descriptions derived from the channel signal vector $\mathbf{x}$ via the linear transformation matrix $\mathbf{G}$:

$$\mathbf{y} = \mathbf{G} \mathbf{x}. \quad (3.2)$$

$\mathbf{G}$ is a 5×5 transformation matrix with real-valued elements. Layered descriptions of multi-channel audio signals represented by the elements of the vector $\mathbf{y}$, are in turn transmitted over physical channels, i.e., paths in a network, to the destination. Some of the transmitted layer descriptions may not reach to their destinations; therefore, let $\mathbf{z}$ be the vector of received layers defined as:

$$\mathbf{z} = \mathbf{C} \mathbf{y}, \quad (3.3)$$

where $\mathbf{C}$ is the 5×5 diagonal matrix with diagonal elements that are either “0” or “1” representing the effects of transmitting layers over the network. The presence of a “1” (“0”) on the main diagonal indicates that the corresponding layer has been received (lost). Due to the hierarchy of layer priorities, the MDC property ensures that if the highest priority level among all the received layers is k , then all layers at a priority level $i \leq k$ are also present. This property implies that if $c_{kk} = 1$ for some $1 \leq k \leq 5$, then $c_{ii} = 1$ for $1 \leq i \leq k$, where the notation v_{mn} represents the element of the matrix $\mathbf{V}$ at position (m, n) .

The receiver computes a linear estimate of the channel signal vector $\mathbf{x}$ from received layers $\mathbf{z}$ as:

$$\hat{\mathbf{x}} = \mathbf{H} \mathbf{z}. \quad (3.4)$$

Figure 3.3 illustrates the multi-channel signal transmission and estimation process.

Figure 3.3: Transmission of multi-channel signal $\mathbf{x}$.

We can also represent the estimated multi-channel signal as:

$$\hat{\mathbf{x}} = \mathbf{A} \mathbf{x}, \quad (3.5)$$

where $\mathbf{A}$ is the matrix:

$$\mathbf{A} = \mathbf{H} \mathbf{C} \mathbf{G}. \quad (3.6)$$

We express the distortion resulting from transmitting and estimating the multi-channel signal $\mathbf{x}$ using the model depicted in Figure 3.3 as follows.

$$\begin{aligned}
\mathcal{D} &= \mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}\|^2] \\
&= \mathbb{E}[\|\mathbf{x} - \mathbf{A}\mathbf{x}\|^2] \\
&= \mathbb{E}[\|(\mathbf{I} - \mathbf{A})\mathbf{x}\|^2] \\
&= \mathbb{E}[\mathbf{x}^T(\mathbf{I} - \mathbf{A})^T(\mathbf{I} - \mathbf{A})\mathbf{x}] \\
&= \text{trace}[(\mathbf{I} - \mathbf{A})^T(\mathbf{I} - \mathbf{A})\mathbf{R}_{xx}],
\end{aligned} \tag{3.7}$$

where the “trace” operator applied to an $N \times N$ square matrix $\mathbf{V}$ with elements v_{ij} is:

$$\text{trace}[\mathbf{V}] = \sum_{i=1}^N v_{ii}, \tag{3.8}$$

In Equation (3.7) $\mathbf{I}$ is the identity matrix, and $\mathbf{R}_{xx}$ is the autocorrelation matrix:

$$\mathbf{R}_{xx} = \mathbb{E}[\mathbf{x}\mathbf{x}^T], \tag{3.9}$$

where $\mathbb{E}[\cdot]$ is the expected value operator. We use the notation ρ_{ij} to refer to the elements of the autocorrelation matrix $\mathbf{R}_{xx}$. Using the linearity property of the trace function, we express $\mathcal{D}$ as:

$$\begin{aligned}
\mathcal{D} &= \underbrace{\text{trace}[\mathbf{R}_{xx}]}_{t_1} - \underbrace{\text{trace}[(\mathbf{A} + \mathbf{A}^T)\mathbf{R}_{xx}]}_{t_2} + \underbrace{\text{trace}[\mathbf{A}^T\mathbf{A}\mathbf{R}_{xx}]}_{t_3} \\
&= t_1 - t_2 + t_3.
\end{aligned} \tag{3.10}$$

The first term t_1 equals to the sum of autocorrelation coefficients of the individual channel signals in $\mathbf{x}$:

$$t_1 = \sum_{i=1}^5 \rho_{ii}. \tag{3.11}$$

Since the trace of the product of two symmetric matrices equals to the sum of all products of the elements of the two matrices at the same positions, the terms t_2 and t_3 can also be expressed as:

$$t_2 = 2 \sum_{i=1}^5 \sum_{j=1}^5 \rho_{ij} a_{ij}, \tag{3.12}$$

and

$$t_3 = \sum_{i=1}^5 \sum_{j=1}^5 \rho_{ij} \langle \mathbf{A}_{:,i}, \mathbf{A}_{:,j} \rangle, \quad (3.13)$$

where $\mathbf{A}_{:,i}$ denotes i th column of $\mathbf{A}$, and $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^T \mathbf{u}$ represents the inner product of the column vectors $\mathbf{u}$ and $\mathbf{v}$. Substituting the expressions given in Equations (3.11–3.13) into Equation (3.10), we obtain:

$$\mathcal{D} = \sum_{i=1}^5 \rho_{ii} + \sum_{i=1}^5 \sum_{j=1}^5 \rho_{ij} (\langle \mathbf{A}_{:,i}, \mathbf{A}_{:,j} \rangle - 2 a_{ij}). \quad (3.14)$$

From the definition given in Equation (3.6), we can express the elements a_{ij} of matrix $\mathbf{A}$ as:

$$a_{ij} = \sum_{k=1}^5 h_{ik} g_{kj} c_{kk}, \quad (3.15)$$

where $h_{..}$, $g_{..}$, and $c_{..}$ are the elements of the matrices $\mathbf{H}$, $\mathbf{G}$ and $\mathbf{C}$, respectively.

In this study we use a five layer encoding structure. This structure contains the minimum number of layers which would allow a complete recovery of the five multi-channel signals $\mathbf{L}$, $\mathbf{R}$, $\mathbf{C}$, $\mathbf{Ls}$, and $\mathbf{Rs}$ in case when all layers descriptions arrive at the destination and are available for decoding. Increasing the number of layers above 5 would introduce further redundancy and enhance the robustness of the system with respect the packet losses. However, using an encoding structure where the number of layers are greater than the multi-channel signals would also increase the output data rate. Let $\mathbf{y}_i$ represent the i th layer description. The index i also represents the corresponding priority level, such that $\mathbf{y}_1$ is the highest priority layer and $\mathbf{y}_5$ is the lowest priority layer. The definition of the layer descriptions in terms of the multi-channel signal components are as follows.

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \mathbf{y}_4 \\ \mathbf{y}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{L} + \mathbf{R} + \mathbf{C} \\ \mathbf{L} + \mathbf{R} \\ \mathbf{L} - \mathbf{R} \\ \mathbf{Ls} + \mathbf{Rs} \\ \mathbf{Ls} - \mathbf{Rs} \end{bmatrix}. \quad (3.16)$$

The layer descriptions given in Equation (3.16) result in the encoder transformation matrix:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}. \quad (3.17)$$

The above choice of $\mathbf{G}$ corresponds to a suboptimal transformation matrix as its structure does not represent the outcome of a joint optimization process which would see the minimization of the distortion measure $\mathcal{D}$ given in Equation (3.7) with respect to $\mathbf{G}$, $\mathbf{C}$ and $\mathbf{H}$ matrices. Instead, the structure of $\mathbf{G}$ reflects the subjective assessment of the significance of multi-channel audio signals, and the observed inter-channel correlations as presented in Section 3.2. In particular, layer $\mathbf{y}_1$ contains the sum of the $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{C}$ channel signals. As the highest priority layer, $\mathbf{y}_1$ will be the only description available to all network clients if only one layer is received. The information contained in the front center channel signal $\mathbf{C}$ is similar to that found in the front left ($\mathbf{L}$) and front right ($\mathbf{R}$) channel signals. However, the $\mathbf{C}$ signal also contains unique information which is essential for good sound quality. For example, if the soundtrack includes a conversation, the $\mathbf{L}$ and $\mathbf{R}$ signals may contain individual or mixed voices of the two parties taking part in the conversation, whereas the $\mathbf{C}$ signal may contain additional audio information such as ambient sound, background sound or music. These observations resulted in the decision to include the $\mathbf{C}$ signal together with the $\mathbf{L}$ and $\mathbf{R}$ signals in the highest priority layer $\mathbf{y}_1$. Thus, $\mathbf{y}_1$ contains all front channel signals which together represent perceptually the most significant portion of a multi-channel audio signal. Furthermore, the inclusion of the $\mathbf{L}$ and $\mathbf{R}$ signals in $\mathbf{y}_1$ creates the foundation of a hierarchical layered coding framework as both $\mathbf{L}$ and $\mathbf{R}$ signals are highly correlated with each other and the $\mathbf{C}$ signal.

Layers described by $\mathbf{y}_2$ and $\mathbf{y}_3$ contain respectively the sum and difference of the $\mathbf{L}$ and $\mathbf{R}$ channel signals. The structure of $\mathbf{y}_2$ and $\mathbf{y}_3$ is therefore similar to the signal formats used in MPEG intensity coding option for stereo audio signals. The

MPEG intensity coding algorithm first generates and then codes the mono ($\mathbf{L}+\mathbf{R}$) and difference ($\mathbf{L}-\mathbf{R}$) signals. A key characteristic of the intensity coding algorithm is the emphasis it places on the mono signal. Consequently, the proposed layer descriptions described in Equation (3.17) represents an extension of intensity coding for stereo signals to multi-channel audio signals where the combined front channel signal ($\mathbf{C} + \mathbf{L} + \mathbf{R}$) in $\mathbf{y}_1$ plays the same role as the mono signal.

If the highest priority layer $\mathbf{y}_1$ is the only description available at the receiver, the decoder estimates all channel signals by multiplying $\mathbf{y}_1$ with scaling factors h_{ij} to obtain $\hat{\mathbf{x}}$. The scaling factors, i.e., the elements of the decoder matrix $\mathbf{H}$, are obtained by minimizing $\mathcal{D}$ with respect to $\mathbf{H}$ when $\mathbf{C} = \text{diag}[1, 0, 0, 0, 0]$. Under this scenario the optimization process needs to determine only $\mathbf{H}_{:,1}$.

Consider the case when the decoder receives only the highest two priority layers $\mathbf{y}_1$ and $\mathbf{y}_2$. This case corresponds to $\mathbf{C} = \text{diag}[1, 1, 0, 0, 0]$, and allows full recovery of the $\mathbf{C}$ channel signal. The corresponding estimate $\hat{\mathbf{x}}$ of the multi-channel audio signal can be expressed as:

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{H} \mathbf{C} \mathbf{G} \mathbf{x} \\ &= \begin{bmatrix} h_{11} & h_{12} & * & * & * \\ h_{21} & h_{22} & * & * & * \\ 1 & -1 & * & * & * \\ h_{41} & h_{42} & * & * & * \\ h_{51} & h_{52} & * & * & * \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{L} \\ \mathbf{R} \\ \mathbf{C} \\ \mathbf{Ls} \\ \mathbf{Rs} \end{bmatrix}, \end{aligned} \quad (3.18)$$

In the above equation “*” entries in $\mathbf{H}$ represent the “do not care” values. Therefore, the optimization process determines the scaling coefficients h_{ij} , $i \in \{1, 2, 4, 5\}$ and $j \in \{1, 2\}$, as the elements of $\mathbf{H}$ which would minimize $\mathcal{D}$.

