LIBRARY OF CONGRESS

Office of Business Enterprises
Duplication Services Section

THIS IS TO CERTIFY that the collections of the Library of Congress contain a bound volume entitled **AUDIO SIGNAL PROCESSING AND CODING,** call number TK 5102.92.S73 2007, Copy 1. The attached – Cover Page, Title Page, Copyright Page, Table of Contents Pages, Chapter 1 and Chapter 10 - are a true and complete representation from that work.

THIS IS TO CERTIFY FURTHER, that work is marked with a Library of Congress Cataloging-in-Publication stamp dated March 5, 2007.

IN WITNESS WHEREOF, the seal of the Library of Congress is affixed hereto on May 18, 2018.

Deirdre Scott

Deirdre Scott
Business Enterprises Officer
Office of Business Enterprises
Library of Congress

WILEY

Audio Signal Processing and Coding

Andreas Spanias, Ted Painter, and Venkatraman Atti

AUDIO SIGNAL PROCESSING AND CODING

Andreas Spanias
Ted Painter
Venkatraman Atti

WILEY-INTERSCIENCE
A John Wiley & Sons, Inc., Publication

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Wiley Bicentennial Logo: Richard J. Pacifico

Library of Congress Cataloging-in-Publication Data:

Spanias, Andreas.
Audio signal processing and coding/by Andreas Spanias, Ted Painter, Venkatraman Atti.
p. cm.
"Wiley-Interscience publication."
Includes bibliographical references and index.
ISBN: 978-0-471-79147-8
1. Coding theory. 2. Signal processing–Digital techniques. 3. Sound–Recording and reproducing–Digital techniques. I. Painter, Ted, 1967- II. Atti, Venkatraman, 1978- III. Title.

TK5102.92.S73 2006
621.382'8–dc22

2006040507

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CONTENTS

PREFACE xv

1 INTRODUCTION 1

1.1 Historical Perspective 1
1.2 A General Perceptual Audio Coding Architecture 4
1.3 Audio Coder Attributes 5
1.3.1 Audio Quality 6
1.3.2 Bit Rates 6
1.3.3 Complexity 6
1.3.4 Codec Delay 7
1.3.5 Error Robustness 7
1.4 Types of Audio Coders – An Overview 7
1.5 Organization of the Book 8
1.6 Notational Conventions 9
Problems 11
Computer Exercises 11

2 SIGNAL PROCESSING ESSENTIALS 13

2.1 Introduction 13
2.2 Spectra of Analog Signals 13
2.3 Review of Convolution and Filtering 16
2.4 Uniform Sampling 17
2.5 Discrete-Time Signal Processing 20

2.5.1 Transforms for Discrete-Time Signals 20
2.5.2 The Discrete and the Fast Fourier Transform 22
2.5.3 The Discrete Cosine Transform 23
2.5.4 The Short-Time Fourier Transform 23
2.6 Difference Equations and Digital Filters 25
2.7 The Transfer and the Frequency Response Functions 27
2.7.1 Poles, Zeros, and Frequency Response 29
2.7.2 Examples of Digital Filters for Audio Applications 30
2.8 Review of Multirate Signal Processing 33
2.8.1 Down-sampling by an Integer 33
2.8.2 Up-sampling by an Integer 35
2.8.3 Sampling Rate Changes by Noninteger Factors 36
2.8.4 Quadrature Mirror Filter Banks 36
2.9 Discrete-Time Random Signals 39
2.9.1 Random Signals Processed by LTI Digital Filters 42
2.9.2 Autocorrelation Estimation from Finite-Length Data 44
2.10 Summary 44
Problems 45
Computer Exercises 47

3 QUANTIZATION AND ENTROPY CODING 51

3.1 Introduction 51
3.1.1 The Quantization–Bit Allocation–Entropy Coding Module 52
3.2 Density Functions and Quantization 53
3.3 Scalar Quantization 54
3.3.1 Uniform Quantization 54
3.3.2 Nonuniform Quantization 57
3.3.3 Differential PCM 59
3.4 Vector Quantization 62
3.4.1 Structured VQ 64
3.4.2 Split-VQ 67
3.4.3 Conjugate-Structure VQ 69
3.5 Bit-Allocation Algorithms 70
3.6 Entropy Coding 74
3.6.1 Huffman Coding 77
3.6.2 Rice Coding 81
3.6.3 Golomb Coding 82

3.6.4 Arithmetic Coding 83
3.7 Summary 85
Problems 85
Computer Exercises 86

4 LINEAR PREDICTION IN NARROWBAND AND WIDEBAND CODING 91

4.1 Introduction 91
4.2 LP-Based Source-System Modeling for Speech 92
4.3 Short-Term Linear Prediction 94
4.3.1 Long-Term Prediction 95
4.3.2 ADPCM Using Linear Prediction 96
4.4 Open-Loop Analysis-Synthesis Linear Prediction 96
4.5 Analysis-by-Synthesis Linear Prediction 97
4.5.1 Code-Excited Linear Prediction Algorithms 100
4.6 Linear Prediction in Wideband Coding 102
4.6.1 Wideband Speech Coding 102
4.6.2 Wideband Audio Coding 104
4.7 Summary 106
Problems 107
Computer Exercises 108

5 PSYCHOACOUSTIC PRINCIPLES 113

5.1 Introduction 113
5.2 Absolute Threshold of Hearing 114
5.3 Critical Bands 115
5.4 Simultaneous Masking, Masking Asymmetry, and the Spread of Masking 120
5.4.1 Noise-Masking-Tone 123
5.4.2 Tone-Masking-Noise 124
5.4.3 Noise-Masking-Noise 124
5.4.4 Asymmetry of Masking 124
5.4.5 The Spread of Masking 125
5.5 Nonsimultaneous Masking 127
5.6 Perceptual Entropy 128
5.7 Example Codec Perceptual Model: ISO/IEC 11172-3 (MPEG - 1) Psychoacoustic Model 1 130
5.7.1 Step 1: Spectral Analysis and SPL Normalization 131

5.7.2 Step 2: Identification of Tonal and Noise Maskers 131
5.7.3 Step 3: Decimation and Reorganization of Maskers 135
5.7.4 Step 4: Calculation of Individual Masking Thresholds 136
5.7.5 Step 5: Calculation of Global Masking Thresholds 138
5.8 Perceptual Bit Allocation 138
5.9 Summary 140
Problems 140
Computer Exercises 141

6 TIME-FREQUENCY ANALYSIS: FILTER BANKS AND TRANSFORMS 145

6.1 Introduction 145
6.2 Analysis-Synthesis Framework for M-band Filter Banks 146
6.3 Filter Banks for Audio Coding: Design Considerations 148
6.3.1 The Role of Time-Frequency Resolution in Masking Power Estimation 149
6.3.2 The Role of Frequency Resolution in Perceptual Bit Allocation 149
6.3.3 The Role of Time Resolution in Perceptual Bit Allocation 150
6.4 Quadrature Mirror and Conjugate Quadrature Filters 155
6.5 Tree-Structured QMF and CQF M-band Banks 156
6.6 Cosine Modulated "Pseudo QMF" M-band Banks 160
6.7 Cosine Modulated Perfect Reconstruction (PR) M-band Banks and the Modified Discrete Cosine Transform (MDCT) 163
6.7.1 Forward and Inverse MDCT 165
6.7.2 MDCT Window Design 165
6.7.3 Example MDCT Windows (Prototype FIR Filters) 167
6.8 Discrete Fourier and Discrete Cosine Transform 178
6.9 Pre-echo Distortion 180
6.10 Pre-echo Control Strategies 182
6.10.1 Bit Reservoir 182
6.10.2 Window Switching 182
6.10.3 Hybrid, Switched Filter Banks 184
6.10.4 Gain Modification 185
6.10.5 Temporal Noise Shaping 185
6.11 Summary 186
Problems 188
Computer Exercises 191

7 TRANSFORM CODERS 195

7.1 Introduction 195
7.2 Optimum Coding in the Frequency Domain 196
7.3 Perceptual Transform Coder 197
7.3.1 PXFM 198
7.3.2 SEPXFM 199
7.4 Brandenburg-Johnston Hybrid Coder 200
7.5 CNET Coders 201
7.5.1 CNET DFT Coder 201
7.5.2 CNET MDCT Coder 1 201
7.5.3 CNET MDCT Coder 2 202
7.6 Adaptive Spectral Entropy Coding 203
7.7 Differential Perceptual Audio Coder 204
7.8 DFT Noise Substitution 205
7.9 DCT with Vector Quantization 206
7.10 MDCT with Vector Quantization 207
7.11 Summary 208
Problems 208
Computer Exercises 210

8 SUBBAND CODERS 211

8.1 Introduction 211
8.1.1 Subband Algorithms 212
8.2 DWT and Discrete Wavelet Packet Transform (DWPT) 214
8.3 Adapted WP Algorithms 218
8.3.1 DWPT Coder with Globally Adapted Daubechies Analysis Wavelet 218
8.3.2 Scalable DWPT Coder with Adaptive Tree Structure 220
8.3.3 DWPT Coder with Globally Adapted General Analysis Wavelet 223
8.3.4 DWPT Coder with Adaptive Tree Structure and Locally Adapted Analysis Wavelet 223
8.3.5 DWPT Coder with Perceptually Optimized Synthesis Wavelets 224
8.4 Adapted Nonuniform Filter Banks 226
8.4.1 Switched Nonuniform Filter Bank Cascade 226
8.4.2 Frequency-Varying Modulated Lapped Transforms 227
8.5 Hybrid WP and Adapted WP/Sinusoidal Algorithms 227

8.5.1 Hybrid Sinusoidal/Classical DWPT Coder 228
8.5.2 Hybrid Sinusoidal/M-band DWPT Coder 229
8.5.3 Hybrid Sinusoidal/DWPT Coder with WP Tree Structure Adaptation (ARCO) 230
8.6 Subband Coding with Hybrid Filter Bank/CELP Algorithms 233
8.6.1 Hybrid Subband/CELP Algorithm for Low-Delay Applications 234
8.6.2 Hybrid Subband/CELP Algorithm for Low-Complexity Applications 235
8.7 Subband Coding with IIR Filter Banks 237
Problems 237
Computer Exercise 240

9 SINUSOIDAL CODERS 241

9.1 Introduction 241
9.2 The Sinusoidal Model 242
9.2.1 Sinusoidal Analysis and Parameter Tracking 242
9.2.2 Sinusoidal Synthesis and Parameter Interpolation 245
9.3 Analysis/Synthesis Audio Codec (ASAC) 247
9.3.1 ASAC Segmentation 248
9.3.2 ASAC Sinusoidal Analysis-by-Synthesis 248
9.3.3 ASAC Bit Allocation, Quantization, Encoding, and Scalability 248
9.4 Harmonic and Individual Lines Plus Noise Coder (HILN) 249
9.4.1 HILN Sinusoidal Analysis-by-Synthesis 250
9.4.2 HILN Bit Allocation, Quantization, Encoding, and Decoding 251
9.5 FM Synthesis 251
9.5.1 Principles of FM Synthesis 252
9.5.2 Perceptual Audio Coding Using an FM Synthesis Model 252
9.6 The Sines + Transients + Noise (STN) Model 254
9.7 Hybrid Sinusoidal Coders 255
9.7.1 Hybrid Sinusoidal-MDCT Algorithm 256
9.7.2 Hybrid Sinusoidal-Vocoder Algorithm 257
9.8 Summary 258
Problems 258
Computer Exercises 259

10 AUDIO CODING STANDARDS AND ALGORITHMS 263

10.1 Introduction 263
10.2 MIDI *Versus* Digital Audio 264
10.2.1 MIDI Synthesizer 264
10.2.2 General MIDI (GM) 266
10.2.3 MIDI Applications 266
10.3 Multichannel Surround Sound 267
10.3.1 The Evolution of Surround Sound 267
10.3.2 The Mono, the Stereo, and the Surround Sound Formats 268
10.3.3 The ITU-R BS.775 5.1-Channel Configuration 268
10.4 MPEG Audio Standards 270
10.4.1 MPEG-1 Audio (ISO/IEC 11172-3) 275
10.4.2 MPEG-2 BC/LSF (ISO/IEC-13818-3) 279
10.4.3 MPEG-2 NBC/AAC (ISO/IEC-13818-7) 283
10.4.4 MPEG-4 Audio (ISO/IEC 14496-3) 289
10.4.5 MPEG-7 Audio (ISO/IEC 15938-4) 309
10.4.6 MPEG-21 Framework (ISO/IEC-21000) 317
10.4.7 MPEG Surround and Spatial Audio Coding 319
10.5 Adaptive Transform Acoustic Coding (ATRAC) 319
10.6 Lucent Technologies PAC, EPAC, and MPAC 321
10.6.1 Perceptual Audio Coder (PAC) 321
10.6.2 Enhanced PAC (EPAC) 323
10.6.3 Multichannel PAC (MPAC) 323
10.7 Dolby Audio Coding Standards 325
10.7.1 Dolby AC-2, AC-2A 325
10.7.2 Dolby AC-3/Dolby Digital/Dolby SR · D 327
10.8 Audio Processing Technology APT-x100 335
10.9 DTS – Coherent Acoustics 338
10.9.1 Framing and Subband Analysis 338
10.9.2 Psychoacoustic Analysis 339
10.9.3 ADPCM – Differential Subband Coding 339
10.9.4 Bit Allocation, Quantization, and Multiplexing 341
10.9.5 DTS-CA Versus Dolby Digital 342
Problems 342
Computer Exercise 342

11 LOSSLESS AUDIO CODING AND DIGITAL WATERMARKING 343

11.1 Introduction 343

11.2 Lossless Audio Coding (L^2AC) 344
11.2.1 L^2AC Principles 345
11.2.2 L^2AC Algorithms 346
11.3 DVD-Audio 356
11.3.1 Meridian Lossless Packing (MLP) 358
11.4 Super-Audio CD (SACD) 358
11.4.1 SACD Storage Format 362
11.4.2 Sigma-Delta Modulators (SDM) 362
11.4.3 Direct Stream Digital (DSD) Encoding 364
11.5 Digital Audio Watermarking 368
11.5.1 Background 370
11.5.2 A Generic Architecture for DAW 374
11.5.3 DAW Schemes – Attributes 377
11.6 Summary of Commercial Applications 378
Problems 382
Computer Exercise 382

12 QUALITY MEASURES FOR PERCEPTUAL AUDIO CODING 383

12.1 Introduction 383
12.2 Subjective Quality Measures 384
12.3 Confounding Factors in Subjective Evaluations 386
12.4 Subjective Evaluations of Two-Channel Standardized Codecs 387
12.5 Subjective Evaluations of 5.1-Channel Standardized Codecs 388
12.6 Subjective Evaluations Using Perceptual Measurement Systems 389
12.6.1 CIR Perceptual Measurement Schemes 390
12.6.2 NSE Perceptual Measurement Schemes 390
12.7 Algorithms for Perceptual Measurement 391
12.7.1 Example: Perceptual Audio Quality Measure (PAQM) 392
12.7.2 Example: Noise-to-Mask Ratio (NMR) 396
12.7.3 Example: Objective Audio Signal Evaluation (OASE) 399
12.8 ITU-R BS.1387 and ITU-T P.861: Standards for Perceptual Quality Measurement 401
12.9 Research Directions for Perceptual Codec Quality Measures 402

REFERENCES 405

INDEX 459

CHAPTER 1

INTRODUCTION

Audio coding or *audio compression* algorithms are used to obtain compact digital representations of high-fidelity (wideband) audio signals for the purpose of efficient transmission or storage. The central objective in audio coding is to represent the signal with a minimum number of bits while achieving transparent signal reproduction, i.e., generating output audio that cannot be distinguished from the original input, even by a sensitive listener ("golden ears"). This text gives an in-depth treatment of algorithms and standards for transparent coding of high-fidelity audio.

1.1 HISTORICAL PERSPECTIVE

The introduction of the compact disc (CD) in the early 1980s brought to the fore all of the advantages of digital audio representation, including true high-fidelity, dynamic range, and robustness. These advantages, however, came at the expense of high data rates. Conventional CD and digital audio tape (DAT) systems are typically sampled at either 44.1 or 48 kHz using pulse code modulation (PCM) with a 16-bit sample resolution. This results in uncompressed data rates of 705.6/768 kb/s for a monaural channel, or 1.41/1.54 Mb/s for a stereo-pair. Although these data rates were accommodated successfully in first-generation CD and DAT players, second-generation audio players and wirelessly connected systems are often subject to bandwidth constraints that are incompatible with high data rates. Because of the success enjoyed by the first-generation

Audio Signal Processing and Coding, by Andreas Spanias, Ted Painter, and Venkatraman Atti

systems, however, end users have come to expect "CD-quality" audio reproduction from any digital system. Therefore, new network and wireless multimedia digital audio systems must reduce data rates without compromising reproduction quality. Motivated by the need for compression algorithms that can satisfy simultaneously the conflicting demands of high compression ratios and transparent quality for high-fidelity audio signals, several coding methodologies have been established over the last two decades. Audio compression schemes, in general, employ design techniques that exploit both *perceptual irrelevancies* and *statistical redundancies*.

PCM was the primary audio encoding scheme employed until the early 1980s. PCM does not provide any mechanisms for redundancy removal. Quantization methods that exploit the signal correlation, such as differential PCM (DPCM), delta modulation [Jaya76] [Jaya84], and adaptive DPCM (ADPCM) were applied to audio compression later (e.g., PC audio cards). Owing to the need for drastic reduction in bit rates, researchers began to pursue new approaches for audio coding based on the *principles of psychoacoustics* [Zwic90] [Moor03]. Psychoacoustic notions in conjunction with the basic properties of signal quantization have led to the theory of *perceptual entropy* [John88a] [John88b]. Perceptual entropy is a quantitative estimate of the fundamental limit of transparent audio signal compression. Another key contribution to the field was the characterization of the auditory filter bank and particularly the time-frequency analysis capabilities of the inner ear [Moor83]. Over the years, several *filter-bank* structures that mimic the critical band structure of the auditory filter bank have been proposed. A filter bank is a parallel bank of bandpass filters covering the audio spectrum, which, when used in conjunction with a perceptual model, can play an important role in the identification of perceptual irrelevancies.