With the first three layers received, it is possible to fully recover the $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{C}$ channel signals. The decoder once again estimates the remaining channels from the received layers by minimizing $\mathcal{D}$ with respect to $\mathbf{H}$ using the channel matrix

$\mathbf{C} = \text{diag}[1, 1, 1, 0, 0]$. In this scenario the minimization of $\mathcal{D}$ with respect to $\mathbf{H}$ would involve only a subset of the elements in the first three columns of $\mathbf{H}$.

If the first four layers are received, we have $\mathbf{C} = \text{diag}[1, 1, 1, 1, 0]$. As in the previous case, the decoder fully recovers the $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{C}$ channel signals and estimates the back channel signals $\mathbf{L}_s$ and $\mathbf{R}_s$ from the received layers.

When all layers are received, the decoder recovers all five channel signals. This case corresponds to $\mathbf{C} = \text{diag}[1, 1, 1, 1, 1]$ with the decoder achieving minimal distortion for $\mathbf{H} = \mathbf{G}^{-1}$.

Appendix B lists the values of the decoder coefficients h_{ij} determined from the minimization of the distortion $\mathcal{D}$ measure under all five scenarios.

3.4 Encoder

Section 3.3 presented a framework of how individual channel signals can be transformed into layered descriptions of multi-channel audio signals which would be suitable for transmission over packet networks. In this section we present the encoder which processes the channel signals by performing the following sequence of signal processing operations: segmentation, layering, normalization, audio coding and packetization. Figure 3.4 presents the encoder structure.

Figure 3.4: Encoder block diagram.

3.4.1 Segmentation

The layered multiple description coding is a block process. The segmentation stage separates the channel signals into blocks of M samples. Let $\mathbf{L}_n$, $\mathbf{R}_n$, $\mathbf{C}_n$, $\mathbf{Ls}_n$, $\mathbf{Rs}_n$, and $\mathbf{W}_n$ be the n th block of the channel signals $\mathbf{L}$, $\mathbf{R}$, $\mathbf{C}$, $\mathbf{Ls}$, $\mathbf{Rs}$ and $\mathbf{W}$, respectively. In particular, the n th block of a channel signal includes samples from that channel signal with indices $\{(n-1)M+1, \dots, nM\}$. For example, $\mathbf{L}_n$ is defined as:

$$\mathbf{L}_n = \{ \mathbf{L}[(n-1)M+1], \dots, \mathbf{L}[nM] \}, \quad (3.19)$$

where $n = 1, 2, \dots$. We use the notation $\mathbf{u}[k]$ is the k th element of the vector/sequence $\mathbf{U}$. Let $\mathbf{x}_n$ be the n th block of multi-channel signal vector $\mathbf{x}$.

The distortion measure $\mathcal{D}$ given in Equation (3.7) is a function of the autocorrelation matrix $\mathbf{R}_{xx}$. Consequently, the decoder matrix $\mathbf{H}$, whose elements are optimized to minimize the $\mathcal{D}$ resulting from estimating channel signals, is also a function of the autocorrelation function. The segmentation stage calculates the autocorrelation matrix for each block of channel signals. Let $\mathbf{R}_{xx,n}$ be the autocorrelation matrix for the n th block of multi-channel signal defined as:

$$\mathbf{R}_{xx,n} = \mathbb{E}[\mathbf{x}_n \mathbf{x}_n^T], \quad (3.20)$$

The block size M is one of the system parameters. Chapter 4 discusses how M is determined and investigates the impact of M on the system performance.

3.4.2 Layering

Section 3.3 presented how layer descriptions are generated from multi-channel audio signals. Segmentation operates on the layer descriptions given in Equation (3.16) and generates the n th layered signal block $\mathbf{y}_n = [\mathbf{y}_{1,n}^T \mathbf{y}_{2,n}^T \mathbf{y}_{3,n}^T \mathbf{y}_{4,n}^T \mathbf{y}_{5,n}^T]^T$ by processing the n th block of the multi-channel signal vector $\mathbf{x}_n$.

Section 3.3 also presented the rationale why the low frequency channel signal $\mathbf{W}$ is not part of the multi-channel audio signal vector $\mathbf{x}$, which drives the layering process. Therefore, the segmented low frequency channel signal $\mathbf{W}_n$, when present, is transmitted as a separate layer independent from the others.

3.4.3 Normalization

Creation of layer descriptions involves the computation of linear combinations of channel signals. As a result, sample magnitudes in the layer descriptions may exceed the maximum values that can be represented within the allowable word length—typical word length is 16 bits/sample. Therefore, the encoder calculates the maximum sample magnitude across all layers and normalizes all samples.

Let α_n be the maximum sample magnitude determined from the n th layer signal blocks $\mathbf{y}_{1,n}, \dots, \mathbf{y}_{5,n}$:

$$\alpha_n = \max\{ |\mathbf{y}_{1,n}[1]|, \dots, |\mathbf{y}_{1,n}[M]|, |\mathbf{y}_{2,n}[1]|, \dots, |\mathbf{y}_{5,n}[M]| \} \quad (3.21)$$

Once the parameter α_n is determined the encoder normalizes the layer blocks and generates the normalized layer blocks:

$$\mathcal{L}_{i,n} = \frac{1}{\alpha_n} \mathbf{y}_{i,n}, \quad i = 1, \dots, 5. \quad (3.22)$$

Alternatively, one can determine the maximum sample magnitude for each layer separately. Such an approach may preserve a wider dynamic range of sample amplitudes as each layer is normalized by its respective maximum. However, in our studies we were not able to observe any discernible improvement in the resulting sound quality relative to the case when a single scaling factor was used. Therefore, we decided to use α_n defined in Equation (3.21) as the only scaling factor to be used for all layer signals. Using a single scaling factor also has the advantage of minimizing the amount of side-information that needs to be transmitted with each block of samples.

As the encoder treats the low frequency channel as a stand alone layer, samples in the n th low frequency channel block $\mathbf{W}_n$ do not require any normalization.

3.4.4 Audio Coding

The audio coding block processes the normalized layer blocks and generates the corresponding compressed MPEG-1 Layer III (mp3) bitstreams $\mathcal{L}_{i,n}^c$, $i = 1, \dots, 5$ and $\mathbf{W}_n^c$.

The mp3 bitstream is a sequence of frames each consisting of two parts. The first part contains a header, cyclic redundancy check (CRC) bits and side information. The second part, the main data portion, is the bitstream corresponding to compressed audio samples. A pointer, `main_data_begin`, indicates the beginning of the audio data in the current frame. The pointer may point backwards to the area of the previous frame for optimal usage of space, as shown in Figure 3.5.

Figure 3.5: An example of mp3 frames [21].

The proposed algorithm does not partition the mp3 bitstream, because if some frames which are significant for the neighboring frames in the bitstream, were lost during transmission, the received neighboring frames would be useless. The partitioning of the bitstream could place the `main_data_begin` pointer and the actual beginning of a frame's audio data into separate partitions. Because of this, the segmentation stage precedes the audio compression, and each segment is compressed separately.

The audio compression stage reduces the number of bits used to represent the M -sample long blocks. Compressed data in each block from all layers is of equal length and is represented by N words where each word is 16 bits long.

3.4.5 Packetization

Packetization is the final signal processing operation implemented by the encoder. In this stage, the encoder applies the PET packetization scheme introduced in Section 2.6 by combining the N -word long blocks² from the compressed layer signals into five distinct descriptions. Figure 3.6 depicts the packetization process for $N \gg 5$.

We recall that the layer index also represents the priority level assigned to individual layers such that $\mathcal{L}_{1,n}^c$ is the highest priority layer and $\mathcal{L}_{5,n}^c$ is the lowest priority layer. As per PET algorithm, the packetization process partitions each $\mathcal{L}_{i,n}^c$ into i words. If N is not integer divisible by i , then the corresponding block is padded with zeros to the length $\lceil N/i \rceil$ where $\lceil \beta \rceil$ is the smallest integer greater than or equal to β . Each i -word long portion in the i th layer represents a message to be encoded. A Reed-Solomon encoder (see Appendix A) codes each i -word long message into a 5-word long codeword using using $\mathcal{RS}(i, 5)$ code such that each codeword would contains an i -word long message and $5 - i$ parity symbols. Thus, codewords generated by encoding messages from higher priority layers include more parity symbols.

In Figure 3.6, $m(i, k)$ denotes the k th word of zero padded compressed layer block $\mathcal{L}_{i,n}^c$ where $1 \leq k \leq i \lceil N/i \rceil$. The notation $p(i, q, r)$ represents the r th parity symbol in the q th codeword associated with the $\mathcal{L}_{i,n}^c$ block where $1 \leq r \leq 5 - i$ and $1 \leq q \leq \lceil N/i \rceil$.

Using the above described layered coding and packetization processes we can also determine the encoder output data rate as a function of N . We first observe that transmission of six compressed audio channels would require a total of $6N$ words if

²In a strict sense, N is greater than the number of words in each compressed description. The side information consisting of α_n and $\mathbf{R}_{xx,n}$ is incorporated into the highest priority description $\mathcal{L}_{1,n}^c$ prior to the packetization process. The 15 unique elements in $\mathbf{R}_{xx,n}$ together with the scaling factor α_n amount to a total side information of 16 words per block. Therefore, including the side information the length of $\mathcal{L}_{1,n}^c$ will be $N + 16$ words. The length of $\mathcal{L}_{1,n}^c$ in turn determines how data from lower priority descriptions are partitioned into codewords. In this report, with some abuse of notation, we will continue to use N to represent the length of the compressed descriptions.

Figure 3.6: The packetization process.

no layered coding is used. The layered coding algorithm extends each $\mathcal{L}_{i,n}^c$ to a length of $\lceil N/i \rceil$ by zero-padding and then maps each i -word long in a 5-word long codeword. Furthermore, the compressed low-frequency channel $\mathbf{W}_n^c$ continues to be represented in N -words as it does partake in the layered structure. Therefore, the total output data rate (in words per M -sample input block) of the encoder equals

$$5 \sum_{i=1}^5 \lceil \frac{N}{i} \rceil + N. \quad (3.23)$$

With increasing N , the relative increase in the encoder output data rate approaches a constant value:

$$\lim_{N \rightarrow \infty} \frac{1}{6N} \left(5 \sum_{i=1}^5 \lceil \frac{N}{i} \rceil + N \right) = \frac{5}{6} \left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} \right) + \frac{1}{6} = 2.069\bar{4}. \quad (3.24)$$

Consequently, the layered coding algorithm results in approximately 2-fold increase in the encoder output data rate.

Descriptions

Finally, the encoder forms descriptions by building packets P_i , $i = 1, \dots, 5$ such that P_i consists of i th words from all the codewords generated from processing the samples in $\{\mathcal{L}_{i,n}^c, i = 1, \dots, 5\}$. We also observe that all packets contain $\sum_{i=1}^5 \lceil N/i \rceil$ words after taking into account any necessary zero padding on layer signals. In the following sections we will use the terms “packet” and “description” interchangeably.

The normalized low frequency layer blocks $\mathbf{W}_n^c$ shown in Figure 3.4, are not part of the PET packetization process. Instead, the encoder forms the N -word long packet P_W directly from the samples in $\mathbf{W}_n^c$.

Packets $P_1, \dots, P_5$ represent equally important **descriptions** of the layer signals. If any $5 - l$ packets are lost during transmission, the received l packets allow the recovery of all the message words in layer blocks $\mathcal{L}_{i,n}^c$, where $1 \leq i \leq l$, and $1 \leq l \leq 5$.

3.5 Decoder

Once the encoder generates the packets, they are transmitted over the network. Let $\mathcal{P} = \{P_1, \dots, P_5\}$ be the set of transmitted packets. The channel $\mathcal{C}$ module in Figure 3.3 represents a communication network with a multitude of transmission links such that packets may follow different paths towards the destination. During the transmission some of the packets may be lost. Let $\mathcal{Q} \subseteq \mathcal{P}$ be the set of packets that arrive at the destination. The decoder has the responsibility to recover the message words from $\mathcal{Q}$. As a direct result of the Reed-Solomon coding used to generate the codewords, the decoder can recover message words from all layers at priority levels $1 \leq i \leq l$ where l is the number of received packets in $\mathcal{Q}$. The recovered layer signals are then in turn used to estimate the multi-channel audio signals. Figure 3.7 depicts the case when $\mathcal{Q} = \{P_2, P_3\}$. The scenario illustrated in the figure also assumes that the decoder receives P_W representing the low frequency channel signal W_n^c .

Figure 3.7: Decoder structure for the case $\mathcal{Q} = \{P_2, P_3\}$.

3.5.1 Unpacketization

If l is the number of packets in $\mathcal{Q}$ then the decoder can recover message words from all layers at priority levels $1 \leq i \leq l$. Using the received packets, the decoder first forms a codeword structure similar to the one given in Figure 3.6. This structure will have empty slots corresponding to the the lost packets. These empty/lost slots represent erasures in the Reed-Solomon codewords.

Example: Figure 3.8 provides an example of a recovered codeword data structure for the case $Q = \{P_2, P_3\}$. In this example the k th codeword generated from $m(1,k)$

		P_3	P_2		
		↓	↓		
LOST	LOST	$p(1,1,2)$	$p(1,1,1)$	LOST	
LOST	LOST	$p(1,2,2)$	$p(1,2,1)$	LOST	
⋮	⋮	⋮	⋮	⋮	
LOST	LOST	$p(1,N,2)$	$p(1,N,1)$	LOST	
LOST	LOST	$p(2,1,1)$	$m(2,2)$	LOST	
LOST	LOST	$p(2,2,1)$	$m(2,4)$	LOST	
⋮	⋮	⋮	⋮	⋮	
LOST	LOST	$p(2, \lfloor \frac{N}{2} \rfloor, 1)$	$m(2, 2 \lfloor \frac{N}{2} \rfloor)$	LOST	
LOST	LOST	$m(3,3)$	$m(3,2)$	LOST	

Figure 3.8: Recovered codeword data structure for the case $Q = \{P_2, P_3\}$.

in $\mathcal{L}_{1,n}^c$ is of the form:

LOST	LOST	$p(1,k,2)$	$p(1,k,1)$	LOST
------	------	------------	------------	------

The parity symbols $p(1,k,1)$ and $p(1,k,2)$ are obtained as the k th elements from packets P_2 and P_3 . The “LOST” symbols represent erasures in the given codeword because the k th elements from the lost packets P_1 , P_4 and P_5 are not available. The codeword generated from each message word $m(1,k)$ in the highest priority layer $\mathcal{L}_{1,n}^c$ contains 4 parity symbols. The structure of the codeword allows full recovery of $m(1,k)$ in all cases up to 4 erasures. Therefore, in this example with 3 erasures, the decoder can successfully recover all the highest priority message words $\{m(1,k), k = 1, \dots, N\}$.

Similarly, the k th codeword generated from the message words $m(2, 2k)$ and $m(2, 2k - 1)$ in $\mathcal{L}_{2,n}^c$ is of the form:

LOST	LOST	$p(2,k,1)$	$m(2,2k)$	LOST
------	------	------------	-----------	------

The parity symbol $p(2, k, 1)$ and the message word $m(2, 2k)$ are obtained as the $(N + k)$ th elements of packets P_2 and P_3 , respectively. As all the codewords generated from $\mathcal{L}_{2,n}^c$ contain three parity symbols, the decoder can again recover all message words $\{m(2, k), k = 1, \dots, N\}$ from the layer-2 signal.

The k th codeword generated from $\mathcal{L}_{3,n}^c$ contains symbols:

LOST	LOST	$m(3, k \lceil \frac{N}{3} \rceil)$	$m(3, k \lceil \frac{N}{3} \rceil - 1)$	LOST
------	------	-------------------------------------	---	------

Symbols $m(3, k \lceil \frac{N}{3} \rceil)$ and $m(3, k \lceil \frac{N}{3} \rceil - 1)$ are obtained as the $(N + \lceil \frac{N}{2} \rceil + k)$ th elements of the packets P_2 and P_3 , respectively. In this case, the number of erasures is greater than two, which is the number of parity symbols. Therefore, the codewords generated from $\mathcal{L}_{3,n}^c$ cannot be correctly decoded. Similarly, codewords generated from the remaining sets $\mathcal{L}_{4,n}^c$ and $\mathcal{L}_{5,n}^c$ can also be not decoded. ■

In general, if l is the number of packets/descriptions in $\mathcal{Q}$, then the decoder successfully recovers all message words from layers at priority levels $1 \leq i \leq l$. The decoder accomplishes this task by decoding the first $\sum_{i=1}^l \lceil N/i \rceil$ symbols from each received packet. The remaining words in the received packets are ignored as they do not include sufficient number of parity symbols for decoding codewords with $5 - l$ erasures. Finally, for each recovered layer i , where $1 \leq i \leq l$, the decoder concatenates the message words $\{m(i, k), k = 1, \dots, N\}$ to generate the complete layer signal block $\mathcal{L}_{i,n}^c$.

The unpacketization process also strips any zero-valued message words from the recovered layer signals if zero padding was applied at the packetization stage. In particular, the last $i \lceil N/i \rceil - N$ words in each recovered layer $\mathcal{L}_{i,n}^c$ will be zero-valued.

3.5.2 Audio Decoding

Each $\mathcal{L}_{i,n}^c$ represents an mp3 bitstream. The decoder feeds each $\mathcal{L}_{i,n}^c$ into individual mp3 audio decoders (shown as the AC^{-1} blocks in Figure 3.7). The output of each audio decoder module is an M -sample long layer signal block. Let $\mathcal{L}_{i,n}^d$ be this M -sample long block for layer i . We use the superscript $(.)^d$ notation to differentiate each $\mathcal{L}_{i,n}^d$ from the corresponding decoded audio samples in $\mathcal{L}_{i,n}$. The difference between the “raw” uncompressed and decoded audio samples is due to the lossy mp3 compression algorithm.

Similarly, packets in the separate low frequency data stream P_W , if present, are decoded to generate the uncompressed low frequency audio samples $\mathbf{W}_n^d$.

3.5.3 Estimation

To estimate the channel signals from the decoded audio samples, we first form the $5 \times M$ -element matrix:

$$\mathcal{L}_n^d = \alpha_n \begin{bmatrix} \mathcal{L}_{1,n}^d \\ \vdots \\ \mathcal{L}_{l,n}^d \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (3.25)$$

where l is the number of packets/descriptions in $\mathcal{Q}$ and there are $(5 - l)$ trailing rows of zeros in place of the $5 - l$ lost packets. The formulation of the reconstructed data vector $\mathcal{L}_n^d$ also involves the denormalization of sample magnitudes using the scaling parameter α_n . The decoder uses the recovered $\mathbf{R}_{xx,n}$ autocorrelation matrix and formulates an estimate of the channels signals using the linear estimation process

described in Section 3.3:

$$\hat{\mathbf{x}}_n = \begin{bmatrix} \hat{L}_n \\ \hat{R}_n \\ \hat{C}_n \\ \hat{L}s_n \\ \hat{R}s_n \end{bmatrix} = \mathbf{H}(l) \mathcal{L}_n^d. \quad (3.26)$$

Thus, the structure of the data vector as defined in Equation (3.25) implies that only the elements in the first l columns of the matrix $\mathbf{H}(l)$ are significant. The elements in the remaining $5 - l$ columns of $\mathbf{H}(l)$ are multiplied by the zero-valued entries in $\mathcal{L}_{i,n}^d$ and therefore will not affect the signal estimate $\hat{\mathbf{x}}_n$. The pre-calculated elements of the matrix $\mathbf{H}(l)$ that would minimize the distortion for each of the cases when $1 \leq l \leq 5$ are available to the decoder. Appendix B lists for each case the elements of $\mathbf{H}(l)$ as a function of the elements of autocorrelation matrix $\mathbf{R}_{xx,n}$.

3.5.4 Synthesis

The function of the *Estimation* block is to re-generate the multi channel audio signals from the layer signals. In the case when some of the layer signals are lost during transmission, the *Estimation* block estimates the multi-channel audio signals from the recovered layers. In the worst case scenario, *all* packets representing the n th block (or multiple blocks) are lost and the decoder simply does not have access to any data which can be used to estimate the missing audio samples. In such a case the decoder can use different strategies. One possible approach is to silence the output for the duration of the missing block(s). Alternatively, the decoder can substitute the previously decoded block in place of the missing block motivated by the fact that each block represents a short audio segment (at 48 kHz sampling rate the duration of each M -sample long block is approximately $20M \mu s$) and the repetition of a single or a few consecutive blocks will not introduce noticeable distortion. The decoder can implement this substitution strategy up to a certain number of consecutive missing blocks. If the number of missing blocks exceed this threshold, then the decoder can silence the output. However, we recognize that complete packet loss particularly, for

an extended period of time, is a rather rare event. Therefore, the decoder used in this study implements a simple substitution strategy.

The compressed low frequency channel signal $\mathbf{W}_n^c$, when present, is transmitted independently and is not part of the PET packetization process. If any of the packets in the corresponding low frequency data stream P_W is lost, the decoder implements a similar packet substitution process as in the case of the missing packets from layers $\mathcal{L}_i$, $1 \leq i \leq 5$. We use the notation $\widehat{\mathbf{W}}_n$ to represent the decoded LFE channel signal.

To achieve the above described packet substitution strategy the *Synthesis* unit at the decoder buffers the most recently recovered block of audio samples. Thus, the function of the *Synthesis* unit is two-fold. First, recovered samples from received layers are buffered. Second, decoded samples in consecutive blocks (which may be fully recovered, partially estimated or repeated) are concatenated to build the audio channel estimates $\widehat{\mathbf{L}}$, $\widehat{\mathbf{R}}$, $\widehat{\mathbf{C}}$, $\widehat{\mathbf{L}}_s$, and $\widehat{\mathbf{R}}_s$. Each of these audio channel estimates includes the concatenated samples from consecutive blocks. For example, $\widehat{\mathbf{L}}$ will have the structure:

$$\begin{aligned}\widehat{\mathbf{L}} &= \{\widehat{\mathbf{L}}[1], \widehat{\mathbf{L}}[2], \dots\} \\ &= \{\widehat{\mathbf{L}}_1[1], \dots, \widehat{\mathbf{L}}_1[M], \widehat{\mathbf{L}}_2[1], \dots, \widehat{\mathbf{L}}_2[M], \dots\}.\end{aligned}\tag{3.27}$$

Finally, the output of the decoder will be:

$$\widehat{\mathbf{x}} = \begin{bmatrix} \widehat{\mathbf{L}} \\ \widehat{\mathbf{R}} \\ \widehat{\mathbf{C}} \\ \widehat{\mathbf{L}}_s \\ \widehat{\mathbf{R}}_s \end{bmatrix}, \tag{3.28}$$

which together with the concatenated LFE channel samples $\widehat{\mathbf{W}}$ represents the decoded multi-channel audio signal.