During the early 1990s, several workgroups and organizations such as the International Organization for Standardization/International Electro-technical Commission (ISO/IEC), the International Telecommunications Union (ITU), AT&T, Dolby Laboratories, Digital Theatre Systems (DTS), Lucent Technologies, Philips, and Sony were actively involved in developing perceptual audio coding algorithms and standards. Some of the popular commercial standards published in the early 1990s include Dolby's Audio Coder-3 (AC-3), the DTS Coherent Acoustics (DTS-CA), Lucent Technologies' Perceptual Audio Coder (PAC), Philips' Precision Adaptive Subband Coding (PASC), and Sony's Adaptive Transform Acoustic Coding (ATRAC). Table 1.1 lists chronologically some of the prominent audio coding standards. The commercial success enjoyed by these audio coding standards triggered the launch of several multimedia storage formats.

Table 1.2 lists some of the popular multimedia storage formats since the beginning of the CD era. High-performance stereo systems became quite common with the advent of CDs in the early 1980s. A compact-disc–read only memory (CD-ROM) can store data up to 700–800 MB in digital form as "microscopic-pits" that can be read by a laser beam off of a reflective surface or a medium. Three competing storage media – DAT, the digital compact cassette (DCC), and the

Table 1.1. List of perceptual and lossless audio coding standards/algorithms.

Standard/algorithm	Related references
1. ISO/IEC MPEG-1 audio	[ISOI92]
2. Philips' PASC (for DCC applications)	[Lokh92]
3. AT&T/Lucent PAC/EPAC	[John96c] [Sinh96]
4. Dolby AC-2	[Davi92] [Fiel91]
5. AC-3/Dolby Digital	[Davis93] [Fiel96]
6. ISO/IEC MPEG-2 (BC/LSF) audio	[ISOI94a]
7. Sony's ATRAC; (MiniDisc and SDDS)	[Yosh94] [Tsut96]
8. SHORTEN	[Robi94]
9. Audio processing technology – APT-x100	[Wyli96b]
10. ISO/IEC MPEG-2 AAC	[ISOI96]
11. DTS coherent acoustics	[Smyt96] [Smyt99]
12. The DVD Algorithm	[Crav96] [Crav97]
13. MUSICompress	[Wege97]
14. Lossless transform coding of audio (LTAC)	[Pura97]
15. AudioPaK	[Hans98b] [Hans01]
16. ISO/IEC MPEG-4 audio version 1	[ISOI99]
17. Meridian lossless packing (MLP)	[Gerz99]
18. ISO/IEC MPEG-4 audio version 2	[ISOI00]
19. Audio coding based on integer transforms	[Geig01] [Geig02]
20. Direct-stream digital (DSD) technology	[Reef01a] [Jans03]

Table 1.2. Some of the popular audio storage formats.

Audio storage format	Related references
1. Compact disc	[CD82] [IECA87]
2. Digital audio tape (DAT)	[Watk88] [Tan89]
3. Digital compact cassette (DCC)	[Lokh91] [Lokh92]
4. MiniDisc	[Yosh94] [Tsut96]
5. Digital versatile disc (DVD)	[DVD96]
6. DVD-audio (DVD-A)	[DVD01]
7. Super audio CD (SACD)	[SACD02]

MiniDisc (MD) – entered the commercial market during 1987–1992. Intended mainly for back-up high-density storage (~1.3 GB), the DAT became the primary source of mass data storage/transfer [Watk88] [Tan89]. In 1991–1992, Sony proposed a storage medium called the MiniDisc, primarily for audio storage. MD employs the ATRAC algorithm for compression. In 1991, Philips introduced the DCC, a successor of the analog compact cassette. Philips DCC employs a compression scheme called the PASC [Lokh91] [Lokh92] [Hoog94]. The DCC began

as a potential competitor for DATs but was discontinued in 1996. The introduction of the digital versatile disc (DVD) in 1996 enabled both video and audio recording/storage as well as text-message programming. The DVD became one of the most successful storage media. With the improvements in the audio compression and DVD storage technologies, multichannel surround sound encoding formats gained interest [Bosi93] [Holm99] [Bosi00].

With the emergence of streaming audio applications, during the late 1990s, researchers pursued techniques such as combined speech and audio architectures, as well as joint source-channel coding algorithms that are optimized for the packet-switched Internet. The advent of ISO/IEC MPEG-4 standard (1996–2000) [ISOI99] [ISOI00] established new research goals for high-quality coding of audio at low bit rates. MPEG-4 audio encompasses more functionality than perceptual coding [Koen98] [Koen99]. It comprises an integrated family of algorithms with provisions for scalable, object-based speech and audio coding at bit rates from as low as 200 b/s up to 64 kb/s per channel.

The emergence of the DVD-audio and the super audio CD (SACD) provided designers with additional storage capacity, which motivated research in *lossless* audio coding [Crav96] [Gerz99] [Reef01a]. A lossless audio coding system is able to reconstruct perfectly a bit-for-bit representation of the original input audio. In contrast, a coding scheme incapable of perfect reconstruction is called *lossy*. For most audio program material, lossy schemes offer the advantage of lower bit rates (e.g., less than 1 bit per sample) relative to lossless schemes (e.g., 10 bits per sample). Delivering real-time lossless audio content to the network browser at low bit rates is the next grand challenge for codec designers.

1.2 A GENERAL PERCEPTUAL AUDIO CODING ARCHITECTURE

Over the last few years, researchers have proposed several efficient signal models (e.g., transform-based, subband-filter structures, wavelet-packet) and compression standards (Table 1.1) for high-quality digital audio reproduction. Most of these algorithms are based on the generic architecture shown in Figure 1.1.

The coders typically segment input signals into quasi-stationary frames ranging from 2 to 50 ms. Then, a time-frequency analysis section estimates the temporal and spectral components of each frame. The time-frequency mapping is usually matched to the analysis properties of the human auditory system. Either way, the ultimate objective is to extract from the input audio a set of time-frequency parameters that is amenable to quantization according to a *perceptual distortion metric*. Depending on the overall design objectives, the time-frequency analysis section usually contains one of the following:

- Unitary transform
- Time-invariant bank of critically sampled, uniform/nonuniform bandpass filters

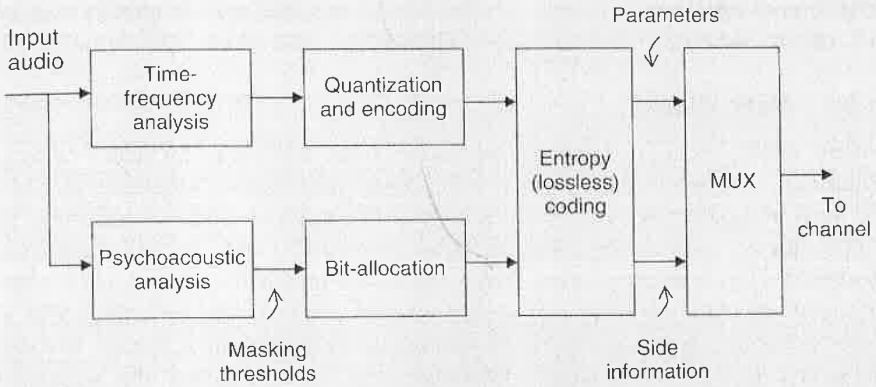


Figure 1.1. A generic perceptual audio encoder.

- Time-varying (signal-adaptive) bank of critically sampled, uniform/nonuniform bandpass filters
- Harmonic/sinusoidal analyzer
- Source-system analysis (LPC and multipulse excitation)
- Hybrid versions of the above.

The choice of time-frequency analysis methodology always involves a fundamental tradeoff between time and frequency resolution requirements. Perceptual distortion control is achieved by a psychoacoustic signal analysis section that estimates signal masking power based on psychoacoustic principles. The psychoacoustic model delivers masking thresholds that quantify the maximum amount of distortion at each point in the time-frequency plane such that quantization of the time-frequency parameters does not introduce audible artifacts. The psychoacoustic model therefore allows the quantization section to exploit perceptual irrelevancies. This section can also exploit statistical redundancies through classical techniques such as DPCM or ADPCM. Once a quantized compact parametric set has been formed, the remaining redundancies are typically removed through noiseless run-length (RL) and entropy coding techniques, e.g., Huffman [Cove91], arithmetic [Witt87], or Lempel-Ziv-Welch (LZW) [Ziv77] [Welc84]. Since the output of the psychoacoustic distortion control model is signal-dependent, most algorithms are inherently variable rate. Fixed channel rate requirements are usually satisfied through buffer feedback schemes, which often introduce encoding delays.

1.3 AUDIO CODER ATTRIBUTES

Perceptual audio coders are typically evaluated based on the following attributes: audio reproduction quality, operating bit rates, computational complexity, codec delay, and channel error robustness. The objective is to attain a high-quality (transparent) audio output at low bit rates (<32 kb/s), with an acceptable

algorithmic delay (~5 to 20 ms), and with low computational complexity (~1 to 10 million instructions per second, or MIPS).

1.3.1 Audio Quality

Audio quality is of paramount importance when designing an audio coding algorithm. Successful strides have been made since the development of simple near-transparent perceptual coders. Typically, classical objective measures of signal fidelity such as the signal to noise ratio (SNR) and the total harmonic distortion (THD) are inadequate [Ryde96]. As the field of perceptual audio coding matured rapidly and created greater demand for listening tests, there was a corresponding growth of interest in perceptual measurement schemes. Several subjective and objective quality measures have been proposed and standardized during the last decade. Some of these schemes include the noise-to-mask ratio (NMR, 1987) [Bran87a] the perceptual audio quality measure (PAQM, 1991) [Beer91], the perceptual evaluation (PERCEVAL, 1992) [Pail92], the perceptual objective measure (POM, 1995) [Colo95], and the objective audio signal evaluation (OASE, 1997) [Spor97]. We will address these and several other quality assessment schemes in detail in Chapter 12.

1.3.2 Bit Rates

From a codec designer's point of view, one of the key challenges is to represent high-fidelity audio with a minimum number of bits. For instance, if a 5-ms audio frame sampled at 48 kHz (240 samples per frame) is represented using 80 bits, then the encoding bit rate would be 80 bits/5 ms = 16 kb/s. Low bit rates imply high compression ratios and generally low reproduction quality. Early coders such as the ISO/IEC MPEG-1 (32–448 kb/s), the Dolby AC-3 (32–384 kb/s), the Sony ATRAC (256 kb/s), and the Philips PASC (192 kb/s) employ high bit rates for obtaining transparent audio reproduction. However, the development of several sophisticated audio coding tools (e.g., MPEG-4 audio tools) created ways for efficient transmission or storage of audio at rates between 8 and 32 kb/s. Future audio coding algorithms promise to offer reasonable quality at low rates along with the ability to scale both *rate* and *quality* to match different requirements such as time-varying channel capacity.

1.3.3 Complexity

Reduced computational complexity not only enables real-time implementation but may also decrease the power consumption and extend battery life. Computational complexity is usually measured in terms of millions of instructions per second (MIPS). Complexity estimates are processor-dependent. For example, the complexity associated with Dolby's AC-3 decoder was estimated at approximately 27 MIPS using the Zoran ZR38001 general-purpose DSP core [Vern95]; for the Motorola DSP56002 processor, the complexity was estimated at 45 MIPS [Vern95]. Usually, most of the audio codecs rely on the so-called asymmetric encoding principle. This means that the codec complexity is not evenly

shared between the encoder and the decoder (typically, encoder 80% and decoder 20% complexity), with more emphasis on reducing the decoder complexity.

1.3.4 Codec Delay

Many of the network applications for high-fidelity audio (streaming audio, audio-on-demand) are delay tolerant (up to 100–200 ms), providing the opportunity to exploit long-term signal properties in order to achieve high coding gain. However, in two-way real-time communication and voice-over Internet protocol (VoIP) applications, low-delay encoding (10–20 ms) is important. Consider the example described before, i.e., an audio coder operating on frames of 5 ms at a 48 kHz sampling frequency. In an ideal encoding scenario, the minimum amount of delay should be 5 ms at the encoder and 5 ms at the decoder (same as the frame length). However, other factors such as analysis-synthesis filter bank window, the look-ahead, the bit-reservoir, and the channel delay contribute to additional delays. Employing shorter analysis-synthesis windows, avoiding look-ahead, and re-structuring the bit-reservoir functions could result in low-delay encoding, nonetheless, with reduced coding efficiencies.

1.3.5 Error Robustness

The increasing popularity of streaming audio over packet-switched and wireless networks such as the Internet implies that any algorithm intended for such applications must be able to deal with a noisy time-varying channel. In particular, provisions for error robustness and error protection must be incorporated at the encoder in order to achieve reliable transmission of digital audio over error-prone channels. One simple idea could be to provide better protection to the error-sensitive and priority (important) bits. For instance, the audio frame header requires the maximum error robustness; otherwise, transmission errors in the header will seriously impair the entire audio frame. Several error detecting/correcting codes [Lin82] [Wick95] [Bayl97] [Swee02] [Zara02] can also be employed. Inclusion of error correcting codes in the bitstream might help to obtain error-free reproduction of the input audio, however, with increased complexity and bit rates.

From the discussion in the previous sections, it is evident that several tradeoffs must be considered in designing an algorithm for a particular application. For this reason, audio coding standards consist of several tools that enable the design of scalable algorithms. For example, MPEG-4 provides tools to design algorithms that satisfy a variety of bit rate, delay, complexity, and robustness requirements.

1.4 TYPES OF AUDIO CODERS – AN OVERVIEW

Based on the signal model or the analysis-synthesis technique employed to encode audio signals, audio coders can be broadly classified as follows:

- Linear predictive
- Transform

- Subband
- Sinusoidal.

Algorithms are also classified based on the lossy or the lossless nature of audio coding. Lossy audio coding schemes achieve compression by exploiting perceptually irrelevant information. Some examples of lossy audio coding schemes include the ISO/IEC MPEG codec series, the Dolby AC-3, and the DTS CA. In lossless audio coding, the audio data is merely "packed" to obtain a bit-for-bit representation of the original. The meridian lossless packing (MLP) [Gerz99] and the direct stream digital (DSD) techniques [Brue97] [Reef01a] form a class of high-end lossless compression algorithms that are embedded in the DVD-audio [DVD01] and the SACD [SACD02] storage formats, respectively. Lossless audio coding techniques, in general yield high-quality digital audio without any artifacts at high rates. For instance, perceptual audio coding yields compression ratios from 10:1 to 25:1, while lossless audio coding can achieve compression ratios from 2:1 to 4:1.

1.5 ORGANIZATION OF THE BOOK

This book is organized as follows. In Chapter 2, we review basic signal processing concepts associated with audio coding. Chapter 3 provides introductory material to waveform quantization and entropy coding schemes. Some of the key topics covered in this chapter include scalar quantization, uniform/nonuniform quantization, pulse code modulation (PCM), differential PCM (DPCM), adaptive DPCM (ADPCM), vector quantization (VQ), bit-allocation techniques, and entropy coding schemes (Huffman, Rice, and arithmetic).

Chapter 4 provides information on linear prediction and its application in narrow and wideband coding. First, we address the utility of LP analysis/synthesis approach in speech applications. Next, we describe the open-loop analysis-synthesis LP and closed-loop analysis-by-synthesis LP techniques.

In Chapter 5, psychoacoustic principles are described. Johnston's notion of perceptual entropy is presented as a measure of the fundamental limit of transparent compression for audio. The ISO/IEC 11172-3 MPEG-1 psychoacoustic analysis model 1 is used to describe the five important steps associated with the global masking threshold computation. Chapter 6 explores filter bank design issues and algorithms, with a particular emphasis placed on the modified discrete cosine transform (MDCT) that is widely used in several perceptual audio coding algorithms. Chapter 6 also addresses pre-echo artifacts and control strategies.

Chapters 7, 8, and 9 review established and emerging techniques for transparent coding of FM and CD-quality audio signals, including several algorithms that have become international standards. Transform coding methodologies are described in Chapter 7, subband coding algorithms are addressed in Chapter 8, and sinusoidal algorithms are presented in Chapter 9. In addition to methods based on uniform bandwidth filter banks, Chapter 8 covers coding methods that

utilize discrete wavelet transforms (DWT), discrete wavelet packet transforms (DWPT), and other nonuniform filter banks. Examples of hybrid algorithms that make use of more than one signal model appear throughout Chapters 7, 8, and 9.

Chapter 10 is concerned with standardization activities in audio coding. It describes coding standards and products such as the ISO/IEC MPEG family (−1 ".MP1/2/3", −2, −4, −7, and −21), the Sony Minidisc (ATRAC), the cinematic Sony SDDS, the Lucent Technologies PAC/EPAC/MPAC, the Dolby AC-2/AC-3, the Audio Processing Technology APT-x100, and the DTS-coherent acoustics (DTS-CA). Details on the MP3 and MPEG-4 AAC algorithms that are popular in Web and in handheld media applications, e.g., Apple iPod, are provided.

Chapter 11 focuses on lossless audio coding and digital audio watermarking techniques. In particular, the SHORTEN, the DVD algorithm, the MUSICompress, the AudioPaK, the C-LPAC, the LTAC, and the IntMDCT lossless coding schemes are described in detail. Chapter 11 also addresses the two popular high-end storage formats, i.e., the SACD and the DVD-Audio. The MLP and the DSD techniques for lossless audio coding are also presented.