The *Synthesis* unit may impose a time limit on the arrival of packets from the same audio segment. For real-time processing, packets must be decoded within this limit. Therefore, the decoder will consider packets whose arrival time exceed this limit as lost.

Chapter 4

Results

IN THIS chapter we present the results obtained from the application of the layered multiple description multi-channel audio coding algorithm developed in Chapter 3 to a variety of sound files. Section 4.1 presents an analysis of how the optimal block size can be determined. Section 4.2 focuses on the normalized distortion values as a function of packet loss probability. Section 4.3 presents the results of the listening tests. Section 4.4 provides an extensive discussion of the results obtained and the nature of the listening tests.

4.1 Block Size M

The block size M is a key parameter that used used to segment input sequences into finite length blocks for processing. In an attempt to have a better understanding of how the M affects the performance of the coding algorithm we first investigate how the normalized distortion and output data rate change as a function of M and the number of descriptions received.

4.1.1 Distortion vs. Number of Descriptions Received

We first conducted a set of simulation studies aimed to quantify how the distortion resulting from estimating the multi-channel audio signal changes as a function of

the number of descriptions received. Towards this goal we calculate the average normalized distortion by calculating the mean squared error between the elements of the multi-channel audio signal $\mathbf{x}$ and its estimate $\hat{\mathbf{x}}$:

$$\mathcal{D} = \sum_{i=1}^5 \mathbb{E}[\|\mathbf{x}_{i:} - \hat{\mathbf{x}}_{i:}\|^2] / \mathbb{E}[\|\mathbf{x}_{i:}\|^2] \quad (4.1)$$

where the vectors $\mathbf{x}$, $\hat{\mathbf{x}}$ and their elements $\mathbf{x}_{i:}$, $\hat{\mathbf{x}}_{i:}$ are defined in Equations (3.1) and (3.28), respectively. In calculating the distortion measure we also assumed that the LFE channel $\mathbf{W}$, if one is present, is received in all simulation runs.

As a first test, we measured distortion as a function of the number of descriptions received. For each sound file in the database we formed the layered descriptions $\mathcal{L}_1^c, \dots, \mathcal{L}_5^c$. We then generated estimates of the multi-channel audio signal $\hat{\mathbf{x}}$ using different number of descriptions. Figure 4.1 depicts the maximum, minimum and average values of the normalized distortion over all sound files in the database as a function of the number of descriptions received. Since we wanted to measure the effect of the number descriptions received on distortion, these tests treated all sound files as a single data stream rather than processing them on a block-by-block basis. The figure shows decreasing distortion values with increasing number of descriptions received. This result is certainly expected, as with an increasing number of descriptions receiver, the decoder can fully recover some of the channel signals. For example, with only one description received, the decoder can determine only $\mathcal{L}_1^d$ and all channel signal must be estimated from $\mathcal{L}_1^d$; on the other hand if three descriptions are received, then the decoder can determine $\mathcal{L}_1^d$, $\mathcal{L}_2^d$ and $\mathcal{L}_3^d$ which are in turn used to fully recover $\mathbf{L}$, $\mathbf{R}$, $\mathbf{C}$ and to estimate the two surround channels $\mathbf{L}_s$ and $\mathbf{R}_s$. Therefore, the distortion values corresponding to the three descriptions received case will be much lower than the values in the case when only a single description is received.

Table 4.1 presents the normalized distortion values averaged over all the files in the database for several values of the block size M . Table 4.1 points out that distortion values decrease with increasing number of descriptions received. More significantly, Table 4.1 also indicates that distortion values are relatively insensitive to block size.

Figure 4.1: Normalized distortions values as a function of the number of descriptions received calculated over all sound files in the database.

Block size (samples)	Number of received streams				
	1	2	3	4	5
70000	1.2452×10^{-6}	9.9574×10^{-7}	7.6348×10^{-7}	6.2307×10^{-7}	0
110000	1.2413×10^{-6}	9.4582×10^{-7}	7.1467×10^{-7}	6.1776×10^{-7}	0
150000	1.2332×10^{-6}	9.3815×10^{-7}	7.0653×10^{-7}	6.1589×10^{-7}	0
250000	1.234×10^{-6}	9.348×10^{-7}	7.0311×10^{-7}	6.0016×10^{-7}	0

Table 4.1: Normalized distortion values averaged over all sound files.

4.1.2 Distortion vs. Block Size

To understand how distortion values depend on the block size, we conducted a second set of tests and simulated the transmission of packets formed from layered descriptions over a network with a packet loss probability of 10^{-1} . Under this scenario, packets corresponding to different blocks of input samples would be dropped randomly, and therefore the number of descriptions available for decoding would also change on a per block basis. We processed all sound files in the database using 48 different block sizes between 10^4 and 48×10^4 samples and measured the peak and average distortion values. Figure 4.2 presents the results. The data presented in Figure 4.2 supports our

Figure 4.2: Peak and average of normalized distortion values as a function of the blocksize at a packet loss probability of 10^{-1} .

earlier observation based on the results shown Table 4.1, namely that the distortion values are insensitive to block size. When we conducted these tests with different packet loss probabilities, very similar results were obtained.

4.2 Effects of Packet Loss Probability

Table 4.1 and Figure 4.1 present the results of the simulations when the number of descriptions received were forced to be identical for all blocks, i.e., for the entire test file. While these simulations are useful in assessing the quality of the estimates, they do not reflect a real scenario. As we remarked earlier, under a more realistic scenario packets corresponding to different blocks would be dropped randomly, and therefore the number of descriptions available for decoding would also change randomly on a per block basis. In this section, we explore how distortion changes as a function packet loss probability.

To simulate the dynamic environment that may be encountered in a network we tested the performance of the layered coding algorithm for 5 packet loss probability values at 0, 10^{-4} , 10^{-3} , 10^{-2} , and 10^{-1} . In order to distinguish results obtained with different packet loss probabilities (particularly at low probability values) we had to conduct our tests on sufficiently long sound files. Therefore, we measured the average and peak normalized distortion values using 10 sound files of 105 second duration each. These files are the extended versions of some of the shorter sound files in the main database. With a selected block size of $M = 3000$ samples, a sound file of 105 second duration generated at a sampling rate of 48 kHz would result in a total of 10,080 packets representing all six audio channels.

Figure 4.3 depicts the peak and average normalized distortion values resulting from simulation tests using these 10 sound files. Table 4.2 provides the numerical results corresponding to the same simulation runs. The results depicted in Figure 4.3 and Table 4.2 indicate that the distortion values decrease with decreasing probability of packet losses.

Figure 4.3: Peak and average normalized distortion values determined with 10 sound files of 105 second duration with $M = 3000$ samples.

Packet loss probability	Normalized distortion	Lost packets
0	3.1139×10^{-8}	0
10^{-4}	3.1236×10^{-8}	1
10^{-3}	3.2386×10^{-8}	9.1
10^{-2}	4.7596×10^{-8}	92.2
10^{-1}	1.9289×10^{-7}	969.1

Table 4.2: Peak and average normalized distortion values determined with 10 sound files of 105 second duration with $M = 3000$ samples.

4.3 Listening Tests

4.3.1 Test Procedures

Listening tests were conducted in accordance with a formal procedure ITU-R Recommendation BS.1116 [25] suitable for subjective quality measures of the small impairments. In the procedure, the listener rates sound quality with respect to the Mean Opinion Score (MOS) scale, or equivalently, by the impairment level scale. The MOS grade 5 corresponds to the *excellent* quality, and lower grades are assigned to the *good* (4), *fair* (3), *poor* (2), and *bad* (1) quality. Table 4.3 summarizes the grading scales [25].

Impairment	Mean opinion score	Subjective quality	Absolute grade	Subjective difference grade
Imperceptible	5	Excellent	5	0
Perceptible, not annoying	4	Good	[4, 5)	[-1, 0)
Slightly annoying	3	Fair	[3, 4)	[-2, -1)
Annoying	2	Poor	[2, 3)	[-3, -2)
Very annoying	1	Bad	[1, 2)	[-4, -3)

Table 4.3: Grading scales, [25].

At each step of the listening test, listeners are asked to listen to three stimuli. One stimulus is the original sound and represents the reference. The listener has to compare the reference with the two other sounds, one of which is identical to the reference and the other is the estimated multi-channel audio signal generated by the coding algorithm. Such a test environment and the accompanying procedures are referred to as a double-blind triple-stimulus hidden reference system. For rating the sound quality in the listening tests we used the software developed in [39] which was modified for multi-channel audio signals. Figure 4.4 depicts the user-interface of this software.

	A	B	C
Imperceptible		☐	☒
Perceptible - Not Annoying		☒	☐
Slightly Annoying		☐	☐
Annoying		☐	☐
Very Annoying		☐	☐

1 of 7

Next

Figure 4.4: Listening test interface, [39].

We subtract the score given to the hidden reference from the score given to the coded signal in order to obtain the subjective difference grade (SDG) relative to the hidden reference. The SDG can have a negative or zero value. The SDG parameter is of importance in cases when a listener without prior knowledge that one of the two other stimuli is identical to the reference signal, evaluates the hidden reference with a grade lower than 5, and the coded sound with an equal or lower grade (a higher MOS score implies better sound quality). This scenario implies that the listener correctly observed the difference in quality between the two stimuli. However, since the hidden reference was not graded 5, it is assumed that MOS grades for both hidden reference and the coded signal are reduced by the same number, and the marking is still considered valid as if the hidden reference was graded 5.

The mapping of the SDG to the descriptive values for subjective quality as listed in Table 4.3 is obtained by adding 5 to the SDG value, thus yielding the absolute grade. Table 4.3 provides both the SDG and absolute grades as values that belong to specific intervals. This is a consequence of the listening tests, when we calculated the average values of the SDG and absolute grades with the averages taken over all grades assigned by listeners. This averaging operation results in non-integer values for the SDG and absolute grades.

For listening tests we selected twelve multi-channel sound files from the database (**File 1–File 12**). All files had a duration of 9–10 s. With the block size $M = 65,000$ samples, there were on the average 7 blocks per file. In each listening test the original sound file was available as a reference (button A in Figure 4.4) and also as one of the two stimuli offered for grading (under button B or C, which was not known to the listener). The coded sound signal was assigned to the third button as the second stimuli.

The subjective grade for each sound file is determined as the average of all the MOS grades obtained from the listeners. In determining the subjective grades we have taken into account only the grades assigned by listeners who correctly identified the hidden reference—correct identification implies that the assessed sound quality of the hidden reference was better or equal than the sound quality of the coded signal. Since not all the listeners assigned the highest absolute grade (5) to hidden reference signals, we also report average absolute grades, which provide a more subtle insight into the subjective qualities of various sound files.