Chapter 12 provides information on subjective quality measures for perceptual codecs. The five-point absolute and differential subjective quality scales are addressed, as well as the subjective test methodologies specified in the ITU-R Recommendation BS.1116. A set of subjective benchmarks is provided for the various standards in both stereophonic and multichannel modes to facilitate algorithm comparisons.

1.6 NOTATIONAL CONVENTIONS

Unless otherwise specified, bit rates correspond to single-channel or monaural coding throughout this text. Subjective quality measurements are specified in terms of either the five-point mean opinion score (MOS, Table 1.3) or the 41-point subjective difference grade (SDG, Chapter 12, Table 12.1). Table 1.4 lists some of the symbols and notation used in the book.

Table 1.3. Mean opinion score (MOS) scale.

MOS	Perceptual quality
1	Bad
2	Poor
3	Average
4	Good
5	Excellent

Table 1.4. Symbols and notation used in the book.

Symbol/notation	Description
t, n	Time index/sample index
ω, Ω	Frequency index (analog domain, discrete domain)
$f(=\omega/2\pi)$	Frequency (Hz)
F_s, T_s	Sampling frequency, sampling period
$x(t) \leftrightarrow X(\omega)$	Continuous-time Fourier transform (FT) pair
$x(n) \leftrightarrow X(\Omega)$	Discrete-time Fourier transform (DTFT) pair
$x(n) \leftrightarrow X(z)$	z transform pair
$s[.]$	Indicates a particular element in a coefficient array
$h(n) \leftrightarrow H(\Omega)$	Impulse-frequency response pair of a discrete time system
$e(n)$	Error/prediction residual
$H(z) = \frac{B(z)}{A(z)} = \frac{1 + b_1 z^{-1} + \ldots + b_L z^{-L}}{1 + a_1 z^{-1} + \ldots + a_M z^{-M}}$	Transfer function consisting of numerator-polynomial and denominator-polynomial (corresponding to b-coefficients and a-coefficients)
$\tilde{s}(n) = \sum_{i=0}^{M} a_i s(n-i)$	Predicted signal
$Q\langle s(n)\rangle = \hat{s}(n)$	Quantization/approximation operator or estimated/encoded value
$s^{[.]}$	Square brackets in the superscript denote recursion
$s^{(.)}$	Parenthesis superscript; time dependency
N, N_f, N_{sf}	Total number of samples, samples per frame, samples per subframe
log(.), ln(.), $\log_p(.)$	Log to the base-10, log to the base-e, log to the base-p
E[.]	Expectation operator
ε	Mean squared error (MSE)
μ_x, σ_x^2	Mean and the variance of the signal, $x(n)$
$r_{xx}(m)$	Autocorrelation of the signal, $x(n)$
$r_{xy}(m)$	Cross-correlation of $x(n)$ and $y(n)$
$R_{xx}(e^{j\Omega})$	Power spectral density of the signal, $x(n)$
Bit rate	Number of bits per second (b/s, kb/s, or Mb/s)
dB, SPL	Decibels, sound pressure level

PROBLEMS

The objective of these introductory problems are to introduce the novice to simple relations between the sampling rate and the bit rate in PCM coded sequences.

1.1. Consider an audio signal, $s(n)$, sampled at 44.1 kHz and digitized using a) 8-bit, b) 24-bit, and c) 32-bit resolution. Compute the data rates for the cases (a)–(c). Give the number of samples within a 16-ms frame and compute the number of bits per frame.

1.2. List some of the typical data rates (in kb/s) and sampling rates (in kHz) employed in applications such as a) video streaming, b) audio streaming, c) digital audio broadcasting, d) digital compact cassette, e) MiniDisc, f) DVD, g) DVD-audio, h) SACD, i) MP3, j) MP4, k) video conferencing, and l) cellular telephony.

COMPUTER EXERCISES

The objective of this exercise is to familiarize the reader with the handling of sound files using MATLAB and to expose the novice to perceptual attributes of sampling rate and bit resolution.

1.3. For this computer exercise, use MATLAB workspace ***ch1pb1.mat*** from the website.

Load the workspace ***ch1pb1.mat*** using.

```
>> load('ch1pb1.mat');
```

Use whos command to view the variables in the workspace. The data-vector 'audio_in' contains 44,100 samples of audio data. Perform the following in MATLAB:

```
>> wavwrite(audio_in,44100,16,'pb1_aud44_16.wav');
>> wavwrite(audio_in,10000,16,'pb1_aud10_16.wav');
>> wavwrite(audio_in,44100,8,'pb1_aud44_08.wav');
```

Listen to the wave files ***pb1_aud44_16.wav***, ***pb1_aud10_16.wav***, and ***pb1_aud44_08.wav*** using a media player. Comment on the perceptual quality of the three wave files.

1.4. Down-sample the data-vector 'audio_in' in problem 1.3 using

```
>> aud_down_4 = downsample(audio_in, 4);
```

Use the following commands to listen to audio_in and aud_down_4. Comment on the perceptual quality of the data vectors in each of the cases below:

```
>> sound(audio_in, fs);
>> sound(aud_down_4, fs);
>> sound(aud_down_4, fs/4);
```

CHAPTER 10

AUDIO CODING STANDARDS AND ALGORITHMS

10.1 INTRODUCTION

Despite the several advances, research towards developing lower rate coders for *stereophonic* and *multichannel surround sound* systems is strong in many industry and university labs. Multimedia applications such as online radio, web jukeboxes, and teleconferencing created a demand for audio coding algorithms that can deliver real-time wireless audio content. This will in turn require audio compression algorithms to deliver high-quality audio at *low bit-rates* with *resilience/robustness* to bit errors. Motivated by the need for audio compression *algorithms for streaming audio*, researchers pursue techniques such as combined speech/audio architectures, as well as joint source-channel coding algorithms that are optimized for the packet switched Internet [Ben99] [Liu99] [Gril02], Bluetooth [Joha01] [Chen04] [BWEB], and in some cases wideband cellular network [Ji02] [Toh03]. Also the need for transparent reproduction quality coding algorithms in *storage media* such as the super audio CD (SACD) and the DVD-audio provided designers with new challenges. There is in fact an ongoing debate over the *quality* limitations associated with lossy compression. Some experts believe that *uncompressed* digital CD-quality audio (44.1 kHz/16 bit) is inferior to the analog original. They contend that sample rates above 55 kHz and word lengths greater than 20 bits are necessary to achieve transparency in the absence of any compression.

As a result, several standards have been developed [ISOI92] [ISOI94a] [Davi94] [Fiel96] [Wyl96b] [ISOI97b], particularly in the last five years [Gerz99] [ISOI99]

Audio Signal Processing and Coding, by Andreas Spanias, Ted Painter, and Venkatraman Atti

[ISOI00] [ISOI01b] [ISOI02a] [Jans03] [Kuzu03], and several are now being deployed commercially. This chapter and the next address some of the important audio coding algorithms and standards deployed during the last decade. In particular, we describe the lossy audio compression (LAC) algorithms in this chapter, and the lossless audio coding (L^2AC) schemes in Chapter 11.

Some of the LAC schemes (Figure 10.1) described in this chapter include the ISO/MPEG codec series, the Sony ATRAC, the Lucent Technologies PAC/EPAC/MPAC, the Dolby AC-2/AC-3, the APT-x 100, and the DTS-coherent acoustics.

The rest of the chapter is organized as follows. Section 10.2 reviews the MIDI standard. Section 10.3 serves as an introduction to the multichannel surround sound format. Section 10.4 is dedicated to MPEG audio standards. In particular, Sections 10.4.1 through 10.4.6, respectively, describe the MPEG-1, MPEG-2 BC/LSF, MPEG-2 AAC, MPEG-4, MPEG-7, and MPEG-21 audio standards. Section 10.5 presents the adaptive transform acoustic coding (ATRAC) algorithm, the MiniDisc and the Sony dynamic digital sound (SDDS) systems. Section 10.6 reviews the Lucent Technologies perceptual audio coder (PAC), the enhanced PAC (EPAC), and the multichannel PAC (MPAC) coders. Section 10.7 describes the Dolby AC-2 and the AC-3/Dolby Digital algorithms. Section 10.8 is devoted to the Audio Processing Technology – APTx-100 system. Finally, in Section 10.9, we examine the principles of coherent acoustics in coding, that are embedded in the Digital Theater Systems–Coherent Acoustics (DTS-CA).

10.2 MIDI *VERSUS* DIGITAL AUDIO

The musical instrument digital interface (MIDI) encoding is an efficient way of extracting and representing semantic features from audio signals [Lehr93] [Penn95] [Hube98] [Whit00]. MIDI synthesizers, originally established in 1983, are widely used for musical transcriptions. Currently, the MIDI standards are governed by the MIDI Manufacturers Association (MMA) in collaboration with the Japanese Association of Musical Electronics Industry (AMEI).

The digital audio representation contains the actual sampled audio data, while a MIDI synthesizer represents only the instructions that are required to play the sounds. Therefore, the MIDI data files are extremely small when compared to the digital audio data files. Despite being able to represent high-quality stereo data at 10–30 kb/s, there are certain limitations with MIDI formats. In particular, the MIDI protocol uses a slow serial interface for data streaming at 31.25 kb/s [Foss95]. Moreover, MIDI is hardware dependent. Despite such limitations, musicians prefer the MIDI standard because of its simplicity and high-quality sound synthesis capability.

10.2.1 MIDI Synthesizer

A simple MIDI system (Figure 10.2) consists of a MIDI controller, a sequencer, and a MIDI sound module. The keyboard is an example of a MIDI controller

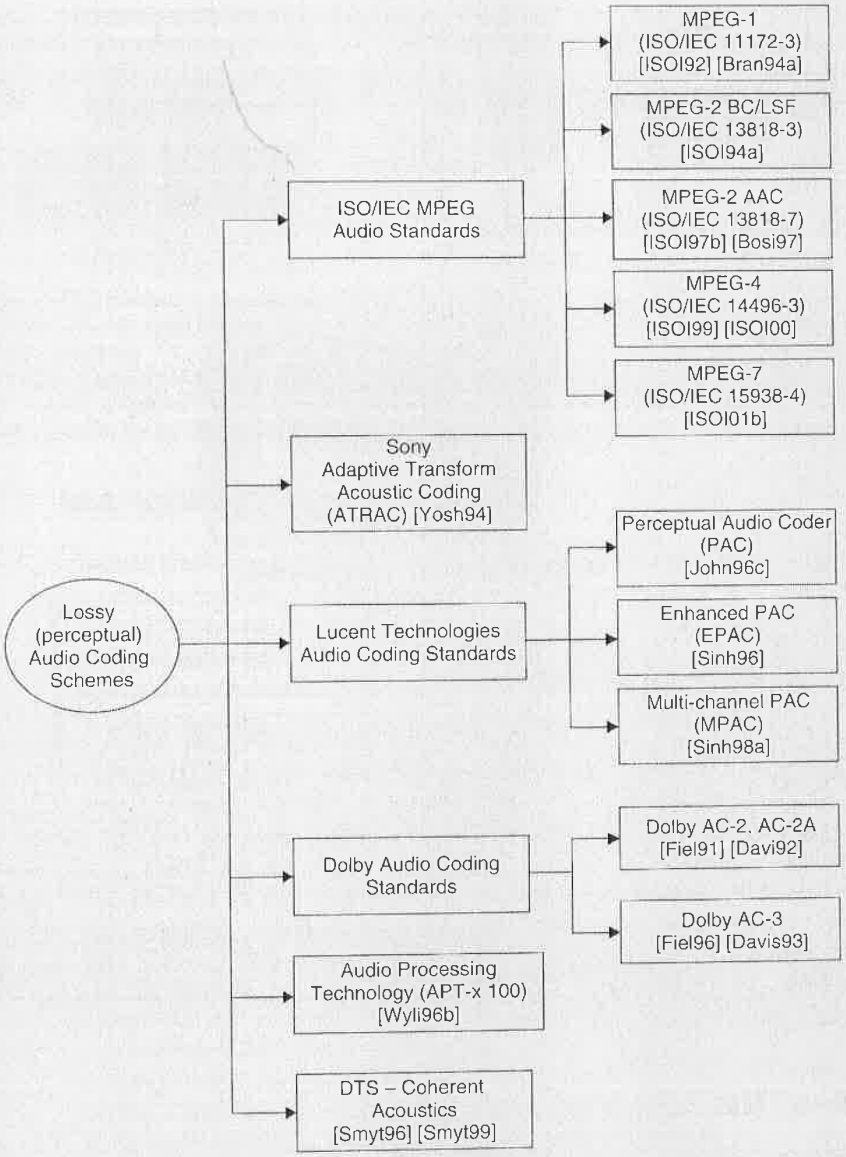


Figure 10.1. A list of some of the lossy audio coding algorithms.

that translates the music notes into a real-time MIDI data stream. The MIDI data stream includes a start bit, 8 data bits, and one stop bit. A MIDI sequencer captures the MIDI data sequence, and allows for various manipulations (e.g., editing, morphing, combining, etc.). On the other hand a MIDI sound module acts as a sound player.

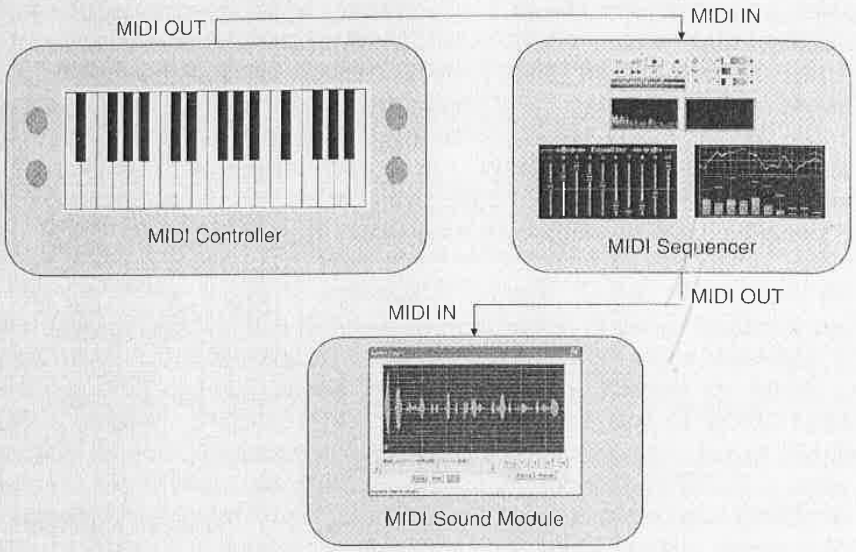


Figure 10.2. A simple MIDI system.

10.2.2 General MIDI (GM)

In order to facilitate a greater degree of file compatibility, the MMA developed the general MIDI (GM) standard. The GM constitutes a MIDI synthesizer with a standard set of voices (16 categories of 8 different sounds $= 16 \times 8 = 128$ sounds) that are fixed. Although the GM standard does not describe the sound quality of synthesizer outputs, it provides details on the MIDI compatibility, i.e., the MIDI sounds composed on one sequencer can be reproduced or played back on any other system with reduced or no distortion. Different GM versions are available in the market today, i.e., GM Level-1, GM Level-2, GM lite, and scalable polyphonic MIDI (SPMIDI). Table 10.1 summarizes the various GM levels and versions.

10.2.3 MIDI Applications

MIDI has been successful in a wide range of applications including music-retrieval and classification [Mana02], music databases search [Kost96], musical instrument control [MID03], MIDI karaoke players [MIDI], real-time object-based coding [Bros03], automatic recognition of musical phrases [Kost96], audio authoring [Mode98], waveform-editing [MIDI], singing voice synthesis [Maco97], loudspeaker design [Bald96], and feature extraction [Kost95]. The MPEG-4 structured audio tool incorporates many MIDI-like features. Other applications of MIDI are attributed to MIDI GM Level-2 [Mode00], XMIDI [LKur96], and PCM to MIDI transportation [Mart02] [MID03] [MIDI].

Table 10.1. General MIDI (GM) formats and versions.

GM specifications	GM L-1	GM L-2	GM Lite	SPMIDI
Number of MIDI channels	16	16	16	16
Percussion (drumming) channel	10	10, 11	10	–
Polyphony (voices)	24 voices	32 voices	Limited	–

Other related information [MID03]:
GM L-2: This is the latest standard introduced with capabilities of registered parameter controllers, MIDI tuning, universal system exclusive messages. GM L-2 is backwards compatible with GM L-1.
GM Lite: As the name implies, this is a light version of GM L-1 and is intended for devices with limited polyphony.
SPMIDI: Intended for mobile devices. SPMIDI, functions based on the fundamentals of GM Lite and scalable polyphony. This GM standard has been adopted by the Third-Generation Partnership Project (3GPP) for the multimedia messaging applications in cellular phones.

10.3 MULTICHANNEL SURROUND SOUND

Surround sound tracks (or channels) were included in motion pictures, in the early 1950s, in order to provide a more realistic cinema experience. Later, the popularity of surround sound resulted in its migration from cinema halls to home theaters equipped with matrixed multichannel sound (e.g., Dolby ProLogic™). This can be attributed to the multichannel surround sound format [Bosi93] [Holm99] [DOLBY] and subsequent improvements in the audio compression technology.

Until the early 1990s, almost all surround sound formats were based on matrixing, i.e., the information from all the channels (front and surround) was encoded as a two-channel stereo as shown in Figure 10.3. In the mid-1990s, discrete encoding, i.e., 5.1 separate channels of audio, was introduced by Dolby Laboratories and Digital Theater Systems (DTS).