4.3.2 Results

In the first part of the listening test, which consisted of five steps, our objective was to measure the subjective sound quality as a function of the number of descriptions received. Each step corresponded to one of the cases when exactly 1, 2, 3, 4 or 5 descriptions were received. We presented the coded sound files for each of the five cases to seven listeners in random order. The reason for the random ordering was to keep the listeners alert by avoiding the expected gradual sound degradation. The sound files used in this listening test were **File 1–File 5**.

Table 4.4 displays the results of the first part of this test. The results represent the average of the opinion scores assigned to sound files by seven listeners.

In the second part of the listening test, we conducted listening tests using **File 1–File 5** with packet loss probabilities at 0.01 and 0.7. Table 4.5 displays the results which are the average of the opinion scores assigned to sound files by all listeners.

Number of received descriptions	Average MOS grade				
	File 1	File 2	File 3	File 4	File 5
1	3.4286	3.0000	4.0000	3.8571	3.1429
2	3.7143	3.2857	4.0000	3.7143	3.8571
3	4.0000	3.8333	4.4286	4.1667	4.0000
4	4.0000	4.5714	4.4286	4.1429	4.4286
5	4.8333	5.0000	4.6667	4.7143	4.5000

(a)

Number of received descriptions	Average absolute grade				
	File 1	File 2	File 3	File 4	File 5
1	3.7143	3.4286	4.1429	4.0000	3.2857
2	4.0000	3.4286	4.1429	4.0000	4.0000
3	4.1429	4.1667	4.5714	4.1667	4.1429
4	4.1429	4.7143	4.5714	4.1429	4.5714
5	5.0000	5.0000	4.6667	5.0000	4.5000

(b)

Table 4.4: Listening tests results: (a) average MOS grades and (b) absolute grades as a function of the number of descriptions received.

As multi-channel audio signals may represent diverse sound sources we conducted a second listening test which consisted of the same steps as before (**Part 1:** listening test with coded signals as a function of the number of descriptions received, and **Part 2:** listening test with coded signals as a function of the packet loss probability). However, the coded sound file presented to the listeners in each step of the second listening test corresponded to a different sound file. In particular, **Files 6–12** used in the second test are as follows.

File 6: coded sound file with 1 description received.

File 7: coded sound file with 2 descriptions received.

Packet loss probability	Average MOS grade				
	File 1	File 2	File 3	File 4	File 5
0.01	4.1429	4.6667	4.5714	4.2857	4.5000
0.7	2.8571	2.7143	3.4286	3.1429	3.1429

(a)

Packet loss probability	Average absolute grade				
	File 1	File 2	File 3	File 4	File 5
0.01	4.2857	4.6667	4.7143	4.4286	4.5000
0.7	3.0000	2.8571	3.5714	3.4286	3.2857

(b)

Table 4.5: Listening tests results: (a) average MOS grades and (b) absolute grades as a function of the packet loss probability.

File 8: coded sound file with 3 descriptions received.

File 9: coded sound file with 4 descriptions received.

File 10: coded sound file with 5 (all) descriptions received.

File 11: coded sound file at packet loss probability 0.01.

File 12: coded sound file at packet loss probability 0.7.

Table 4.6 displays the results of the listening tests with **File 6–File 12**. The results represent the average of the opinion scores assigned to sound files by eight listeners.

4.4 Discussion

In this chapter we tested the proposed algorithm with several audio files and presented the results. Since the sound quality resulting from an audio coding algorithm is subjective by nature, we tested the performance of the algorithm with respect to the following two criteria:

- distortion resulting from estimating the audio signal in cases when some of the

Number of received descriptions	Signal	Average	
		MOS grade	Absolute grade
1	File 6	3.625	3.875
2	File 7	3.430	3.430
3	File 8	3.670	3.670
4	File 9	4.000	4.170
5	File 10	4.670	4.830

(a)

Packet loss probability	Signal	Average	
		MOS grade	Absolute grade
0.01	File 11	4.80	4.80
0.7	File 12	2.57	2.71

(b)

Table 4.6: Listening tests results: (a) average MOS grades and (b) absolute grades for various sound files with different number of descriptions received and packet loss probabilities.

descriptions are not available at the decoder due to packet losses, and

- subjective sound quality assessed in listening tests performed by independent listeners.

In particular, we measured both criteria as functions of the number of descriptions received and the packet loss probability.

4.4.1 Distortion

Number of descriptions received

In terms of the number of the received descriptions, the main objective of the tests performed was to measure the changes in the distortion as a function of the descrip-

tions received. The expected result was that distortion values would decrease as more descriptions are received. This is certainly an expected result, as more descriptions provide additional information on the transmitted signal. Consequently, the decoder can estimate the transmitted signal with a higher degree of fidelity. Indeed, the results presented in Figure 4.1 and Table 4.1 confirm these expectations.

In Figure 4.1, the case with four received descriptions may be singled out, because of the unexpectedly high maximum distortion value, which contributes to the slightly higher average value that does not conform exactly to the almost linear trend of four other average values. One of the sound files in the database was the reason for this observed deviation. The audio database consists of nearly 80 different files, and as such it can be considered to include a fair representation of the diversity in audio samples. Therefore, the “offending” file may still represent an example of a sound file which may be encountered in practice. This was the main reason why we started with an extensive database to test the algorithm with several different audio files.

Packet loss probability

In terms of the packet loss probability, the objective was to observe changes in the distortion resulting from various packet loss probabilities. At higher packet loss probabilities, more packets are likely to be lost, which would in turn imply that fewer descriptions will be available at the decoder, thus potentially increasing the distortion. In the simulation runs with 10 long files, the length of each extended length sound file provided sufficient number of packets to realistically test average and maximum distortion values resulting from packet losses. Results given in Figure 4.3 in Table 4.2 confirm predicted trends in performance. In particular, we measured both criteria as functions of the number of descriptions received and packet loss probability.

4.4.2 Subjective Sound Quality

A low distortion measure does not necessarily translate into good sound quality. Therefore, we conducted extensive tests to assess the sound quality of the decoded

multi-channel audio signals. We recognize that the outcome of the listening tests are still subjective, and therefore, they must be interpreted in the light of the circumstances that may influence listeners.

The informal listening tests involved a total of twelve listeners who were all non-expert listeners. The tests took place in a sound chamber, an audio-metric room, equipped with 5.1 surround sound system. Each listener individually evaluated the sounds presented via the testing software described earlier. Some listeners were familiar with the interface, since they took part in an earlier listening tests which were independent of this work and used different coding techniques. A few listeners, however, asked for clarification regarding the descriptive grades they had been asked to assign at the end of each listening test. The listening tests conducted also represented the first time for all listeners when they sat in an audio-metric room. Listeners' views of the room environment differed greatly, some found it claustrophobic while others found it pleasing. Both factors, namely the listeners' familiarity with the testing procedures and their assessment of the physical environment, i.e., the audio-metric room, had been likely factors that may have influenced the listening test results.

The selection of the sound files could have been another factor affecting the results. Listeners evaluated a classical choral piece, sound tracks from a musical as well as from action, adventure and animated movies. The subjective preference of each listener for a particular genre may be reflected in the ratings assigned by the listener, since particular type of sounds might have helped to focus the listener's attention. Also, familiarity with the content raises listener's alertness to the sound differences.

There is yet another factor linked with the presented sound content. The way in which the proposed algorithm formed the layers, assumed that most of the time the perceptually significant part in a sound track was in the three main channels (center, left and right). However, this was simply not necessarily the case with each and every sound file in the database. For example, the sound in one of the files was "panning" from one channel to the next starting at the left surround channel, thus creating an auditory illusion that the object was moving in the spatial domain.

Familiarity of the listeners with the person who co-ordinated and conducted the listening tests, may also have been an influencing factor. In addition, it has been observed that verbal encouragement given, and words of appreciation extended to the participants, motivated listeners for mental readiness and concentration during the numerous tests. In preparation for the listening tests, the listeners had not been informed that one of the two stimuli they had been asked to assess, was always the same as the reference signal. This approach where a hidden reference signal is presented as one of the two stimuli, allows the elimination of invalid responses as discussed earlier. Furthermore, to eliminate any further bias, the ordering of all stimuli in all listening tests was randomized.

There were two sets of listening experiments conducted. In the first experiment, listeners were presented with versions of the same sound file in all seven tests, as described in Section 4.3. In the second experiment, listeners assessed test files, which for each of the seven tests—corresponding to different coding scenarios—had been generated from different sound files. The feedback from a significant number of listeners indicated that the latter approach was the preferred model for the experiment. In the first experiment a listener would listen to seven different versions of the same sound file; predictably this process may quickly become tedious for the listener and may result in a loss of concentration, which is essential for successfully differentiating subtle variations in the sound quality. On the other hand, the ability to listen to different versions of the same sound file in quick succession may allow those listeners who are more sensitive to differences (or who may have short auditory memories) to successfully differentiate such subtle variations.

We remind the reader that there are many factors that may influence the outcome of a listening test. Listening tests are always subjective and the results must always be interpreted as such. However, we believe that by using an extensive sound database and proper experimental procedures, we were able to create a test environment which mitigated the effects of such factors.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this study, we developed a novel coding algorithm that is proven to be suitable for transmitting multi-channel audio samples over packet networks. The algorithm exploits the statistical inter-channel dependencies that are inherent in multi-channel audio signals. This information is in turn used to generate multiple descriptions and also to formulate estimation strategies. The algorithm combines key features of multiple description and layered audio coding techniques. We measured the performance of the new multi-channel audio coding scheme in terms of the peak and average distortion values resulting from estimating the audio signal in cases when the decoder does not have access to all descriptions due to packet losses. We tested the performance of the algorithm under various packet loss probabilities, and also evaluated the subjective sound quality of the estimated multi-channel audio signal.

We state the following conclusions resulting from the analysis of the algorithm, simulation results and the observations we made in developing the algorithm.

- The proposed algorithm generates layered descriptions by forming linear combinations of multi channel signals. The structure of linear transformation used is in part based on the perceptual significance of channel signals.

- We use a priority encoded transmission technique to transform layered signals into coded multiple descriptions such that all packets representing the multiple descriptions are of equal value. The quality of the decoded signal is directly proportional to the number of descriptions available at the decoder which is in turn determined by the number of packets received. The proposed algorithm combines the best characteristics of both (multiple description and layered coding) techniques.
- The proposed algorithm is robust with respect to packet losses. At a packet loss probability value of 10^{-1} —which may be considered a value higher than typically encountered in today’s packet transmission networks—the subjective sound quality of the decoded signal is rated as excellent with no perceptible impairments due to coding and/or packet losses.
- Due to pre-determined estimator parameters the decoder structure is simple to implement. This structure renders the algorithm particularly suitable for real-time applications.
- The encoder in the proposed algorithm can be implemented as pre-processing (layering and normalization) and post-processing (priority encoded transmission and packetization) modules to standard audio compression algorithms. Similarly, the decoder also consists of pre- and post-processing functions with respect to standard audio decompression algorithms. Therefore, the proposed algorithm can be easily retrofitted into existing processing units.

5.2 Future Work

The objects of this study have been multi-channel audio signals. We used the inter-channel correlations to form layers and also to estimate channel signals which were lost during transmission. In particular, we used linear transformations to form layers and used linear estimator structures. An extension of this work would be to extend processing to other transformations and estimators possibly including non-linear

structures. However, linear structures are simple to implement and yield highly satisfactory performance. Therefore, any marginal benefits (in terms of lower distortion and/or higher subjective sound quality) to be gained from other structures should be carefully weighted against the simplicity of implementation with linear structures.

The multiple description layered coding algorithm developed in this study introduces redundancy by Reed-Solomon error correction coding of the compressed audio data before transmitting packets over the network. This results in approximately a twofold increase in the output data rate. In return, the coded output data stream is robust with respect to packet losses. While the tradeoff between the output data rate and robustness can be considered a reasonable compromise, further decrease in the output data rate is a most desirable outcome. The current algorithm uses a stationary strategy for coding compressed audio data streams. A potential extension of the current study would include the investigation of how the algorithm can be made adaptive to dynamic changes in the network, such that less error correcting coding can be used at times of low traffic (and therefore low packet loss probability) thus reducing the output data rate while maintaining the performance measures of the full algorithm. Another network-centric extension would be to further study the logistics of devising a communication procedure that would allow clients with different connection bandwidths to exchange missing layers.

The algorithm developed in this study uses inter-channel correlations and in part the perceptual significance of audio signals to group channel signals into layers. Given the fact that many multimedia signal coding algorithms are becoming more object-oriented, a layering approach based on objects in the audio data stream would represent an interesting approach to audio coding. For example, if a movie soundtrack consists of sound objects such as *dialogue*, *background music*, *background noise*, *sound track*, how would one design an algorithm that uses the significance of the sound objects to form layers and assign priorities? These questions and most importantly solutions to such questions are becoming increasingly relevant for tomorrow's multimedia signal processing algorithms.

Appendix A

Reed-Solomon Codes

The discussion presented in this Appendix is based on [34]. Reed-Solomon codes are a class of error-correcting, non-binary cyclic codes, which are also a subset of the Bose, Chaudhuri, and Hocquenghem (BCH) codes. Let us denote data symbols which are to be coded as a “message”. Each symbol represents a b -bit sequence, $b \in \mathcal{Z}$, $b > 2$. A message consists of k symbols. The codeword obtained by encoding the message consists of a total of K symbols. The applied Reed-Solomon code is denoted $\mathcal{RS}(K, k)$. Such a code exists for:

$$0 < k < K < 2^b + 2. \quad (\text{A.1})$$

The message is included in the codeword. The remaining $K - k$ symbols of the codeword are the parity symbols. The symbol-error correcting capability t of the code is equal to:

$$t = \left\lfloor \frac{K - k}{2} \right\rfloor, \quad (\text{A.2})$$

which implies that it is possible to correct up to t errors. The erasure correcting capability δ of the code is:

$$\delta = K - k, \quad (\text{A.3})$$

which equals to the number of parity symbols. In this study, we are most interested in the erasure correcting capabilities of the code, as packet losses that are at the heart of the problem corresponds to erasures.

The main property of the $\mathcal{RS}$ codes, used in this study, is that the code is able to correct any $K - k$ erasures. There are no constraints where the erasures can be; they can be among the message or the parity symbols. In other words, as long as at least k symbols out of the codeword's K symbols are not erased, it is possible to decode the k symbols long message. This observation implies that all symbols in the codeword are equally important.

The $\mathcal{RS}$ processes of coding and decoding rely on the theory of finite fields, which we address briefly first, before describing the processes.

A.1 Galois Fields in the Construction of $\mathcal{RS}$ Codes

Any prime number p determines a Galois field $GF(p)$ with p elements. In the construction of $\mathcal{RS}$ codes, an extension field of $GF(2)$, a Galois field of 2^b elements denoted $GF(2^b)$, is used, with $b > 0$, $b \in \mathcal{Z}$. There are 2^b unique elements in the $GF(2^b)$, and the set of elements includes 0, 1, and other non-zero elements which all may be represented by an integer power of one of the elements, denoted α :

$$GF(2^b) = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{2^b-2}\}. \quad (\text{A.4})$$

The set $GF(2^b)$ is closed under multiplication.

The finite field $GF(2^b)$ is defined by a *primitive polynomial*. A primitive polynomial is an irreducible polynomial which has at least one root that may generate all the non-zero elements of the field. All b roots of the primitive polynomial $f(X)$ are elements of $GF(2^b)$. The remaining elements of $GF(2^b)$ may be expressed as weighted sums of this root and its powers of order up to $b - 1$, which may be mapped into b -tuples of zeros and ones. The primitive polynomial defines addition and multiplication in the field as well as the elements of the field.

A.2 $\mathcal{RS}$ Encoding

The message being $\mathcal{RS}(K, k)$ encoded may be represented in the form of a polynomial $m(X)$. If the number of the symbols in the message is k , the degree of the polynomial is $k-1$. Symbols are elements of the Galois field described in the previous section: the message is an array of zeros and ones, which is mapped into symbols, i.e., elements of Galois field. The symbols are coefficients of the polynomial.

The resulting codeword also may be represented by a polynomial $U(X)$ of degree $K-1$. Since the codeword contains the message, $U(X)$ may be expressed as:

$$U(X) = p(X) + X^{K-k}m(X), \quad (\text{A.5})$$

with $p(X)$ being the parity polynomial:

$$p(X) = \text{mod}(X^{K-k}m(X), g(X)), \quad (\text{A.6})$$

where $g(X)$ is a generator polynomial for the particular $\mathcal{RS}$ code. The degree of $g(X)$ is $K-k$. $K-k$ elements of the Galois field are roots of $g(X)$. If the elements $\alpha, \alpha^2, \dots, \alpha^{K-k}$ are designated to be roots of $g(X)$ for a particular $\mathcal{RS}$ code, then $g(X)$ is of the form:

$$g(X) = (X - \alpha)(X - \alpha^2) \cdots (X - \alpha^{K-k}). \quad (\text{A.7})$$

Applying the rules of multiplication and addition defined in the Galois field simplifies the polynomial form of $g(X)$.

Determining $p(X)$ and thus $U(X)$ requires modulo operations, i.e., division of $X^{K-k}m(X)$ by $g(X)$, where the rules of multiplication and addition as defined in the Galois field are applied. $U(X)$ may also be represented in the form:

$$U(X) = m(X)g(X), \quad (\text{A.8})$$

therefore $U(X)$ has roots identical to those of $g(X)$.

A.3 $\mathcal{RS}$ Decoding with Focus on Erasures

Let $r(X)$ denote the polynomial of the received codeword. The degree of $r(X)$ is $K - 1$ which is the same as the degree of $U(X)$. In the process of decoding, $r(X)$ is evaluated at all the roots of $g(X)$ in order to obtain $K - k$ values of the syndrome, namely, $S_1, S_2, \dots, S_{K-k}$. If $r(X) = U(X)$, all those values are equal to zero. If there is an error $e(X)$, it will be:

$$r(X) = U(X) + e(X), \quad (\text{A.9})$$

and syndrome values for $r(X)$ and $e(X)$ must be the same. The degree of $e(X)$ is $K - 1$. An erasure is a special case of an error, when the location of the error is known. If the $j \leq K - k$ erasures (errors) are at locations $l_1, l_2, \dots, l_j$, then the error polynomial is of the form:

$$e(X) = e_1 X^{l_1} + e_2 X^{l_2} + \dots + e_j X^{l_j}. \quad (\text{A.10})$$

Substitution of X by the roots of $g(X)$ i.e., $\alpha, \alpha^2, \dots, \alpha^{K-k}$, into (A.10), yields syndrome values, and a system of $K - k$ equations. The equations are of the form:

$$S_i = r(\alpha^i) = e_1(\alpha^{l_1})^i + e_2(\alpha^{l_2})^i + \dots + e_j(\alpha^{l_j})^i, \quad i = 1, 2, \dots, K - k. \quad (\text{A.11})$$

To solve for $e_1, e_2, \dots, e_j$, any j equations from the system may be used. Using the first j equations, the desired solution is:

$$\begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_j \end{bmatrix} = \begin{bmatrix} \alpha^{l_1} & \alpha^{l_2} & \dots & \alpha^{l_j} \\ (\alpha^{l_1})^2 & (\alpha^{l_2})^2 & \dots & (\alpha^{l_j})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha^{l_1})^j & (\alpha^{l_2})^j & \dots & (\alpha^{l_j})^j \end{bmatrix}^{-1} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_j \end{bmatrix}, \quad (\text{A.12})$$

which, replaced in (A.10), yields $e(X)$. Finally, using Equation (A.9) and applying addition rules within the field, we can express the recovered valid codeword as:

$$U(X) = r(X) + e(X), \quad (\text{A.13})$$

and the message $m(X)$ being decoded is contained within $U(X)$, as described previously.

Appendix B

Elements of $\mathbf{H}$

Appendix B lists the elements $\mathbf{H}(l)$ for each of the cases when there are exactly l descriptions received. $\mathbf{H}(l)$ determines the structure of the estimator as shown in Equation (3.26). For each transmitted data block/segment, the decoder first determines l , the number of descriptions received, and extracts the autocorrelation matrix $\mathbf{R}_{xx,n}$ and the scaling factor α_n from the highest priority description. The decoder then determines the elements of $\mathbf{H}(l)$ using $\mathbf{R}_{xx,n}$ and α_n with the pre-determined design equations given in this Appendix. Let $h_{ij}(l)$ and ρ_{ij} be the (i, j) th elements of $\mathbf{H}(l)$ and $\mathbf{R}_{xx,n}$, respectively.

B.1 Case $l = 1$

When only one description is received, all audio channel signals are estimated from $\mathcal{L}_{1,n}^d$. Therefore, only $\mathbf{H}_{:1}(1)$, i.e., the first column of $\mathbf{H}(1)$, needs to be evaluated. In particular, $\mathbf{H}(1)$ will have the structure

$$\mathbf{H}(1) = \begin{bmatrix} h_{11}(1) & * & * & * & * \\ h_{21}(1) & * & * & * & * \\ h_{31}(1) & * & * & * & * \\ h_{41}(1) & * & * & * & * \\ h_{51}(1) & * & * & * & * \end{bmatrix}, \quad (\text{B.1})$$

where once again the elements marked as “*” represent the “do not care” values. Minimizing the distortion $\mathcal{D}$ with respect to $\mathbf{H}_{\cdot 1}(1)$ yields to solution:

$$h_{i1} = \sum_{j=1}^3 \rho_{1j} / \sum_{k=1}^3 \sum_{j=1}^3 \rho_{kj} \quad i = 1, \dots, 5. \quad (\text{B.2})$$

B.2 Case $l = 2$

When there are two descriptions received, all audio channels are estimated from the first two layer descriptions $\mathcal{L}_{1,n}^d$ and $\mathcal{L}_{2,n}^d$. The center audio channel signal $\mathbf{C}_n$ may be fully¹ recovered from scaled difference $\mathcal{L}_{1,n}^d - \mathcal{L}_{2,n}^d$. Therefore the matrix $\mathbf{H}(2)$ is of the form:

$$\mathbf{H}(2) = \begin{bmatrix} h_{11}(2) & h_{12}(2) & * & * & * \\ h_{21}(2) & h_{22}(2) & * & * & * \\ 1 & -1 & * & * & * \\ h_{41}(2) & h_{42}(2) & * & * & * \\ h_{51}(2) & h_{52}(2) & * & * & * \end{bmatrix}. \quad (\text{B.3})$$

The estimator now faces the problem of estimating two groups of audio channel signals. Let $\mathcal{G}_1 = \{\widehat{\mathbf{L}}_n, \widehat{\mathbf{R}}_n\}$ and $\mathcal{G}_2 = \{\widehat{\mathbf{L}}_{s,n}, \widehat{\mathbf{R}}_{s,n}\}$ be the estimated front and surround signal groups, respectively. Based on the outcome of the inter-channel correlation analysis presented in Section 3.2 and the experience in working with the intensity coding of stereo audio signals we identified four potential solutions for the elements of the estimator matrix $\mathbf{H}(2)$. These solutions differ in terms of the dependence of $\mathcal{G}_1$ and $\mathcal{G}_2$ on the two received layer descriptions $\mathcal{L}_{1,n}^d$ and $\mathcal{L}_{2,n}^d$. Table B.1 shows how each solution depends on the two received layer descriptions. The estimator computes the distortion resulting from each of the four solution and chooses the one that would result in the minimum distortion. Elements of $\mathbf{H}(2)$ for each of the four optimal

¹The reader should note that the center audio channel signal $\mathbf{C}_n$ being referred here differs from the original $\mathbf{C}_n$ due to the lossy mp3 audio coding (as represented by the AC block in the encoder and the AC^{-1} block in the decoder). However, to simplify the notation and due to the imperceptible difference between the original and the decoded signals, we continue to use the notation $\mathbf{C}_n$ to represent both signals.