10.3.1 The Evolution of Surround Sound

Table 10.2 lists some of the milestones in the history of multichannel surround sound systems. In the early 1950s, the first commercial multichannel sound format was developed for cinema applications. "Quad" (Quadraphonic) was the first home-multichannel format, promoted in the early 1970s. But, due to some incompatibility issues in the encoding/decoding techniques, the Quad was not successful. In the mid-1970s, Dolby overcame the incompatibility issues associated with the optical sound tracks and introduced a new format, called the Dolby stereo, a special encoding technique that later became very popular. With the advent of *compact discs* (CDs) in the early 1980s, high-performance stereo systems became quite common. With the emergence of *digital versatile discs* (DVDs) in 1995–1996, content creators began to distribute multichannel music in digital format. Dolby laboratories, in 1992, introduced another coding algorithm (Dolby AC-3, Section 10.7), called the Dolby Digital that offers a high-quality multichannel (5.1-channel) surround sound experience. The Dolby Digital was later chosen as the primary audio coding technique for DVDs and for digital

audio broadcasting (DAB). The following year, Digital Theater Systems Inc. (DTS) announced a new format based on the Coherent Acoustics encoding principle (DTS-CA). The same year, Sony proposed the Sony Dynamic Digital Sound (SDDS) system that employs the Adaptive Transform Acoustic Coding (ATRAC) algorithm. Lucent Technologies' Multichannel Perceptual Audio Coder (MPAC) also has a five-channel surround sound configuration. Moreover, the development of two new audio recording technologies, namely, the Meridian Lossless Packing (MLP) and the Direct Stream Digital (DSD), for use in the DVD-Audio [DVD01] and SACD [SACD02] formats, respectively, offer audiophiles listening experiences that promise to be more realistic.

10.3.2 The Mono, the Stereo, and the Surround Sound Formats

Figure 10.4 shows the three most common sound formats, i.e., mono, stereo, and surround. Mono is a simple method of recording sound onto a single channel that is typically played back on one speaker. In stereo encoding, a two-channel recording is employed. Stereo provides a sound field in front, while the multichannel surround sound provides multi-dimensional sound experience. The surround sound systems typically employ a 5.1-channel configuration, i.e., sound tracks are recorded using five main channels: left (L), center (C), right (R), left surround (LS), and right surround (RS). In addition to these five channels, a sixth channel called the low-frequency-effects (LFE) channel is used for the subwoofer. Since the LFE channel covers only a fraction (less than 150 Hz) of the total frequency range, it is referred as the .1-channel.

10.3.3 The ITU-R BS.775 5.1-Channel Configuration

In an effort to evaluate and standardize the so-called 5.1- or 3/2-channel configuration, several technical documents appeared [Bosi93] [ITUR94c] [EBU99] [Holm99] [SMPTE99] [AES00] [Bosi00] [SMPTE02]. Various international standardization bodies became involved in multichannel algorithm adoption/evaluation process. These include: the Audio Engineering Society

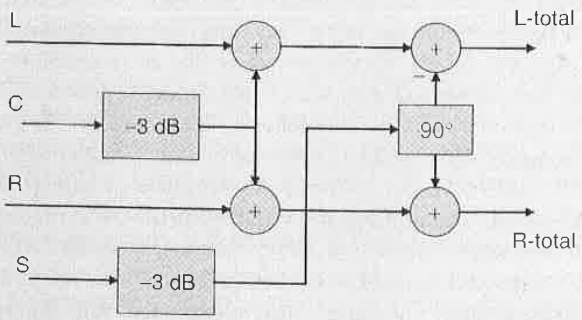


Figure 10.3. Multichannel surround sound matrixing.

Table 10.2. Milestones in multichannel surround sound.

Year	Description
1941	*Fantasia* (Walt-Disney Productions) was the first motion picture to be released in the multichannel format
1955	Introduction of the first 35/70-mm magnetic stripe capable of providing 4/6 channels
1972	Video cassette consumer format – mono (1 channel)
1976	Dolby's stereo in optical format
1978	Videocassette – stereo (2 channels)
1979	Dolby's first stereo surround, called the *split-surround sound format*, offered 3 screen channels, 2 surround channels, and a subwoofer (3/2/0.1)
1982	Dolby surround format implemented on a compact disc (2 channel)
1992	Dolby digital optical (5.1 channel)
1993	Digital Theater Systems (DTS)
1993–94	Sony Dynamic Digital Sound (SDDS) based on ATRAC
1994	The ISO/IEC 13818-3 MPEG-2 Backward compatible audio standard
1995	Dolby digital chosen for DVD (5.1 channel)
1997	DVD video released in market (5.1 channel)
1998	Dolby digital selected for digital audio broadcasting (DAB) in U.S.
1999–	Super Audio CD and DVD-Audio storage formats
2000–	Direct Stream Digital (DSD) and Meridian Lossless Packing (MLP) Technologies

(AES), the European Broadcasting Union (EBU), the Society of Motion Picture and Television Engineers group (SMPTE), the ISO/IEC MPEG, and the ITU-Radio communication sector (ITU-R).

Figure 10.5 shows a 5.1-channel configuration described in the ITU-R BS.775-1 standard [ITUR94c]. Ideally, five full-bandwidth (150 Hz–20 kHz) loudspeakers, i.e., L, R, C, LS, and RS are placed on the circumference of a circle in the following manner: the left (L) and right (R) front loudspeakers are placed at the extremities of an arc subtending, $2\theta = 60^{\circ}$, at the reference listening position (see Figure 10.5), and the center (C) loudspeaker must be placed at 0° from the listener's axis. This enables the compatibility with the listening arrangement for a conventional two-channel system. The two surround speakers, i.e., LS and RS are usually placed at $\phi = 110^{\circ}$ to 120° from the listener's axis. In order to achieve synchronization, the front and surround speakers must be equidistant, λ, (usually 2–4 m) from the reference listening point, with their acoustic centers in the horizontal plane as shown in the figure. The sixth channel, i.e., the LFE channel delivers bass-only omnidirectional information (20–150 Hz). This is because low frequencies imply longer-wavelengths where the ears are not sensitive to localization. The subwoofer placement receives less attention in the ITU-R standard; however, we note that the subwoofers are typically placed in a front corner (see Figure 10.4). In [Ohma97], Ingvar discusses the various problems associated with the subwoofer placement. Moreover, [SMPTE02] provides of information on the

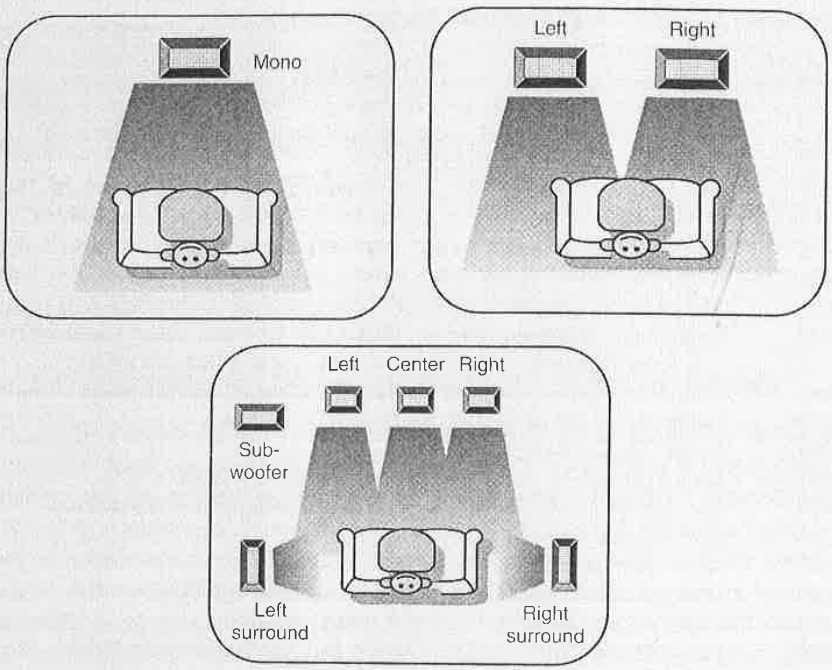


Figure 10.4. Mono, stereo, and surround sound systems.

loudspeaker placement for audio monitoring. The [SMPTE99] specifies the audio channel assignment and their relative levels for audio program recordings (3–6 audio channels) onto storage media for television sound.

10.4 MPEG AUDIO STANDARDS

MPEG is the acronym for *Moving Pictures Experts Group* that forms a workgroup (WG-11) of ISO/IEC JTC-1 subcommittee (SC-29). The main functions of MPEG are: a) to publish technical results and reports related to audio/video compression techniques; b) to define means to multiplex (combine) video, audio, and information bitstreams into a single bitstream, and c) to provide descriptions and syntax for low bit rate audio/video coding tools for Internet and bandwidth-restricted communications applications. MPEG standards do not characterize or provide any rigid encoder specifications, but rather standardizes the type of information that an encoder has to produce as well as the way in which the decoder has to decompress this information. The MPEG workgroup has its own official webpage that can be accessed at [MPEG]. The MPEG video aspect of the standard is beyond the scope of this book, however, we include some tutorial references and relevant standards [LeGal92] [Scha95] [ISO-V96] [Hask97] [Mitc97] [Siko97a] [Siko97b].

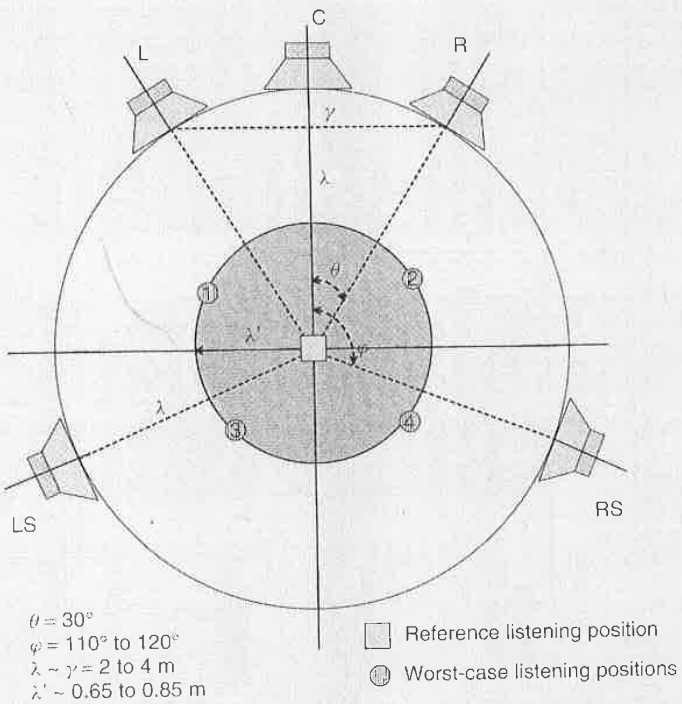


Figure 10.5. A typical 3/2-channel configuration described in the ITU-R BS.775-1 standard [ITUR94c]. L: left, C: center, R: right, LS: left surround, and RS: right surround loud speakers. Note that the above figure is not according to a scale.

MPEG Audio – Background. MPEG has come a long way since the first ISO/IEC MPEG standard was published in 1992. With the emergence of the Internet, MPEG is now also addressing content-based multimedia descriptions and database search. There are five different MPEG audio standards published, i.e., MPEG-1, MPEG-2 BC, MPEG-2 NBC/AAC, MPEG-4, and MPEG-7. MPEG-21 is being formed.

Before proceeding with the details of the MPEG audio standards, however, it is necessary to discuss terminology and notation. The *phases* correspond to the MPEG audio standards type and to a lesser extent to their relative release time, e.g., MPEG-1, MPEG-2, MPEG-4, etc. The *layers* represent a family of coding algorithms within the MPEG standards. Only MPEG-1 and -2 are provided with layers, i.e., MPEG-1 layer-I, -II, and -III; MPEG-2 layer-I, -II, and -III. The *versions* denote the various stages in the audio coding standardization *phase*. MPEG-4 was standardized in two stages (*version*-1 and -2) with new functionality being added to the older version. The newer versions are backward compatible to the older versions. Table 10.3 itemizes the various MPEG audio standards and their specifications. A brief overview of the MPEG standards follows.

Table 10.3. An overview of the MPEG audio standards.

Standard	Standardization details	Bit rates (kb/s)	Sampling rates (kHz)	Channels	Related references	Related information
MPEG-1	ISO/IEC 11172-3 1992	32–448 (Layer I) 32–384 (Layer II) 32–320 (Layer III)	32, 44.1,48	Mono (1), stereo (2)	[ISOI92] [Bran94a] [Shli94] [Pan95] [Noll93] [Noll95] [Noll97] [John99]	A generic compression standard that targets primarily multimedia storage and retrieval.
MPEG-2 BC/LSF	ISO/IEC 13818-3 1994	32–256 (Layer I) 8–160 (Layers II, III)	16, 22.05, 24	Multichannel Surround sound (5.1)	[ISOI94a] [Stol93a] [Gril94] [Noll93] [Noll95] [John99]	First digital television standard that enables lower frequencies and multichannel audio coding.
MPEG-2 NBC/AAC	ISO/IEC 13818-7 1997	8–160	8–96	Multichannel	[ISOI97b] [Bosi96] [Bosi97] [John99] [ISOI99] [Koen99] [Quac98b] [Gril99]	Advanced audio coding scheme that incorporates new coding methodologies (e.g., prediction, noise shaping, etc.).

Table 10.3. (*continued*)

Standard	Standardization details	Bit rates (kb/s)	Sampling rates (kHz)	Channels	Related references	Related information
MPEG-4 (Version 1)	ISO/IEC 14496-3 Oct, 1998	0.2–384	8–96	Multichannel	[Gril97] [Park97] [Edle99] [Purn00a] [Sche98a] [Vaan00] [Sche01]	The first content-based multimedia standard, allowing universality/interactivity and a combination of natural and synthetic material, coded in the form of objects.
MPEG-4 (Version 2)	ISO/IEC 14496-3/AMD-1, Dec, 1999	0.2–384 (finer levels of increment possible)	8–96	Multichannel	[ISOI00] [Kim01] [Herr00a] [Purn99b] [Alla99] [Hilp00] [Sper00] [Sper01]	
MPEG-7	ISO/IEC 15938-4 Sept, 2001	–	–	–	[ISOI01b] [Nack99a] [Nack99b] [Quac01] [Lind99] [Lind00] [Lind01] [Speng01] [ISOI02a]	A normative metadata standard that provides a Multimedia Content Description Interface.
MPEG-21	ISO/IEC-21000	–	–	–	[ISOI03a] [Borm03] [Burn03]	A multimedia framework that provides interoperability in content-access and distribution.

MPEG-1. After four years of extensive collaborative research by audio coding experts worldwide, the first ISO/MPEG audio coding standard, MPEG-1 [ISOI92], for VHS-stereo-CD-quality was adopted in 1992. The MPEG-1 supports video bit rates up to about 1.5 Mb/s providing a Video Home System (VHS) quality, and stereo audio at 192 kb/s. Applications of MPEG-1 range from storing video and audio on CD-ROMs to Internet streaming through the popular MPEG-1 layer III (MP3) format.

MPEG-2 Backward Compatible (BC). In order to extend the capabilities offered by MPEG-1 to support the so-called 3/2 (or 5.1) channel format and to facilitate higher bit rates for video, MPEG-2 [ISOI94a] was published in 1994. The MPEG-2 standard supports digital video transmission in the range of 2–15 Mb/s over cable, satellite, and other broadcast channels; audio coding is defined at the bit rates of 64–192 kb/s/channel. Multichannel MPEG-2 is backward compatible with MPEG-1, hence, the acronym MPEG-2 BC. The MPEG-2 BC standard is used in the high definition Television (HDTV) [ISOI94a] and produces the video quality required in digital television applications.

MPEG-2 Nonbackward Compatible/Advanced Audio Coding (AAC). The backward compatibility constraints imposed on the MPEG-2 BC/LSF algorithm made it impractical to code five channels at rates below 640 kb/s. As a result, MPEG began standardization activities for a nonbackward compatible advanced coding system targeting "indistinguishable" quality at a rate of 384 kb/s for five full bandwidth channels. In less than three years, this effort led to the adoption of the MPEG-2 nonbackward compatible/advanced audio coding (NBC/AAC) algorithm [ISOI97b], a system that exceeded design goals and produced the desired quality at only 320 kb/s for five full bandwidth channels.

MPEG-4. MPEG-4 was established in December 1998 after many proposed algorithms were tested for compliance with the program objectives established by the MPEG committee. MPEG-4 video supports bit rates up to about 1 Gb/s. The MPEG-4 audio [ISOI99] [ISOI00] was released in several steps, resulting in versions 1 and 2. MPEG-4 comprises an integrated family of algorithms with wide-ranging provisions for scalable, object-based speech and audio coding at bit rates from as low as 200 b/s up to 60 kb/s per channel. The distinguishing features of MPEG-4 relative to its predecessors are extensive scalability, object-based representations, user interactivity/object manipulation, and a comprehensive set of coding tools available to accommodate trade-offs between bit rate, complexity, and quality. Very low rates are achieved through the use of structured representations for synthetic speech and music, such as text-to-speech and MIDI. The standard also provides integrated coding tools that make use of different signal models depending upon the desired bit rate, bandwidth, complexity, and quality.

MPEG-7. The MPEG-7 audio committee activities started in 1996. In less than four years, a committee draft was finalized and the first audio standard addressing "multimedia content description interface" was published in September 2001. MPEG-7 [ISOI01b] targets the content-based multimedia applications. In particular, the MPEG-7 audio supports a broad range of applications – multimedia digital libraries, broadcast media selection, multimedia editing and searching,

multimedia indexing/searching. Moreover, it provides ways for efficient audio file retrieval and supports both the text-based and context-based queries.

MPEG-21. The MPEG-21 ISO/IEC-21000 standard [MPEG] [ISOI02a] [ISOI03a] defines interoperable and highly automated tools that enable content distribution across different terminals and networks in a programmed manner. This structure enables end-users to have capabilities for universal multimedia access.