Layer	Solution							
	1		2		3		4	
	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
$\mathcal{L}_{1,n}^d$				✓	✓		✓	✓
$\mathcal{L}_{2,n}^d$	✓	✓	✓	✓	✓	✓	✓	✓

Table B.1: Dependence of $\mathcal{G}_1$ and $\mathcal{G}_2$ estimates on received layers.

solutions are defined as follows.

Solution-I:

(B.4)

$$\begin{aligned}
 \mathbf{h}_{i1}^I(2) &= [0 \ 0 \ 1 \ 0 \ 0]^T \\
 h_{i2}^I(2) &= \begin{cases} \sum_{j=1}^2 \rho_{ij} / \sum_{k=1}^2 \sum_{j=1}^2 \rho_{kj}, & i = 1, 2, 4, 5; \\ -1, & i = 3. \end{cases}
 \end{aligned}$$

Solution-II:

(B.5)

$$\begin{aligned}
 h_{i1}^{II}(2) &= 0, \quad i = 1, 2, \\
 h_{3j}^{II}(2) &= (-1)^{j-1}, \quad j = 1, 2, \\
 h_{ij}^{II}(2) &= (-1)^{j-1} \frac{\rho_{i3}}{\rho_{33}}, \quad i = 4, 5, j = 1, 2, \\
 h_{i2}^{II}(2) &= \sum_{l=1}^2 \rho_{il} / \sum_{k=1}^2 \sum_{j=1}^2 \rho_{kj}, \quad i = 1, 2.
 \end{aligned}$$

Solution-III:

(B.6)

$$\begin{aligned}
 h_{i1}^{III}(2) &= \frac{\rho_{i3}(\sum_{k=1}^2 \sum_{j=1}^2 \rho_{kj}) - (\sum_{l=1}^2 \rho_{il})(\sum_{m=1}^2 \rho_{m3})}{\rho_{33}(\sum_{k=1}^2 \sum_{j=1}^2 \rho_{kj}) - (\sum_{m=1}^2 \rho_{m3})^2}, \quad i = 1, 2, \\
 h_{3j}^{III}(2) &= (-1)^{j-1}, \quad j = 1, 2, \\
 h_{i1}^{III}(2) &= 0 \quad i = 4, 5, \\
 h_{i2}^{III}(2) &= \left(\frac{\rho_{i3}}{\rho_{33}} - h_{i1}^{III}(2) \right) \left(\sum_{k=1}^3 \rho_{k3} / \sum_{l=1}^2 \rho_{l3} \right) - \frac{\rho_{i3}}{\rho_{33}}, \quad i = 1, 2, \\
 h_{i2}^{III}(2) &= \sum_{l=1}^2 \rho_{il} / \sum_{k=1}^2 \sum_{j=1}^2 \rho_{kj}, \quad i = 4, 5.
 \end{aligned}$$

Solution-IV:

$$\begin{aligned}
 h_{i1}^{IV}(2) &= \frac{\rho_{i3}(\sum_{k=1}^2 \sum_{j=1}^2 \rho_{kj}) - (\sum_{l=1}^2 \rho_{il})(\sum_{m=1}^2 \rho_{m3})}{\rho_{33}(\sum_{k=1}^2 \sum_{j=1}^2 \rho_{kj}) - (\sum_{m=1}^2 \rho_{m3})^2}, \quad i = 1, 2, \\
 h_{3j}^{IV}(2) &= (-1)^{j-1}, \quad j = 1, 2, \\
 h_{ij}^{III}(2) &= (-1)^{j-1} \frac{\rho_{i3}}{\rho_{33}}, \quad i = 4, 5, j = 1, 2, \\
 h_{i2}^{IV}(2) &= \left(\frac{\rho_{i3}}{\rho_{33}} - h_{i1}^{III}(2) \right) \left(\sum_{k=1}^3 \rho_{k3} / \sum_{l=1}^2 \rho_{l3} \right) - \frac{\rho_{i3}}{\rho_{33}}, \quad i = 1, 2.
 \end{aligned} \tag{B.7}$$

B.3 Case $l = 3$

When there are three descriptions received, all five audio channel signals are determined using the first three layer descriptions $\mathcal{L}_{1,n}^d$, $\mathcal{L}_{2,n}^d$, and $\mathcal{L}_{3,n}^d$. The estimator matrix $\mathbf{H}(3)$ is given as follows:

$$\mathbf{H}(3) = \begin{bmatrix} h_{11}(3) & 1/2 & 1/2 & * & * \\ h_{21}(3) & 1/2 & -1/2 & * & * \\ 1 & -1 & h_{33}(3) & * & * \\ h_{41}(3) & h_{42}(3) & h_{43}(3) & * & * \\ h_{51}(3) & h_{52}(3) & h_{53}(3) & * & * \end{bmatrix}. \tag{B.8}$$

In this case, the estimator fully recovers $\mathbf{L}_n$, $\mathbf{R}_n$, and $\mathbf{C}_n$, and determines the estimates of the surround channels $\widehat{\mathbf{L}}\mathbf{s}_n$ and $\widehat{\mathbf{R}}\mathbf{s}_n$ using either one of the two solutions for the elements of $\mathbf{H}(3)$. Table B.2 shows the dependence of the solutions considered on the received layer descriptions.

The particulars of the two solutions for the elements of $\mathbf{H}(3)$ are listed below.

Solution-I:

$$\begin{aligned}
 h_{11}^I(3) &= h_{21}^I(3) = h_{41}^I(3) = h_{51}^I(3) = h_{33}^I(3) = 0, \\
 h_{31}^I(3) &= 1 = -h_{32}^I(3),
 \end{aligned} \tag{B.9}$$

Layer	Solution	
	1	2
$\mathcal{L}_{1,n}^d$		✓
$\mathcal{L}_{2,n}^d$	✓	✓
$\mathcal{L}_{3,n}^d$	✓	

Table B.2: Layers used in the estimation of $\widehat{\mathbf{L}}\mathbf{s}_n$ and $\widehat{\mathbf{R}}\mathbf{s}_n$.

$$\begin{aligned}
h_{12}^I(3) &= h_{22}^I(3) = h_{13}^I(3) = \frac{1}{2} = -h_{23}^I(3), \\
h_{42}^I(3) &= h_{43}^I(3) = \frac{\rho_{41}}{2\rho_{11}}, \\
h_{52}^I(3) &= \frac{\rho_{52}}{2\rho_{22}} = -h_{53}^I(3).
\end{aligned}$$

Solution-II:

(B.10)

$$\begin{aligned}
h_{11}^{II}(3) &= h_{21}^{II}(3) = h_{33}^{II}(3) = h_{43}^{II}(3) = h_{53}^{II}(3) = 0, \\
h_{31}^{II}(3) &= 1 = -h_{32}^{II}(3), \\
h_{ij}^{II}(3) &= (-1)^{j-1} \frac{\rho_{i3}}{\rho_{33}}, \quad i = 4, 5, j = 1, 2, \\
h_{12}^{II}(3) &= h_{22}^{II}(3) = h_{13}^{II}(3) = \frac{1}{2} = -h_{23}^{II}(3).
\end{aligned}$$

B.4 Case $l = 4$

When there are four descriptions received, all five audio channel signals are determined using the first four layer descriptions $\mathcal{L}_{1,n}^d$, $\mathcal{L}_{2,n}^d$, $\mathcal{L}_{3,n}^d$, and $\mathcal{L}_{4,n}^d$. The estimator matrix $\mathbf{H}(4)$ is given as follows:

$$\mathbf{H}(4) = \begin{bmatrix} h_{11}(4) & 1/2 & 1/2 & h_{14}(4) & * \\ h_{21}(4) & 1/2 & -1/2 & h_{24}(4) & * \\ 1 & -1 & h_{33}(4) & h_{34}(4) & * \\ h_{41}(4) & h_{42}(4) & h_{43}(4) & h_{44}^{(4)} & * \\ h_{51}(4) & h_{52}(4) & h_{53}(4) & h_{54}^{(4)} & * \end{bmatrix}. \quad (\text{B.11})$$

In this case, the estimator again fully recovers $\mathbf{L}_n$, $\mathbf{R}_n$, and $\mathbf{C}_n$, and determines the estimates of the surround channels $\widehat{\mathbf{L}}\mathbf{s}_n$ and $\widehat{\mathbf{R}}\mathbf{s}_n$ using one of the three solutions identified for the elements of $\mathbf{H}(4)$. Table B.3 shows the dependence of the solutions considered on the received layer descriptions.

Layer	Solution		
	1	2	3
$\mathcal{L}_{1,n}^d$			✓
$\mathcal{L}_{2,n}^d$		✓	✓
$\mathcal{L}_{3,n}^d$		✓	✓
$\mathcal{L}_{4,n}^d$	✓	✓	✓

Table B.3: Layers used in the estimation of $\widehat{\mathbf{L}}\mathbf{s}_n$ and $\widehat{\mathbf{R}}\mathbf{s}_n$.

Elements of $\mathbf{H}(4)$ for the three solutions considered are listed below.