10.4.1 MPEG-1 Audio (ISO/IEC 11172-3)

The MPEG-1 audio standard (ISO/IEC 11172-3) [ISOI92] comprises a flexible hybrid coding technique that incorporates several methods including subband decomposition, filter-bank analysis, transform coding, entropy coding, dynamic bit allocation, nonuniform quantization, adaptive segmentation, and psychoacoustic analysis. MPEG-1 audio codec operates on 16-bit PCM input data at sample rates of 32, 44.1, and 48 kHz. Moreover, MPEG-1 offers separate modes for mono, stereo, dual independent mono, and joint stereo. Available bit rates are 32–192 kb/s for mono and 64–384 kb/s for stereo. Several tutorials on the MPEG-1 standards [Noll93] [Bran94a] [Shli94] [Bran95] [Herr95] [Noll95] [Pan95] [Noll97] [John99] have appeared. Chapter 5, Section 5.7, presents step-by-step procedure involved in the ISO/IEC 11172-3 (MPEG-1, layer I) psychoacoustic model 1 [ISOI92] simulation. We summarize these steps in the context of MPEG-1 audio standard.

The MPEG-1 architecture contains three layers of increasing complexity, delay, and output quality. Each higher layer incorporates functional blocks from the lower layers. Figure 10.6 shows the MPEG-1 layer I/II encoder block diagram. The input signal is first decomposed into 32 critically subsampled subbands using a polyphase realization of a pseudo-QMF (PQMF) bank (see also Chapter 6). The channels are equally spaced such that a 48-kHz input signal is split into 750-Hz subbands, with the subbands decimated 32:1. A 511th-order prototype filter was chosen such that the inherent overall PQMF distortion remains below the threshold of audibility. Moreover, the prototype filter was designed

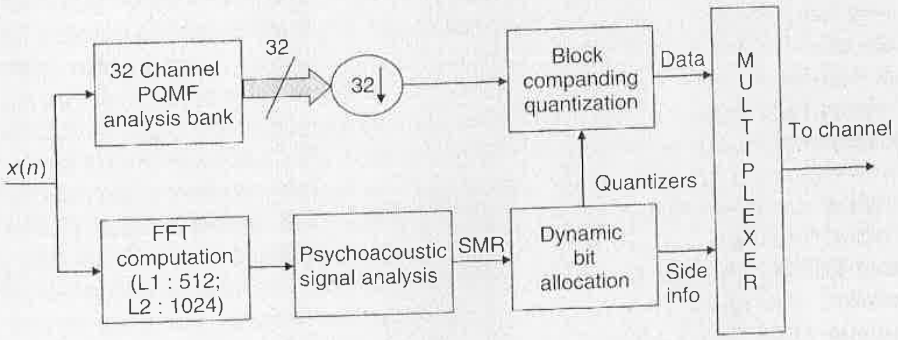


Figure 10.6. ISO/MPEG-1 layer I/II encoder.

for a high sidelobe attenuation (96 dB) to insure that intraband aliasing remains negligible. For the purposes of psychoacoustic analysis and determination of just noticeable distortion (JND) thresholds, a 512 (layer I) or 1024 (layer II) point FFT is computed in parallel with the subband decomposition for each decimated block of 12 input samples (8 ms at 48 kHz). Next, the subbands are block companded (normalized by a scale factor) such that the maximum sample amplitude in each block is unity, then an iterative bit allocation procedure applies the JND thresholds to select an optimal quantizer from a predetermined set for each subband. Quantizers are selected such that both the masking and bit rate requirements are simultaneously satisfied. In each subband, scale factors are quantized using 6 bits and quantizer selections are encoded using 4 bits.

10.4.1.1 Layers I and II For layer I encoding, decimated subband sequences are quantized and transmitted to the receiver in conjunction with side information, including quantized scale factors and quantizer selections. Layer II improves three portions of layer I in order to realize enhanced output quality and reduce bit rates at the expense of greater complexity and increased delay. First, the layer II perceptual model relies upon a higher-resolution FFT (1024 points) than does layer I (512 points). Second, the maximum subband quantizer resolution is increased from 15 to 16 bits. Despite this increase, a lower overall bit rate is achieved by decreasing the number of available quantizers with increasing subband index. Finally, scale factor side information is reduced while exploiting temporal masking by considering properties of three adjacent 12-sample blocks and optionally transmitting one, two, or three scale factors plus a 2-bit side parameter to indicate the scale factor mode. Average mean opinion scores (MOS) of 4.7 and 4.8 were reported [Noll93] for monaural layer I and layer II codecs operating at 192 and 128 kb/s, respectively. Averages were computed over a range of test material.

10.4.1.2 Layer III The layer III MPEG architecture (Figure 10.7) achieves performance improvements by adding several important mechanisms on top of the layer I/II foundation. The MPEG layer-III algorithm operates on consecutive frames of data. Each frame consists of 1152 audio samples; a frame is further split into two subframes of 576 samples each. A subframe is called a granule. At the decoder, every granule can be decoded independently. A hybrid filter bank is introduced to increase frequency resolution and thereby better approximate critical band behavior. The hybrid filter bank includes adaptive segmentation to improve pre-echo control. Sophisticated bit allocation and quantization strategies that rely upon nonuniform quantization, analysis-by-synthesis, and entropy coding are introduced to allow reduced bit rates and improved quality. The hybrid filter bank is constructed by following each subband filter with an adaptive MDCT. This practice allows for higher-frequency resolution and pre-echo control. Use of an 18-point MDCT, for example, improves frequency resolution to 41.67 Hz per spectral line. The adaptive MDCT switches between 6 and 18 points to allow improved pre-echo control. Shorter blocks (4 ms) provide

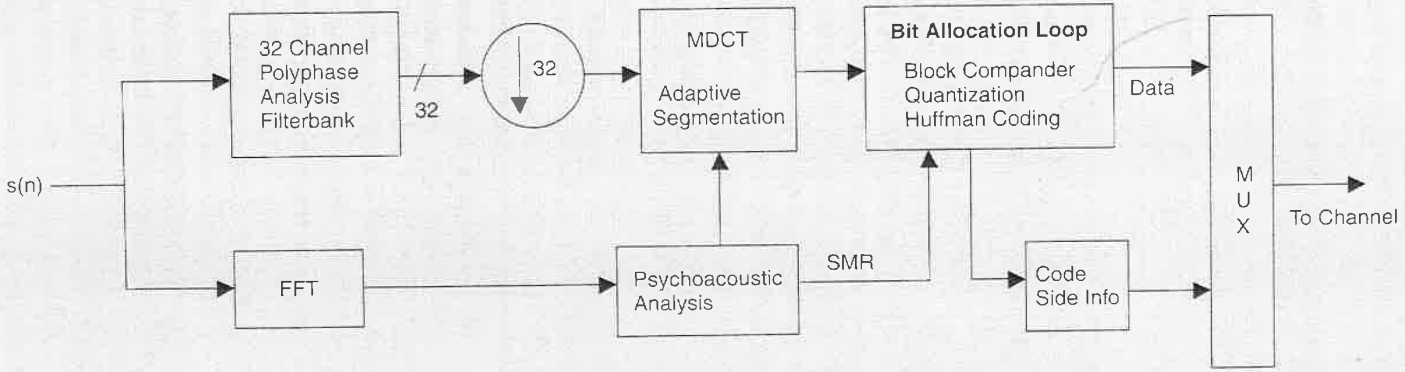


Figure 10.7. ISO/MPEG-1 layer III encoder.

for temporal pre-masking of pre-echoes during transients; longer blocks during steady-state periods improve coding gain by reducing side information and hence bit rates.

Bit allocation and quantization of the spectral lines are realized in a nested loop procedure that uses both nonuniform quantization and Huffman coding. The inner loop adjusts the nonuniform quantizer step sizes for each block until the number of bits required to encode the transform components falls within the bit budget. The outer loop evaluates the quality of the coded signal (analysis-by-synthesis) in terms of quantization noise relative to the JND thresholds. Average MOS of 3.1 and 3.7 were reported [Noll93] for monaural layer II and layer III codecs operating at 64 kb/s.

10.4.1.3 Applications MPEG-1 has been successful in numerous applications. For example, MPEG-1 layer III has become the standard for transmission and storage of compressed audio for both World Wide Web (WWW) and handheld media applications (e.g., IPod™). In these applications, the "MP3" label denotes MPEG-1, layer III. Note that MPEG-1 audio coding has steadily gained acceptance and ultimately has been deployed in several other large scale systems, including the European digital radio (DBA) or Eureka [Jurg96], the direct broadcast satellite or "DBS" [Prit90], and the digital compact cassette or "DCC" [Lokh92]. In particular, the Philips Digital Compact Cassette (DCC) is an example of a consumer product that essentially implements the 384 kb/s stereo mode of MPEG-1 layer I. A discussion of the precision adaptive subband coding (PASC) algorithm and other elements of the DCC system are given in [Lokh92] and [Hoog94].

The collaborative European Advanced Communications Technologies and Services (ACTS) program adopted MPEG audio and video as the core compression technology for the Advanced Television at Low Bit rates And Networked Transmission over Integrated Communication systems (ATLANTIC) project. ATLANTIC is a system intended to provide functionality for television program production and distribution [Stor97] [Gilc98]. This system posed new challenges for MPEG deployment such as seamless bitstream (source) switching [Laub98] and robust transcoding (tandem coding). Bitstream (source) switching becomes nontrivial when different bit rates and/or MPEG layers are associated with different program sources. Robust transcoding is also essential in the video production environment. Editing tasks inevitably require retrieval of compressed bit streams from archival storage, processing of program material in uncompressed form, and then replacement of the recoded compressed bit stream to the archival system. Unfortunately, transcoding is neither guaranteed nor likely to preserve perceptual noise masking [Rits96]. The ATLANTIC designers proposed a buried data "MOLE" signal to mitigate and in some cases eliminate transcoding distortion for cascaded MPEG-1 layer II codecs [Flet98], ideally allowing downstream tandem stages to preserve the original bit stream. The idea behind the MOLE is to apply the same set of quantizers to the same set of data in the downstream codecs as in the original codec. The output bit stream will then be identical to the original bit stream, provided that

numerical precision in the analysis filter banks does not bias the data [tenK96b]. It is possible in a cascade of MPEG-1 layer II codecs to regenerate the same set of decimated subband sequences in the downstream codec filter banks as in the original codec filter bank if the full-band PCM signal is properly time aligned at the input to each cascaded stage. Essentially, delays at the filter-bank input must correspond to integer delays at the subband level [tenK96b], and the analysis frames must contain the same block of data in each analysis filter bank. The MOLE signal, therefore, provides downstream codecs with timing synchronization, bit allocation, and scale-factor information for the MPEG bit stream on each frame. The MOLE is buried in the PCM samples between tandem stages and remains inaudible by occupying the LSB of each 20-bit PCM word. Although optimal time-alignment between codecs is possible even without the MOLE [tenK96b], there is unfortunately no easy way to force selection of the same set of quantizers and thus preserve the bit stream.

The widespread use and maturity of MPEG-1 relative to the more recent standards provided several concrete examples for the above discussion of MPEG-1 audio applications. Various real-time implementation schemes of MPEG-1 layers-I, II, and III codecs were proposed [Gbur96] [Hans96] [Main96] [Wang01]. We will next consider the MPEG-2 BC/LSF, MPEG-2 AAC, the MPEG-4, and the MPEG-7 algorithms. The discussion will focus primarily upon architectural novelties and differences with respect to MPEG-1.

10.4.2 MPEG-2 BC/LSF (ISO/IEC-13818-3)

MPEG-2 BC/LSF Audio [Stol93a] [Gril94] [ISOI94a] [Stol96] extends the capabilities offered by MPEG-1 to support the so-called *3/2-channel format* with left (L), right (R), center (C), and left and right surround (LS and RS) channels. The MPEG-2 BC/LSF audio standard is *backward compatible* with MPEG-1, which means that the 3/2 channel information transmitted by an MPEG-2 encoder can be appropriately decoded for 2-channel presentation by an MPEG-1 receiver. Another important feature that was implemented in MPEG-2 BC/LSF is the *multilingual compatibility*. The acronym BC corresponds to the backward compatibility of MPEG-2 towards MPEG-1, and the extension of sampling frequencies to lower ranges (16, 22.05, and 24 kHz) is denoted by LSF. Several tutorials on MPEG-2 [Noll93] [Noll95] [John99] have appeared. Meares and Theile studied the potential application of matrixed surround sound [Mear97] in MPEG audio algorithms.

10.4.2.1 The Backward Compatibility Feature Depending on the bit-demand constraints, interchannel dependencies, and the complexity allowed at the decoder, different methods can be employed to realize compatibility between the 3/2- and 2-channel formats. These methods include mid/side (MS), intensity coding, simulcast, and matrixing. The MS and intensity coding techniques are particularly handy when bit demand imposed by multiple independent channels exceeds the bit budget. The MS scheme is carefully controlled [Davi98]

to maintain compatibility among the mono, stereo, and the surround sound formats. Intensity coding, also known as channel coupling, is a multichannel irrelevancy reduction coding technique that exploits properties of spatial hearing. The idea behind intensity coding is to transmit only one envelope with some side information instead of two or more from independent channels. The side information consists of a set of coefficients that is used to recover individual spectra from the intensity channel. The simulcast encoding involves transmission of both stereo and multichannel bitstreams. Two separate bitstreams, i.e., one for 2-channel stereo and another one for the multichannel audio are transmitted, resulting in reduced coding efficiency.

MPEG-2 BC/LSF employs matrixing techniques [tenK92] [tenK94] [Mear97] to down-mix the 3/2 channel format to the 2-channel format. Down-mixing capability is essential for the 5.1-channel system since many of the playback systems are stereophonic or even monaural. Figure 10.8 depicts the matrixing technique employed in the MPEG-2 BC/LSF and can be mathematically expressed as follows

$$L\_total = x(L + yC + zL_s) \tag{10.1}$$

$$R\_total = x(R + yC + zR_s), \tag{10.2}$$

where x, y, and z are constants specified by the IS-13818-3 MPEG-2 standard [ISOI94a]. In Eqs. (10.1) and (10.2), L, C, R, L_s, and R_s represent the 3/2-channel configuration and the parameters *L_total* and *R_total* correspond to the 2-channel format.

Three different choices are provided in the MPEG-2 audio standard [ISOI94a] for choosing the values of x, y, and z to perform the 3/2-channel to 2-channel down-mixing. These include:

$$\text{Choice1}: \quad x = \frac{1}{1+\sqrt{2}},\ y = \frac{1}{\sqrt{2}},\ \text{and}\ z = \frac{1}{\sqrt{2}} \tag{10.3}$$

$$\text{Choice2}: \quad x = \frac{2}{3+\sqrt{2}};\ y = \frac{1}{\sqrt{2}};\ \text{and}\ z = \frac{1}{2} \tag{10.4}$$

$$\text{Choice3}: \quad x = 1;\ y = 1;\ \text{and}\ z = 1 \tag{10.5}$$

The selection of the down-mixing parameters is encoder dependent. The availability of the basic stereo format channels, i.e., *L_total* and *R_total* and the surround sound extension channels, i.e., C,L_s, and R_s at the decoder helps to decode both 3/2-channel and 2-channel bitstreams. This insures the backwards compatibility in the MPEG-2 BC/LSF audio coding standard.

10.4.2.2 MPEG-2 BC/LSF Encoder The steps involved in the reduction of the objective redundancies and the removal of the perceptual irrelevancies in MPEG-2 BC/LSF encoding are the same as in MPEG-1 audio standard. However, the differences arise from employing multichannel and multilingual bitstream

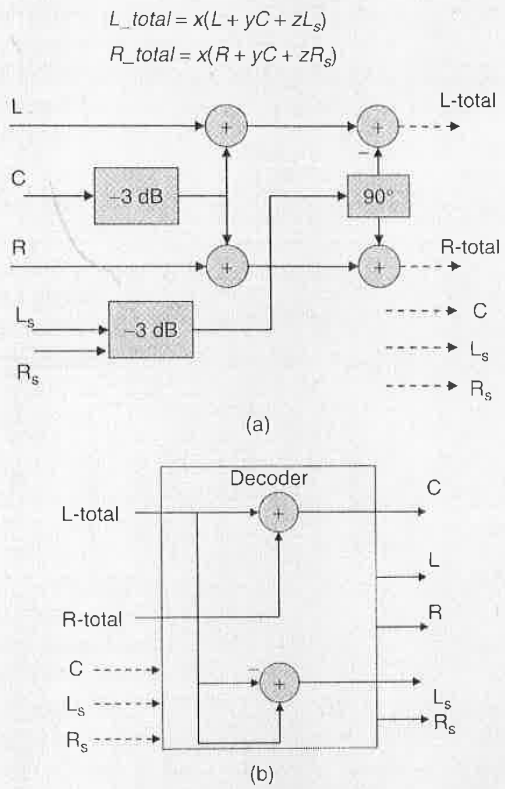


Figure 10.8. Multichannel surround sound matrixing: (a) encoder (b) decoder.

format in the MPEG-2 audio. A *matrixing* module is used for this purpose. Another important feature employed in the MPEG-2 BC/LSF is the "dynamic cross-talk," a multichannel irrelevancy reduction technique. This feature exploits properties of spatial hearing and encodes only one envelope instead of two or more together with some side information (i.e., scale factors). Note that this technique is in some sense similar to the intensity coding that we discussed earlier. In summary *matrixing* enables backwards compatibility between the MPEG-2 and MPEG-1 bitstreams, and *dynamic cross-talk* reduces the interchannel redundancies.

In Figure 10.9, first the segmented audio frames are decomposed into 32 critically subsampled subbands using a polyphase realization of a *pseudo QMF* (PQMF) bank. Next, a *matrixing module* is employed for down-mixing purposes. Matrixing results in two stereo-format channels, i.e., *L_total* and *R_total* and three extension channels, i.e., C, L_s, and R_s. In order to remove statistical redundancies associated with these channels a second-order linear *predictor* is employed [Fuch93] [ISOI94a]. The predictor coefficients are updated on each subband using a backward adaptive LMS algorithm [Widr85]. The resulting

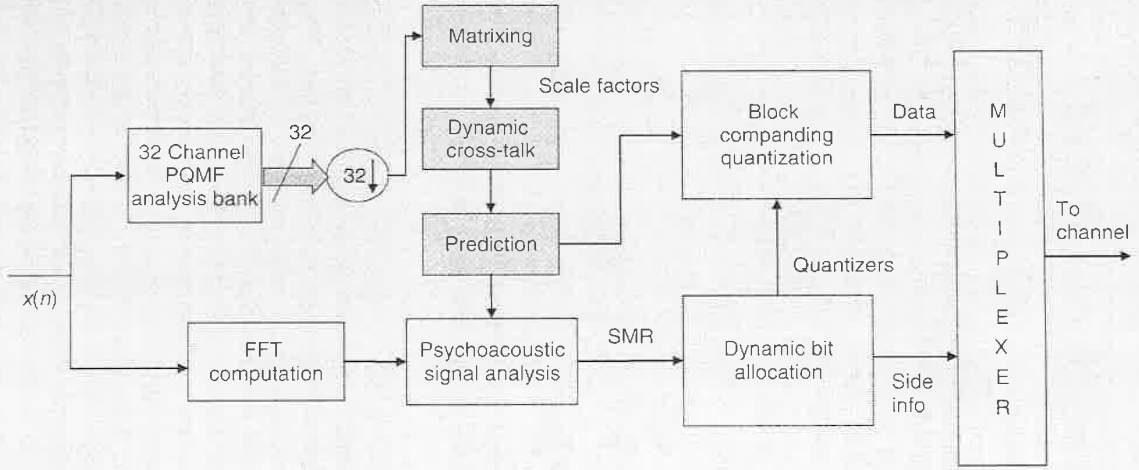


Figure 10.9. ISO/MPEG-2 BC/LSF audio (layer I/II) encoder algorithm.

prediction error is further processed to eliminate interchannel dependencies. JND thresholds are computed in parallel with the subband decomposition for each decimated block. A bit-allocation procedure similar to the one in the MPEG-1 audio standard is used to estimate the number of appropriate bits required for quantization.

10.4.2.3 MPEG-2 BC/LSF Decoder Synchronization followed by error detection and correction are performed first at the decoder. Then the coded audio bitstream is de-multiplexed into the individual subbands of each audio channel. Next, the subband signals are converted to subband PCM signals, based on the instructions in the header and the side information transmitted for every subband. De-matrixing is performed to compute L and R bitstreams as follows:

$$L = \frac{L\_total}{x} - (yC + zL_s) \tag{10.6}$$

$$R = \frac{R\_total}{x} - (yC + zR_s) \tag{10.7}$$

where x, y, and z are constants and are known at the decoder. The inverse-quantized, de-matrixed subband PCM signals are then inverse-filtered to reconstruct the full-band time-domain PCM signals for each channel.

The second MPEG-2 standard, i.e., the MPEG-2 NBC/AAC, sacrificed backward MPEG-1 compatibility to eliminate quantization noise unmasking artifacts [tenK94], which are potentially introduced by the forced backward compatibility.

10.4.3 MPEG-2 NBC/AAC (ISO/IEC-13818-7)

The 11172-3 MPEG-1 and IS13818-3 MPEG-2 BC/LSF are standardized algorithms for high-quality coding of monaural and stereophonic program material. By the early 1990s, however, the demand for high-quality coding of multichannel audio at reduced bit rates had increased significantly. The backwards compatibility constraints imposed on the MPEG-2 BC/LSF algorithm made it impractical to code 5-channel program material at rates below 640 kb/s. As a result, MPEG began standardization activities for a nonbackward compatible advanced coding system targeting "indistinguishable" quality [ITUR91] [ISOI96a] at a rate of 384 kb/s for five full-bandwidth channels. In less than three years, this effort led to the adoption of the IS13818-7 MPEG-2 Non-backward Compatible/Advanced Audio Coding (NBC/AAC) algorithm [ISOI97b], a system that exceeded design goals and produced the desired quality at 320 kb/s for five full-bandwidth channels. While similar in many respects to its predecessors, the AAC algorithm [Bosi96] [Bosi97] [Bran97] [John99] achieves performance improvements by incorporating coding tools previously not found in the standards such as filter-bank window shape adaptation, spectral coefficient prediction, temporal noise shaping (TNS), and bandwidth- and bit-rate-scaleable operation. Bit rate and quality

improvements are also realized through the use of a sophisticated noiseless coding scheme integrated with a two-stage bit allocation procedure. Moreover, the AAC algorithm contains scalability and complexity management tools not previously included with the MPEG algorithms. As far as applications are concerned, the AAC algorithm is embedded in the atob™ and LiquidAudio™ players for streaming of high-fidelity stereophonic audio. It is also a candidate for standardization in the United States Digital Audio Radio (US DAR) project. The remainder of this section describes some of the features unique to MPEG-2 AAC.

The MPEG-2 AAC algorithm (Figure 10.10) is organized as a set of coding tools. Depending upon available CPU or channel resources and desired quality, one can select from among three complexity "profiles," namely main, low, and scalable sample rate profiles. Each profile recommends a specific combination of tools. Our focus here is on the complete set of tools available for main profile coding, which works as follows.

10.4.3.1 Filter Bank First, a high-resolution MDCT filter bank obtains a spectral representation of the input. Like previous MPEG coders, the AAC filter-bank resolution is signal adaptive. Quasi-stationary segments are analyzed with a 2048-point window, while transients are analyzed with a block of eight 256-point windows to maintain time synchronization for channels using different filter-bank resolutions during multichannel operations. The frequency resolution is therefore 23 Hz for a 48-kHz sample rate, and the time resolution is 2.6 ms. Unlike previous MPEG coders, however, AAC eliminates the hybrid filter bank and relies on the MDCT exclusively. The AAC filter bank is also unique in its ability to switch between two distinct MDCT analysis window shapes, i.e., a sine window (Eq. (10.8)) and a Kaiser-Bessel designed (KBD) window (Eq. (10.9)). Given specific input signal characteristics, the idea behind window shape adaptation is to optimize filter-bank frequency selectivity in order to localize the supra-masking threshold signal energy in the fewest spectral coefficients. This strategy seeks essentially to maximize the perceptual coding gain of the filter bank. While both windows satisfy the perfect reconstruction and aliasing cancellation constraints of the MDCT, they offer different spectral analysis properties. The sine window is given by

$$w(n) = \sin\left[\left(n + \frac{1}{2}\right)\frac{\pi}{2M}\right] \tag{10.8}$$

for $0 \leqslant n \leqslant M - 1$, where M is the number of subbands. This particular window is perhaps the most popular in audio coding. In fact, this window has become standard in MDCT audio applications, and its properties are typically referenced as performance benchmarks when new windows are proposed. The so-called KBD window was obtained in a procedure devised at Dolby Laboratories, by applying a transformation of the form

$$w_a(n) = w_s(n)\sqrt{\frac{\sum_{j=0}^{n} v(j)}{\sum_{j=0}^{M} v(j)}}, \quad 0 \leqslant n < M, \tag{10.9}$$

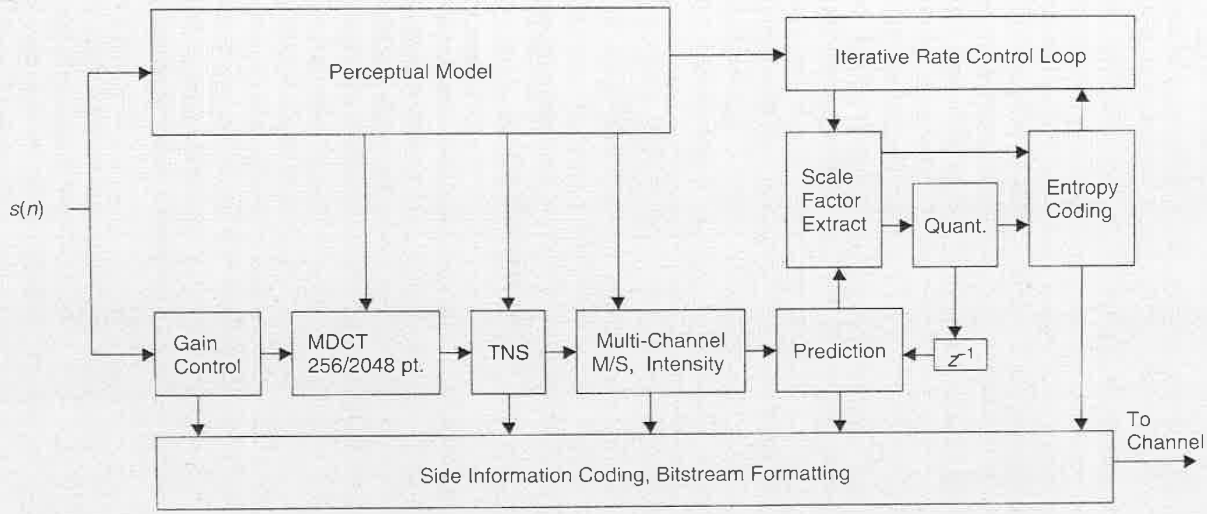


Figure 10.10. ISO/IEC IS13818-7 (MPEG-2 NBC/AAC) encoder ([Bosi96]).

where the sequence $v(n)$ represents the symmetric kernel. The resulting identical analysis and synthesis windows, $w_a(n)$ and $w_s(n)$, respectively, are of length $M + 1$ and symmetric, i.e., $w(2M - n - 1) = w(n)$. More detailed explanation on the MDCT windows is given in Chapter 6, Section 6.7.

A filter-bank simulation exemplifying the performance of the two windows, sine and KBD, for the MPEG-2 AAC algorithm follows. A sine window is selected when narrow pass-band selectivity is more beneficial than strong stop-band attenuation. For example, sounds characterized by a dense harmonic structure (less than 140 Hz spacing) such as harpsichord or pitch pipe benefit from a sine window. On the other hand, a Kaiser-Bessel designed (KBD) window is selected in cases for which stronger stop-band attenuation is required, or for situations in which strong components are separated by more than 220 Hz. The KBD window in AAC has its origins in the MDCT filter bank window designed at Dolby Labs for the AC-3 algorithm using explicit perceptual criteria. By sacrificing pass-band selectivity, the KBD window gains improved stop-band attenuation relative to the sine window. In fact, the stop-band magnitude response is below a conservative composite minimum masking threshold for a tonal masker at the center of the pass-band. A KBD versus sine window simulation example (Figure 10.11) for a signal containing 300 Hz plus 3 harmonics shows the KBD potential for reduced bit allocation. A masking threshold estimate generated by MPEG-1 psychoacoustic model 2 is superimposed (red line). It can be seen that, for the given input, the KBD window is advantageous in terms of supra-threshold component minimization. All of the MDCT components below the superimposed masking threshold will potentially require allocations of zero bits. This tradeoff can ultimately lead to a lower bit rate. Details of the minimum masking template design procedure are given in [Davi94] and [Fiel96].

10.4.3.2 Spectral Prediction The AAC algorithm realizes improved coding efficiency relative to its predecessors by applying prediction over time to the transform coefficients below 16 kHz, as was done previously in [Mahi89] [Fuch93] [Fuch95]. In this case, the spectral prediction tool is applied only during long analysis windows and then only if a bit-rate reduction is obtained when coding the prediction residuals instead of the original coefficients. Side information is minimal, since the second-order lattice predictors are updated on each frame using a backward adaptive LMS algorithm. The predictor banks, which can be selectively activated for individual quantization scale-factor bands, produced an improvement for a fixed bit rate of +1 point on the ITU 5-point impairment scale for the critical pitch pipe and harpsichord test material.

10.4.3.3 Bit Allocation The bit allocation and quantization strategies in AAC bear some similarities to previous MPEG coders in that they make use of a nested-loop iterative procedure, and in that psychoacoustic masking thresholds are obtained from an analysis model similar to MPEG-1, model recommendation number two. Both lossy and lossless coding blocks are integrated into the rate-control loop structure so that redundancy removal and irrelevancy reduction are

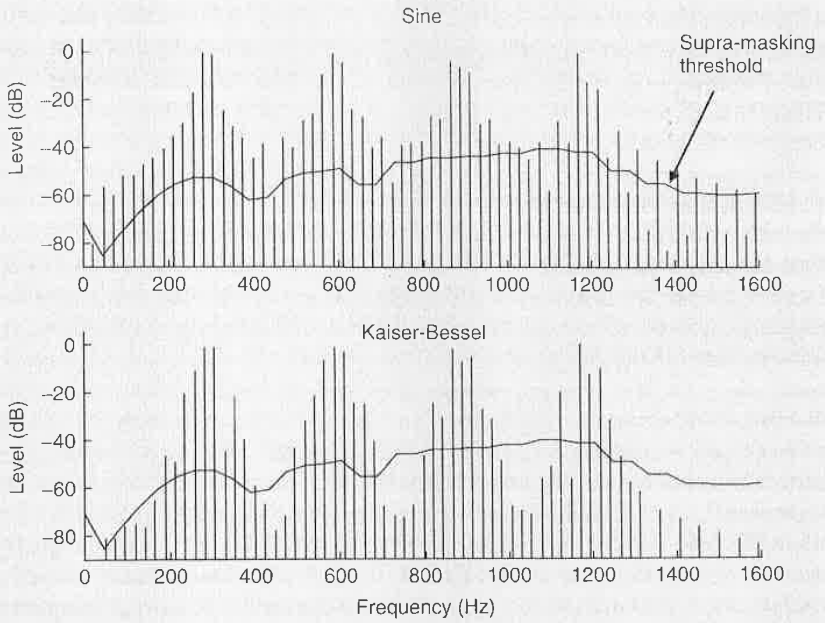


Figure 10.11. Comparison of the MPEG-2 AAC MDCT analysis filter-bank outputs for the sine window vs. the KBD window.

simultaneously affected in a single analysis-by-synthesis process. The scheme works as follows. As in the case of MPEG-1, layer III, the AAC coefficients are grouped into 49 scale-factor bands that mimic the auditory system's frequency resolution.

In the nested-loop allocation procedure, the inner loop adjusts scale-factor quantizer step sizes in increments of 1.5 dB (approximates intensity difference limen (DL)) and obtains Huffman codewords for both quantized scale factors and quantized coefficients until the desired bit rate is achieved. Then, in the outer loop, the quantization noise introduced by the inner loop is compared to the masking threshold in order to assess noise audibility. Undercoded scale factor bands are amplified to force increased coding precision, and then the inner loop is called again for compliance with the desired bit rate. A best result is stored after each iteration since the two-loop process is not guaranteed to converge. As with other algorithms such as the MPEG-1 layer III and the Lucent Technologies PAC [John96c], a bit reservoir is maintained to compensate for time-varying perceptual bit-rate requirements.

10.4.3.4 Noiseless Coding The noiseless coding block [Quac97] embedded in the rate-control loop has several innovative features as well. Twelve Huffman code books are available for 2- and 4-tuple blocks of quantized coefficients. Sectioning and merging techniques are applied to maximize redundancy reduction.

Individual code books are applied to time-varying "sections" of scale-factor bands, and the sections are defined on each frame through a greedy merge algorithm that minimizes the bit rate. Grouping across time and intraframe frequency interleaving of coefficients prior to code-book application are also applied to maximize zero coefficient runs and further reduce bit rates.

10.4.3.5 Other Enhancements Relative to MPEG-1 and MPEG-2 BC/LSF, other enhancements have also been embedded in AAC. For example, the AAC algorithm has an embedded TNS module [Herr96] for pre-echo control (Section 6.9), a special profile for sample-rate scalability (SSR), and time-varying as well as frequency subband selective application of MS and/or intensity stereo coding for 5-channel inputs [John96b].

10.4.3.6 Performance Incorporation of the nonbackward compatible coding enhancements proved to be a judicious strategy for the AAC algorithm. In independent listening tests conducted worldwide [ISOI96d], the AAC algorithm met the strict ITU-R BS.1116 criteria for indistinguishable quality [ITUR94b] at a rate of 320 kb/s for five full-bandwidth channels [Kirb97]. This level of quality was achieved with a manageable decoder complexity. Two-channel, real-time AAC decoders were reported to run on 133-MHz Pentium platforms using 40% and 25% of available CPU resources for the main and low-complexity profiles, respectively [Quac98a]. MPEG-2 AAC maintained its presence as the core "time-frequency" coder reference model for the MPEG-4 standard.

10.4.3.7 Reference Model Validation (RM) Before proceeding with a discussion of MPEG-4, we first consider a significant system-level aspect of MPEG-2 AAC that also propagated into MPEG-4. Both algorithms are structured in terms of so-called *reference models* (RMs). In the RM approach, generic coder blocks or tools (e.g., perceptual model, filter bank, rate-control loop, etc.) adhere to a set of defined interfaces. The RM therefore facilitates the testing of incremental single block improvements without disturbing the existing macroscopic RM structure. For instance, one could devise a new psychoacoustic analysis model that satisfies the AAC RM interface and then simply replace the existing RM perceptual model in the reference software with the proposed model. It is then a straightforward matter to construct performance comparisons between the RM method and the proposed method in terms of quality, complexity, bit rate, delay, or robustness. The RM definitions are intended to expedite the process of evolutionary coder improvements.