Solution-I:

$$\begin{aligned}
 & h_{11}^I(4) = h_{21}^I(4) = h_{41}^I(4) = h_{51}^I(4) = h_{42}^I(4) = h_{52}^I(4) = h_{33}^I(4) = h_{43}^I(4) = h_{53}^I(4) = \\
 & h_{14}^I(4) = h_{24}^I(4) = h_{34}^I(4) = 0, \\
 & h_{31}^I(4) = 1 = -h_{32}^I(4), \\
 & h_{12}^I(4) = h_{22}^I(4) = h_{13}^I(4) = \frac{1}{2} = -h_{23}^I(4), \\
 & h_{i4}^I(4) = \sum_{l=4}^5 \rho_{il} / \sum_{k=4}^5 \sum_{j=4}^5 \rho_{jk}, \quad i = 4, 5.
 \end{aligned} \tag{B.12}$$

Solution-II:

$$\begin{aligned}
 & h_{11}^{II}(4) = h_{21}^{II}(4) = h_{41}^{II}(4) = h_{51}^{II}(4) = h_{33}^{II}(4) = h_{14}^{II}(4) = h_{24}^{II}(4) = h_{34}^{II}(4) = 0, \\
 & h_{31}^{II}(4) = 1 = -h_{32}^{II}(4), \\
 & h_{12}^{II}(4) = h_{22}^{II}(4) = h_{13}^{II}(4) = \frac{1}{2} = -h_{23}^{II}(4), \\
 & h_{42}^{II}(4) = h_{43}^{II}(4) = \frac{1}{2} \left(\frac{\rho_{41}(\sum_{k=4}^5 \sum_{j=4}^5 \rho_{kj}) - (\sum_{l=4}^5 \rho_{4l})(\sum_{m=4}^5 \rho_{m1})}{\rho_{11}(\sum_{k=4}^5 \sum_{j=4}^5 \rho_{kj}) - (\sum_{m=4}^5 \rho_{m1})^2} \right), \\
 & h_{52}^{II}(4) = \frac{1}{2} \left(\frac{\rho_{52}(\sum_{k=4}^5 \sum_{j=4}^5 \rho_{kj}) - (\sum_{l=4}^5 \rho_{5l})(\sum_{m=4}^5 \rho_{m2})}{\rho_{22}(\sum_{k=4}^5 \sum_{j=4}^5 \rho_{kj}) - (\sum_{m=4}^5 \rho_{m2})^2} \right) = -h_{53}^{II}(4),
 \end{aligned} \tag{B.13}$$

$$h_{44}^{II}(4) = \left(\frac{\rho_{41}}{\rho_{11}} - 2h_{42}^{II}(4) \right) \frac{\rho_{11}}{\sum_{k=4}^5 \rho_{k1}},$$

$$h_{54}^{II}(4) = \left(\frac{\rho_{52}}{\rho_{22}} - 2h_{52}^{II}(4) \right) \frac{\rho_{22}}{\sum_{k=4}^5 \rho_{k2}}.$$

Solution-III:

(B.14)

$$h_{11}^{III}(4) = h_{21}^{III}(4) = h_{33}^{III}(4) = h_{43}^{III}(4) = h_{53}^{III}(4) = h_{14}^{III}(4) = h_{24}^{III}(4) = h_{34}^{III}(4) = 0,$$

$$h_{31}^{III}(4) = 1 = -h_{32}^{III}(4),$$

$$h_{12}^{III}(4) = h_{22}^{III}(4) = h_{13}^{III}(4) = \frac{1}{2} = -h_{23}^{III}(4),$$

$$h_{i1}^{IV}(4) = \frac{\rho_{i3}(\sum_{k=4}^5 \sum_{j=4}^5 \rho_{kj}) - (\sum_{l=4}^5 \rho_{il})(\sum_{m=4}^5 \rho_{m3})}{\rho_{33}(\sum_{k=4}^5 \sum_{j=4}^5 \rho_{kj}) - (\sum_{m=4}^5 \rho_{m3})^2} = -h_{i2}^{III}(4), \quad i = 4, 5,$$

$$h_{i4}^{III}(4) = \left(h_{i2}^{III}(4) + \frac{\rho_{i3}}{\rho_{33}} \right) \frac{\rho_{33}}{\sum_{k=4}^5 \rho_{k3}}, \quad i = 4, 5.$$

B.5 Case $l = 5$

When all five descriptions are received, the estimator fully recovers all audio channels signals using the estimator matrix $\mathbf{H}(5)$:

$$\mathbf{H}(5) = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 1/2 & -1/2 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & -1/2 \end{bmatrix}. \quad (\text{B.15})$$

Appendix C

Sound Test Database

This Appendix lists the DVD sources used to extract the sound files that formed the sound database. The DVD sources are as follows:

- Titanic (DVD Video, 1997)
Starring: Leonardo DiCaprio, Kate Winslet, Director: James Cameron
- Shakespeare in Love (DVD Video, 1999)
Starring: Joseph Fiennes, Geoffrey Rush, Director: John Madden
- The Lion King (DVD Video, 1994)
Starring: Jonathan Taylor Thomas, Matthew Broderick, Director: Roger Allers, Rob Minkoff
- The Lion King 2 - Simba's Pride (DVD Video, 2004)
Starring: Matthew Broderick, Moira Kelly, Director: Darrell Rooney, Rob La-Duca
- Legends of the Fall (DVD Video, 1995)
Starring: Brad Pitt, Anthony Hopkins, Director: Edward Zwick
- Evita (DVD Video, 1997)
Starring: Madonna, Antonio Banderas, Director: Alan Parker

- Dances with Wolves (DVD Video, 1990)
Starring: Kevin Costner, Mary McDonnell, Director: Kevin Costner
- Zubin Mehta - Orff: Carmina Burana (DVD Audio, 2001)
Carl Orff (Composer), Zubin Mehta (Conductor), London Philharmonic Orchestra (Orchestra)
- We are not Angels 2, (DVD Video, 2005)
Starring: Nikola Kojo, Mirka Vasiljević, Uros Djurić, Srdjan Todorović, Director: Srdjan Dragojević
- Diana Krall - Live at the Montreal Jazz Festival (DVD Video, 2004)
Starring: Diana Krall
- Finding Nemo (DVD Video, 2003)
Starring: Albert Brooks, Ellen DeGeneres, Director: Andrew Stanton, Lee Unkrich

Bibliography

- [1] A. Watson, "Assessing the Quality of Audio and Video Components in Desktop Multimedia Conferencing", PhD dissertation, University College London, London, May 1999.
- [2] V.K. Goyal, "Beyond traditional transform coding", Ph.D. dissertation, Univ. Calif., Berkeley, 1998.
- [3] A. Leon-Garcia, I. Widjaja, "Communication Networks, Fundamental Concepts and Key Architectures", Second Edition, McGraw-Hill, 2004.
- [4] "Understanding TCP/IP", Cisco Systems Inc. Documentation, 2002.
- [5] V.K. Goyal, "Multiple Description Coding: Compression meets the network", IEEE Signal Processing Magazine, vol. 18, no. 5, pp. 74-93, September 2001.
- [6] J. Bolot, "End-to-end packet delay and loss behaviour in the Internet", Proceedings of ACM SIGCOMM '93, September 1993.
- [7] V. Paxson, "End-to-end Internet packet dynamics", Proceedings of ACM SIGCOMM '97, September 1997.
- [8] J. Sommers, P. Barford, N.G. Duffield, and A. Ron, "Improving Accuracy in End-to-End Packet Loss Measurements", Proceedings ACM SIGCOMM, August 2005.
- [9] <http://www.internettrafficreport.com/>

- [10] <http://weather.nacs.uci.edu/>
- [11] V.A. Vaishampayan, "Design of multiple description scalar quantizers", IEEE Trans. Inform. Theory, vol. 39, pp. 821–834, May 1993.
- [12] A. Gersho, R.M. Gray, "Vector Quantization and Signal Compression", Kluwer Academic Publishers, 1992.
- [13] S.D. Servetto, V.A. Vaishampayan, N.J.A. Sloane, "Multiple description lattice vector quantization", Proc. IEEE Data Compression Conf., Snowbird, UT, March 1999.
- [14] G. Schuller, J. Kovačević, F. Masson, and V.K. Goyal, "Robust low delay audio coding using multiple descriptions", IEEE Trans. Speech Audio Processing, vol. 13, no. 5, pp. 1014–1024, September 2005.
- [15] J. Ostergaard, O.A. Niamut, J. Jensen, R. Heusdens, "Perceptual Audio Coding using N-Channel Lattice Vector Quantization", Proc. IEEE International Conference on Audio, Speech and Signal Processing (ICASSP'06), May 2006.
- [16] V.K. Goyal "Theoretical foundations of transform coding", IEEE Signal Processing Mag., vol. 18, pp. 9–21, September 2001.
- [17] Y. Wang, M. T. Orchard, and A. R. Reibman, "Multiple description image coding for noisy channels by pairing transform coefficients", Proc. IEEE Workshop on Multimedia Signal Processing, Princeton, NJ, pp.419–424, June 1997.
- [18] Y. Wang, M.T. Orchard, V. Vaishampayan and A.R. Reibman, "Multiple description coding using pairwise correlating transforms", IEEE Transactions on Image Processing, vol. 10, no. 3, pp. 351–366, March 2001.

- [19] J. Kovačević and V.K. Goyal, "Generalized multiple description coding with correlating transforms", *IEEE Trans. Inform. Theory*, vol. 47, no. 6, September 2001.
- [20] C. Bauer, C. Cheng, "Improved methods for correlating transforms in multiple description coding systems", *Proc. of ICME 2006*, Toronto, July 2006.
- [21] M. Bosi, R.E. Goldberg, "Introduction to Digital Audio Coding and Standard", Kluwer Academic Publisher, 2003.
- [22] D.T. Yang, C. Kyriakakis, C-C.J. Kuo, "High Fidelity Multichannel Audio Coding", Hindawi Publishing Corporation, 2005.
- [23] <http://en.wikipedia.org/wiki/DVD>
- [24] http://www.mpeg.org/MPEG/DVD/Book_B/Video.html
- [25] T. Painter, A. Spanias, "Perceptual Coding of Digital Audio", *Proc. IEEE*, April 2000.
- [26] R. Arean, J. Kovačević and V.K. Goyal, "Multiple description perceptual audio coding with correlating transforms", *IEEE Trans. Speech Audio Processing*, vol. 8, no. 2, pp. 140–145, March 2000.
- [27] J.A. Kelner, V.K. Goyal and J. Kovačević, "Multiple description lattice vector quantization: Variations and extensions", *Proc. IEEE Data Compression Conf.*, pp. 480–489, Snowbird, Utah, March 2000.
- [28] V.A. Vaishampayan, N.J.A. Sloane and S.D. Servetto, "Multiple description vector quantization with lattice codebooks: design and analysis", *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1718–1734, July 2001.

- [29] C.I. Cheng, W. Jiang, "Multiple description error mitigation techniques for streaming compressed audio over a 802.11 wireless network", AES 120th Convention, May 2006.
- [30] <http://www.castudio.org/dvdaudioextractor/>
- [31] P.A. Chou, H.J. Wang and V.N. Padmanabhan, "Layered multiple description coding", Packet Video Workshop, Nantes, France, April 2003.
- [32] V. Stanković, R. Hamzaoui and Z. Xiong, "Robust Layered Multiple Description Coding of Scalable Media Data for Multicast", IEEE Signal Processing Letters, vol. 12, no. 2, pp. 154–157, February 2005.
- [33] A. Albanese, J. Blömer, J. Edmonds, M. Luby and M. Sudan, "Priority Encoding Transmission", IEEE Trans. Information Theory, vol. 42, pp. 1737–1744, November 1996.
- [34] B. Sklar, *Digital Communications: Fundamentals and Applications*, Second Edition, Upper Saddle River, NJ, Prentice-Hall, 2001.
- [35] Audio Device Technologies for Windows, Supporting Audio Formats, Multiple Channel Audio Data and WAVE Files, <http://www.microsoft.com/>
- [36] "5.1-Channel Music Production Guidelines", Issue 3, Dolby Laboratories Inc., 2005.
- [37] J.G. Proakis and D.G. Manolakis, *Digital signal processing, Principles, Algorithms, and Applications*, Prentice Hall, third edition, 1996.
- [38] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw Hill, 1965.
- [39] K. Umapathy, "Time-frequency modelling of wideband audio and speech signals", M.Phil. Thesis, The University of Hertfordshire and Ryerson University, 2002.