In fact, several practical AAC improvements have already been analyzed within the RM framework. For example, a backward predictor was proposed [Yin97] as a replacement for the existing backward adaptive LMS predictors. This method that relies upon a block LPC estimation procedure rather than a running LMS estimation, was reported to achieve comparable quality with a 38% (instruction) complexity reduction [Yin97]. This contribution was significant in light of the fact that the spectral prediction tool in the AAC main profile decoder constitutes 40%

of the computational complexity [Yin97]. Decoder complexity is further reduced since the block predictors only require updates when the prediction module has been enabled rather than requiring sample-by-sample updating regardless of activation status. Forward adaptive predictors have also been investigated [Ojan99]. In another example of RM efficacy, improvements to the AAC noiseless coding module were reported in [Taka97]. A modification to the greedy merge sectioning algorithm was proposed in which high-magnitude spectral peaks that tended to degrade Huffman coding efficiency were coded separately. The improvement yielded consistent bit-rate reductions up to 11%. In informal listening tests it was found that the bit savings resulted in higher quality at the same bit rate. In yet another example of RM innovation aimed at improving quality for a given bit rate, product code VQ techniques [Gers92] were applied to increase AAC scale-factor coding efficiency [Sree98a]. In the proposed scheme, scale-factors are decorrelated using a DCT and then grouped into subvectors for quantization by a product code VQ. The method is intended primarily for low-rate coding, since the side information bit burden rises from roughly 6% at 64 kb/s to in some cases 25% at 16 kb/s. As expected, subjective tests reflected an insignificant quality improvement at 64 kb/s. On the other hand, the reduction in bits allocated to side information at low rates (e.g., 16 kb/s), allowed more bits for spectral coefficient coding, and therefore produced mean improvements of +0.52 and +0.36 on subjective differential improvement tests at bit rates of 16 and 40 kb/s, respectively [Sree98b]. Additionally, noise-to-mask ratios (NMRs) were reduced by as much as −2.43 for the "harpsichord" critical test item at 16 kb/s. Several architectures for MPEG-2 AAC real-time implementations were proposed. Some of these include [Chen99] [Hilp98] [Geye99] [Saka00] [Hong01] [Rett01] [Taka01] [Duen02] [Tsai02].

10.4.3.8 Enhanced AAC in MPEG-4 The next section is concerned with the multimodal MPEG-4 audio standard, for which the MPEG-2 AAC RM core was selected as the "time-frequency" audio coding RM with some improvements. For example, perceptual noise substitution (PNS) was included [Herr98a] as part of the MPEG-4 AAC RM. Moreover, the long-term prediction (LTP) [Ojan99] and transform-domain weighted interleave VQ (TwinVQ) [Iwak96] modules became part of the MPEG-4 audio. LTP after the MPEG-2 AAC prediction block provides a higher coding precision for tonal signals, while the TwinVQ provided scalability and ultra-low bit-rate audio coding.

10.4.4 MPEG-4 Audio (ISO/IEC 14496-3)

The MPEG-4 ISO/IEC-14496 Part 3 audio was adopted in December 1998 after many proposed algorithms were tested [Cont96] [Edle96a] [ISOI96b] [ISOI96c] for compliance with the program objectives [ISOI94b] established by the MPEG committee. MPEG-4 audio (Figure 10.12) encompasses a great deal more functionality than just perceptual coding [Koen96] [Koen98] [Koen99]. It comprises an integrated family of algorithms with wide-ranging provisions for scalable,

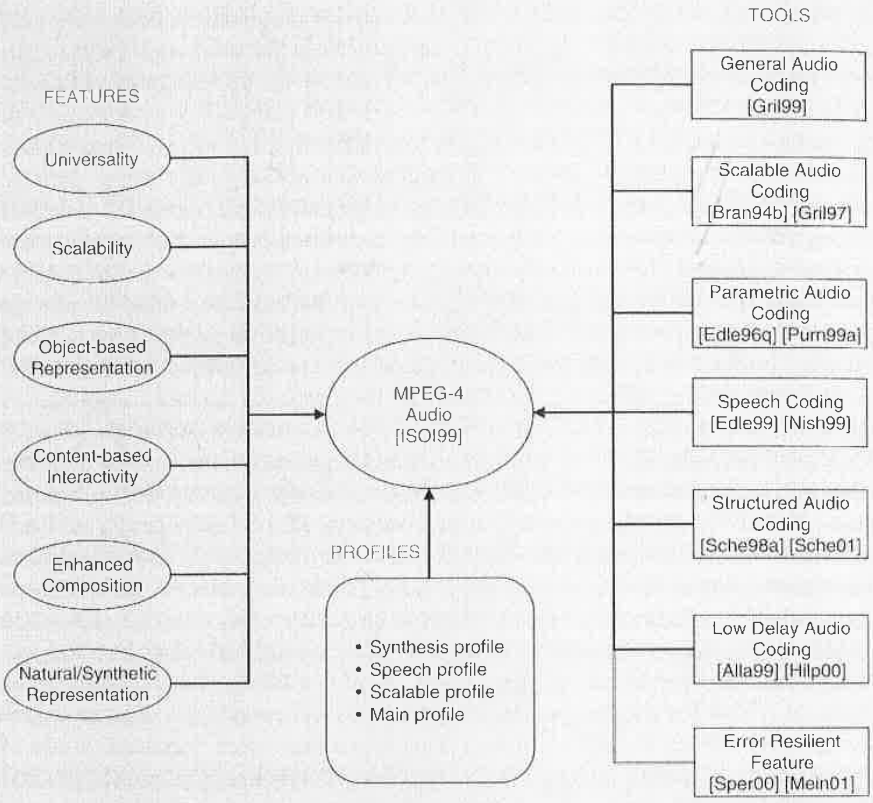


Figure 10.12. An overview of the MPEG-4 audio coder.

object-based speech and audio coding at bit rates from as low as 200 b/s up to 64 kb/s per channel. The distinguishing features of MPEG-4 relative to its predecessors are extensive scalability, object-based representations, user interactivity/object manipulation, and a comprehensive set of coding tools available to accommodate almost any desired tradeoff between bit rate, complexity, and quality. Efficient and flexible coding of different content (objects) such as natural audio/speech and synthetic audio/speech became indispensable for some of the innovative multimedia applications. To facilitate this, MPEG-4 audio provides coding and composition of natural and synthetic audio/speech content at various bit rates. Very low rates are achieved through the use of structured representations for synthetic speech and music, such as text-to-speech and MIDI. For higher bit rates and "natural audio" speech and music, the standard provides integrated coding tools that make use of different signal models, the choice of which is made depending upon desired bit rate, bandwidth, complexity, and quality. Coding tools are also specified in terms of MPEG-4 "profiles" that essentially recommend tool sets for a given level of functionality and complexity. Beyond its provisions specific to coding of speech and audio, MPEG-4 also specifies

numerous sophisticated system-level functions for media-independent transport, efficient buffer management, syntactic bitstream descriptions, and time-stamping for synchronization of audiovisual information units.

10.4.4.1 MPEG-4 Audio Versions The MPEG-4 audio standard was released in several steps due to timing constraints. This resulted in two different versions of MPEG-4. Version 1 [ISOI99] was standardized in February 1999, followed by version 2 [ISOI00] (also referred as Amendment 1 to version 1) in February 2000. New amendments for bandwidth extension, parametric audio extension, MP3 on MP4, audio lossless coding, and scalable to lossless coding have also been considered in the MPEG-4 audio standard.

MPEG-4 Audio Version 1. The MPEG-4 audio version 1 comprises the majority of the MPEG-4 audio tools. These are general audio coding, scalable coding, speech coding techniques, structured audio coding, and text-to-speech synthetic coding. These techniques can be grouped into two main categories, i.e., *natural* [Quac98b] and *synthetic* audio coding [Vaan00]. The MPEG-4 *natural audio coding* part describes traditional type speech coding and high-quality audio coding algorithms at bit rates ranging from 2 kb/s to 64 kb/s and above. Three types of coders enable hierarchical (scalable) coding in MPEG-4 Audio version-1 at different bit rates. Firstly, at lower bit rates ranging from 2 kb/s to 6 kb/s, parametric speech coding is employed. Secondly, a code excited linear predictive (CELP) coding is used for medium bit rates between 6 kb/s and 24 kb/s. Finally, for the higher bit rates typically ranging from 24 kb/s, transform-based (time-frequency) general audio coding techniques are applied. The MPEG-4 *synthetic audio coding* part describes the text-to-speech (TTS) and structured audio synthesis tools. Typically, the structured tools are used to provide effects like echo, reverberation, and chorus effects; the TTS synthetic tools generate synthetic speech from text parameters.

MPEG-4 Audio Version 2. While remaining backwards compatible with MPEG-4 version 1, version 2 adds new profiles that incorporate a number of significant system-level enhancements. These include error robustness, low-delay audio coding, small-step scalability, and enhanced composition [Purn99b]. At the system level, version 2 includes a media independent bit stream format that supports streaming, editing, local playback, and interchange of contents. Furthermore in version 2, an MPEG-J programmatic system specifies an application programming interface (API) for interoperation of MPEG players with JAVA. Version 2 offers improved audio realism in sound rendering. New tools allow parameterization of the acoustical properties of an audio scene, enabling features such as immersive audiovisual rendering, room acoustical modeling, and enhanced 3-D sound presentation. New error resilience techniques in version 2 allow both equal and unequal error protection for the audio bit streams. Low-delay audio coding is employed at low bit rates where the coding delay is significantly high. Moreover, to facilitate the bit rate scalability in small steps, version 2 provides a highly desirable tool called small-step scalability or fine-grain scalability. Text-to-speech (TTS) interfaces from version 1 are enhanced in version 2 with a mark-up TTS intended for

applications such as speech-enhanced web browsing, verbal email, and story-teller on demand. Markup TTS has the ability to process HTML, SABLE, and facial animation parameter (FAP) bookmarks.

10.4.4.2 MPEG-4 Audio Profiles Although many coding and processing tools are available in MPEG-4 audio, cost and complexity constraints often dictate that it is not practical to implement all of them in a particular system. Version 1 therefore defines four complexity-ranked audio profiles intended to help system designers in the task of appropriate tool subset selection. In order of bit rate, they are as follows. The *low rate synthesis audio profile* provides only wavetable-based synthesis and a text-to-speech (TTS) interface. For natural audio processing capabilities, the *speech audio profile* provides a very-low-rate speech coder and a CELP speech coder. The *scalable audio profile* offers a superset of the first two profiles. With bit rates ranging from 6 to 24 kb/s and bandwidths from 3.5 to 9 kHz, this profile is suitable for scalable coding of speech, music, and synthetic music in applications such as Internet streaming or narrow-band audio digital broadcasting (NADIB). Finally, the *main audio profile* is a superset of all other profiles, and it contains tools for both natural and synthetic audio.

10.4.4.3 MPEG-4 Audio Tools Unlike MPEG-1 and MPEG-2, the MPEG-4 audio describes not only a set of compression schemes but also a complete functionality for a broad range of applications from low-bit-rate speech coding to high-quality audio coding or music synthesis. This feature is called the *universality*. MPEG-4 enables scalable audio coding, i.e., variable rate encoding is provided to adapt dynamically to the varying transmission channel capacity. This property is called *scalability*. One of the main features of the MPEG-4 audio is its ability to represent the audiovisual content as a set of objects. This enables the *content-based interactivity*.

Natural Audio Coding Tools. MPEG-4 audio [Koen99] integrates a set of tools (Figure 10.13) for coding of natural sounds [Quac98b] at bit rates ranging from as low as 200 b/s up to 64 kb/s per channel. For speech and audio, three distinct algorithms are integrated into the framework. These include parametric coding, CELP coding, and transform coding. The parametric coding is employed for bit rates of 2–4 kb/s and 8 kHz sampling rate as well as 4–16 kb/s and 8 or 16 kHz sampling rates (Section 9.4). For higher quality, narrow-band (8 kHz sampling rate) and wideband (16 kHz) speech is handled by a CELP speech codec operating between 6 and 24 kb/s. For generic audio at bit rates above 16 kb/s, a time/frequency perceptual coder is employed, and, in particular, the MPEG-2 AAC algorithm with extensions for fine-grain bit-rate scalability [Park97] is specified in MPEG-4 version 1 RM as the time-frequency coder. The multimodal framework of MPEG-4 audio allows the user to tailor the coder characteristics to the program material.

Synthetic Audio Coding Tools. While the earlier MPEG standards treated only natural audio program material, the MPEG-4 audio achieves very-low-rate coding by supplementing its natural audio coding techniques with tools for synthetic audio processing [Sche98a] [Sche01] and interfaces for structured, high-level

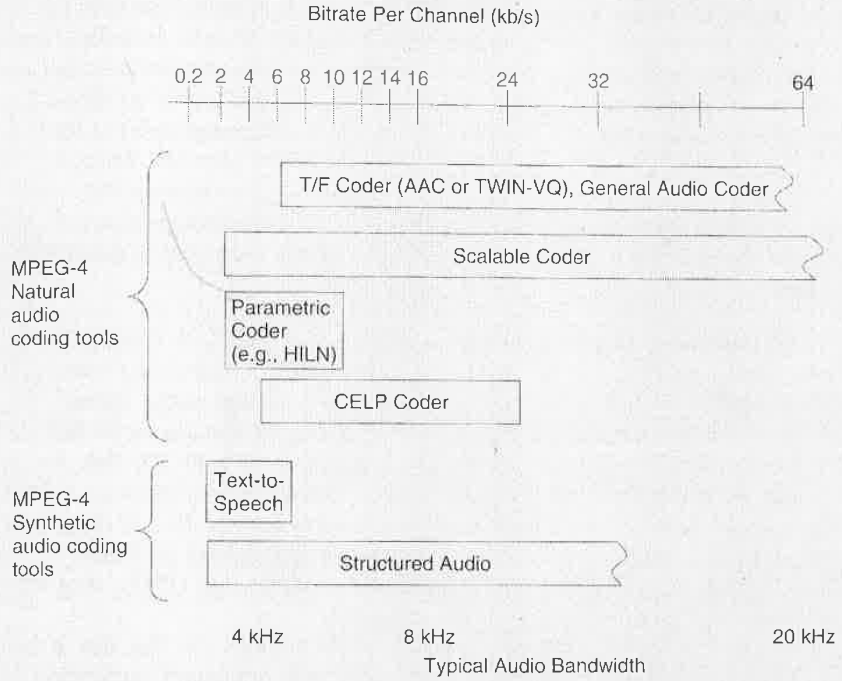


Figure 10.13. ISO/IEC MPEG-4 integrated tools for audio coding ([Koen99]).

audio representations. Chief among these are the text-to-speech interface (TTSI) and methods for score-driven synthesis. The TTSI provides the capability for 200–1200 b/s transmission of synthetic speech that can be represented in terms of either text only or text plus prosodic parameters such as a pitch contour or a set of phoneme durations. Also, one can specify the age, gender, and speech rate of the speaker. Additionally, there are facilities for lip synchronization control, international language, and dialect support, as well as controls for pause, resume, and jump forward/backward. The TTSI specifies only an interface rather than a normative speech synthesis methodology in order to maximize implementation flexibility.

Beyond speech, general music synthesis capabilities in MPEG-4 are provided by a set of structured audio tools [Sche98a] [Sche98d] [Sche98e]. Synthetic sounds are represented using the structured audio orchestra language (SAOL). SAOL [Sche98d] treats music as a collection of instruments. Instruments are then treated as small networks of signal-processing primitives, all of which can be downloaded to a decoder. Some of the available synthesis methods include wavetable, FM, additive, physical modeling, granular synthesis, or nonparametric hybrids of any of these methods [Sche98c]. An excellent tutorial on these and other structured audio methods and applications appeared in [Verc98]. The SAOL instruments are controlled at the decoder by "scores" or scripts in the

structured audio score language (SASL). A score is a time-sequenced set of commands that invokes various instruments at specific times to contribute their outputs to an overall performance. SASL provides significant flexibility in that not only can instruments be controlled, but the existing sounds can be modified. For those situations in which fine control is not required, structured audio in MPEG-4 also provides backward compatibility with the MIDI protocol. Moreover, a standardized "wavetable bank format" is available for low-functionality terminals [Koen99]. In the next seven subsections, i.e., 10.4.4.4 through 10.4.4.10, we describe in detail the features and tools (Figure 10.12) integrated in the MPEG-4 audio.

10.4.4.4 MPEG-4 General Audio Coding The MPEG-4 General Audio Coder (GAC) [Gril99] has the most vital and versatile functionality associated with the MPEG-4 tool-set that covers the arbitrary natural audio signals. The MPEG-4 GAC is often called as the "all-round" coding system among the MPEG-4 audio schemes and operates at bit rates ranging from 6 to 300 kb/s and at sampling rates between 7.35 kHz and 96 kHz. The MPEG-4 GA coder is built around the MPEG-2 AAC (Figure 10.10 discussed in Section 10.4.3) along with some extended features and coder configurations highlighted in Figure 10.14. These features are given by the perceptual noise substitution (PNS), long-term prediction (LTP), Twin VQ coding, and scalability.

Perceptual Noise Substitution (PNS). The PNS exploits the fact that a random noise process can be used to model efficiently transform-coefficients in noise-like frequency subbands, provided the noise vector has an appropriate temporal fine structure [Schu96]. Bit-rate reduction is realized since only a compact, parametric representation is required for each PNS subband (i.e., noise energy) rather than full quantization and coding of subband transform coefficients. The PNS technique was integrated into the existing AAC bitstream definition in a backward-compatible manner. Moreover, PNS actually led to reduced decoder complexity since pseudo-random sequences are less expensive to compute than Huffman decoding operations. Therefore, in order to improve the coding efficiency, the following principle of PNS is employed.

The PNS acronym is composed from the following: *perceptual* coding + *substitute* parametric form of *noise*-like signals, i.e., PNS allows frequency-selective parametric encoding of noise-like components. These noise-like components are detected based on a scale-factor band and are grouped into separate categories. The spectral coefficients corresponding to these categories are not quantized and are excluded from the coding process. Furthermore, only a noise substitution flag along with the total power of these spectral coefficients are transmitted for each band. At the decoder, the spectral coefficients are replaced by the pseudo-random vectors with the desired target noise power. At a bit rate of 32 kb/s, a mean improvement due to PNS of +0.61 on the comparison mean opinion score (CMOS) test (for critical test items such as speech, castanets, and complex sound mixtures) was reported in [Herr98a]. The multichannel PNS modes include some provisions for binaural masking level difference (BMLD) compensation.

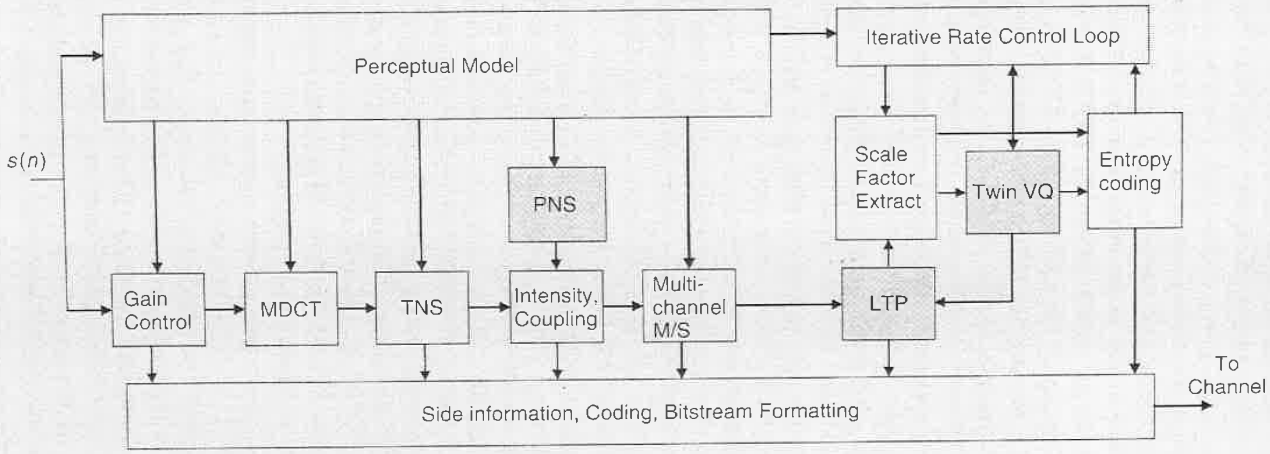


Figure 10.14. MPEG-4 GA coder.

Long-Term Prediction (LTP). Unlike the noise-like signals, the tonal signals require higher coding precision. In order to achieve a required coding precision (20 dB for tone-like and 6 dB for noise-like signals), the long-term prediction (LTP) technique [Ojan99] is employed. In particular, since the tonal signal components are predictable, the speech coding pitch prediction techniques [Span94] can be used to improve the coding precision. The only significant difference between the prediction techniques performed in a common speech coder and in the MPEG-4 GA coder is that in the latter case, the LTP is performed in the frequency domain, while in speech codecs the LTP is carried out in the time domain. A brief description of the LTP scheme in MPEG-4 GA coder follows. First, the input audio is transformed to frequency domain using an analysis filter bank and later a TNS analysis filter is employed for shaping the noise artifacts. Next, the processed spectral coefficients are quantized and encoded. For prediction purposes, these quantized coefficients are transformed back to the time domain by a synthesis filter bank and the associated TNS operation. The optimum pitch lag and the gain parameters are determined based on the residual and the input signal. In the next step, both the input signal and the residual are mapped to a spectral representation via the analysis filter bank and the forward TNS filter bank. Depending on which alternative is more favorable, coding of either the difference signal or the original signal is selected on a scale-factor basis. This is achieved by means of a so-called frequency-selective switch (FSS), which is also used in the context of the MPEG-4 GA scalable systems. The complexity associated with the LTP in MPEG-4 GA scheme is considerably (50%) reduced compared to the MPEG-2 AAC prediction scheme [Gril99].

Twin VQ. Twin VQ [Iwak96] [Hwan01] [Iwak01] is an acronym of the *Transform-domain Weighted Interleave Vector Quantization*. The Twin VQ performs vector quantization of the transformed spectral coefficients based on a perceptually weighted model. The quantization distortion is controlled through a perceptual model [Iwak96]. The Twin VQ provides high coding efficiencies even for music and tonal signals at extremely low bit rates (6–8 kb/s), which CELP coders fail to achieve. The Twin VQ performs quantization of the spectral coefficients in two steps as shown in Figure 10.15. First, the spectral coefficients are flattened and normalized across the frequency axis. Second, the flattened spectral coefficients are quantized based on a perceptually weighted vector quantizer.

From Figure 10.15, the first step includes a linear predictive coding, periodicity computation, a Bark scale spectral estimation scheme, and a power computation block. The LPC provides the overall spectral shape. The periodic component includes information on the harmonic structure. The Bark-scale envelope coding provides the required additional flattening of the spectral coefficients. The normalization restricts these spectral coefficients to a specific target range. In the second step, the flattened and normalized spectral coefficients are interleaved into subvectors. Based on some spectral properties and a weighted distortion measure, perceptual weights are computed for each subvector. These weights are applied to the vector quantizer (VQ). A conjugate-structure VQ that uses a pair of code books is employed. More detailed information on the conjugate structure VQ can

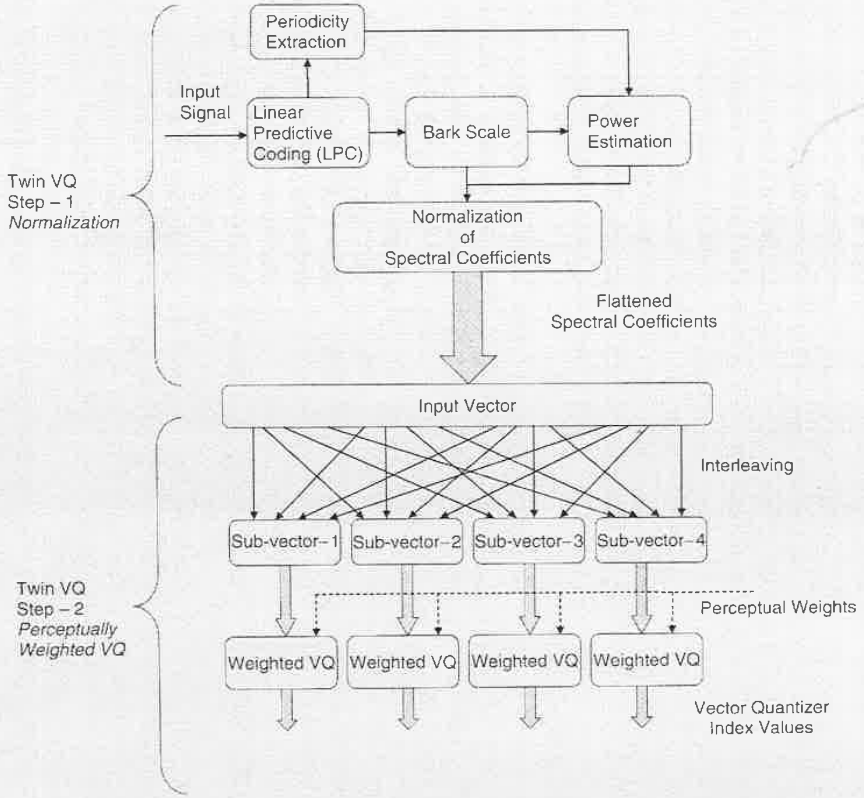


Figure 10.15. Twin VQ scheme in MPEG-4 GA coder.

be obtained from [Kata93] [Kata96]. The MPEG-4 Audio Twin VQ scheme provides audio coding at ultra-low bit rates (6–8 kb/s) and supports the perceptual control of the quantization distortion. Comparative tests of MPEG AAC with and without Twin VQ tool were performed and are given in [ISO198]. Furthermore, the Twin VQ tool has provisions for scalable audio coding, which will be discussed next.

10.4.4.5 MPEG-4 Scalable Audio Coding MPEG-4 scalable audio coding implies a variable rate encoding/decoding of bitstreams at bit rates that can be adapted dynamically to the varying transmission channel capacity [Gril97] [Park97] [Herr98b] [Creu02]. Scalable coding schemes [Bran94b] generate partial bitstreams that can be decoded separately. Therefore, encoding/decoding of a subset of the total bitstream will result in a valid signal at a lower bit rate. The various types of scalability [Gril97] are given by, signal-to-noise ratio (SNR) scalability, noise-to-mask ratio (NMR) scalability, audio bandwidth scalability, and bit-rate scalability. The bit-rate scalability is considered to be one of the core functionalities of the MPEG-4 audio standard. Therefore, in our discussion on the MPEG-4 scalable audio coding, we will consider only the bit-rate scalability and the various scalable coder configurations described in the standard.

The MPEG-4 bit-rate scalability scheme (Figure 10.16) allows an encoder to transmit bitstreams at a high bit rate, while decoding successfully a low-rate bitstream contained within the high-rate code. For instance, if an encoder transmits bitstreams at 64 kb/s, the decoder can decode at bit rates of 16, 32, or 64 kb/s according to channel capacity, receiver complexity, and quality requirements. Typically, scalable audio coders constitute several layers, i.e., a core layer and a series of enhancement layers. For example, Figure 10.16 depicts one core layer and two enhancement layers. The core layer encodes the core (main) audio stream, while the enhancement layers provide further resolution and scalability. In particular, in the first stage, the core layer encodes the input audio, $s(n)$, based on a conventional lossy compression scheme. Next, an error signal (residual), $E_1(n)$ is calculated by subtracting the reconstructed signal, $\hat{s}(n)$ (that is obtained by decoding the compressed bitstream locally) from the input signal, $s(n)$. In the second stage (first enhancement layer), the error signal $E_1(n)$ is encoded to obtain the compressed residual, $e_1(n)$. The above sequence of steps is repeated for all the enhancement layers.

To further demonstrate this principle we consider an example (Figure 10.16) where the core layer uses 32 kb/s, and the two enhancement layers employ bit rates of 16 kb/s and 8 kb/s, and the final sink layer supports 8 kb/s coding. Therefore, if no side information is encoded, then the coding rate associated with the codec is 64 kb/s. At the decoder, one can decode this multiplexed audio bitstream at various rates, i.e., 64, 32, or 40 kb/s, etc., depending up on the bit-rate requirements, receiver complexity, and channel capacity. In particular, the core bitstream guarantees reconstruction of the original input audio with minimum artifacts. On top of the core layer, additional enhancement layers are added to increase the quality of the decoded signal.

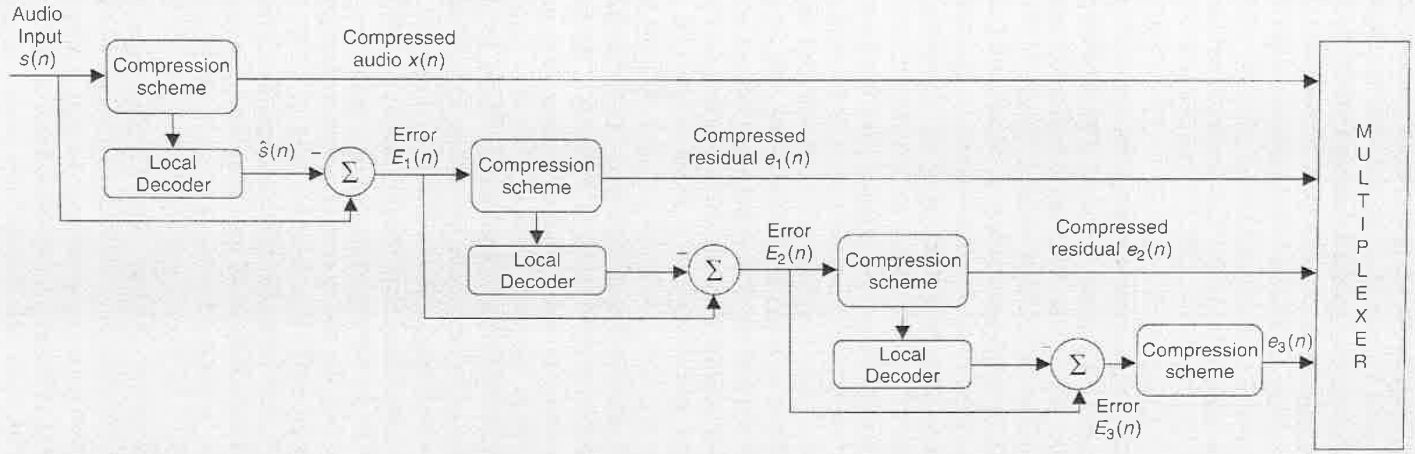


Figure 10.16. MPEG-4 scalable audio coding.

Scalable audio coding finds potential applications in the fields of digital audio broadcasting, mobile multimedia communication, and streaming audio. It supports real-time streaming with a low buffer delay. One of the significant extensions of the MPEG-4 scalable audio coding is the *fine-grain scalability* [Kim01], where a *bit-sliced arithmetic coding* (BSAC) [Kim02a] is used. In each frame, bit planes are coded in the order of significance, beginning with the most significant bits (MSBs) and progressing to the LSBs. This results in a fully embedded coder containing all lower-rate codecs. The BSAC and fine-grain scalability concepts are explained below in detail.

Fine-Grain Scalability. It is important that bit-rate scalability is achieved without significant coding efficiency penalty compared to fixed-bit-rate systems, and with low computational complexity. This can be achieved using the fine-grain scalability technique [Purn99b] [Kim01]. In this approach, *bit-sliced arithmetic coding* is employed along with the combination of advanced audio coding tools (Section 10.4.2). In particular, the noiseless coding of spectral coefficients and the scale-factor selection scheme is replaced by the BSAC technique that provides scalability in steps of 1 kb/s/channel. The BSAC scheme works as follows. First, the quantized spectral values are grouped into frequency bands, each of these groups contain the quantized spectral values in the binary form. Then the bits of each group are processed in slices and in the order of significance, beginning with the MSBs. These bit-slices are then encoded using an arithmetic coding technique (Chapter 3). Usually, the BSAC technique is used in conjunction with the MPEG-4 GA tool, where the Huffman coding is replaced by this special type of arithmetic coding.

10.4.4.6 MPEG-4 Parametric Audio Coding In research proposed as part of an MPEG-4 "core experiment" [Purn97], Purnhagen at the University of Hannover developed in conjunction with Deutsche Telekom Berkom an object-based algorithm. In this approach, harmonic sinusoid, individual sinusoid, and colored noise objects were combined in a hybrid source model to create a parametric signal representation. The enhanced algorithm, known as the "Harmonic and Individual Lines Plus Noise" (HILN) [Purn00a] [Purn00b] is architecturally very similar to the original ASAC [Edle96b] [Edle96c] [Purn98] [Purn99a], with some modifications. The parametric audio coding scheme is a part of MPEG-4 version 2, and is based on the HILN scheme (see also Section 9.4). This technique involves coding of audio signals at bit rates of 4 kb/s and above based on the possibilities of modifying the playback speed or pitch during decoding. The parametric audio coding tools have also been extended to high-quality audio [Oom03].

10.4.4.7 MPEG-4 Speech Coding The MPEG-4 natural speech coding tool [Edle99] [Nish99] provides a generic coding framework for a wide range of applications with speech signals at bit rates between 2 kb/s and 24 kb/s. The MPEG-4 speech coding is based on two algorithms, namely, harmonic vector excitation coding (HVXC) and code excited linear predictive coding (CELP). The

HVXC algorithm, essentially based on the parametric representation of speech, handles very low bit rates of 1.4–4 kb/s at a sampling rate of 8 kHz. On the other hand, the CELP algorithm employs multipulse excitation (MPE) and regular-pulse excitation (RPE) coding techniques (Chapter 4); and supports higher bit rates of 4–24 kb/s operating at sampling rates of 8 kHz and 16 kHz. The specifications of MPEG-4 Natural Speech Coding Tool Set are summarized in Table 10.4.

In all the aforementioned algorithms, i.e., HVXC, CELP-MPE, and CELP-RPE, the idea is that an LP analysis filter models the human vocal tract while an excitation signal models the vocal chord and the glottal activity. All the three configurations share the same LP analysis method, while they generally differ only in the excitation computation. In the LP analysis, first, the autocorrelation coefficients of the input speech are computed once every 10 ms and are converted to LP coefficients using the Levinson-Durbin algorithm. The LP coefficients are transformed to line spectrum pairs using Chebyshev polynomials [Kaba86]. These are later quantized using a two-stage, split-vector quantizer. The excitation signal is chosen in such a way that the error between the original and reconstructed signal is minimized according to a perceptually weighted distortion measure.

Multiple Bit Rates/Sampling Rates, Scalability. The speech coder family in MPEG-4 audio is different from the standard speech coding algorithms (e.g., ITU-T G.723.1, G.729, etc.). Some of the salient features and functionalities (Figure 10.17) of the MPEG-4 speech coder include multiple sampling rates and bit rates, bit-rate scalability [Gril97], and bandwidth scalability [Nomu98].

The *multiple bit rates/sampling rates* functionality provides flexible bit rate selection among multiple available bit rates (1.4–24 kb/s) based on the channel conditions and the bandwidth availability (8 kHz and 16 kHz). At lower bit rates, an algorithmic delay of the order of 30–40 ms is expected, while at higher bit

Table 10.4. MPEG-4 speech coding sampling rates and bandwidth specifications [Edle99].

Specification	HVXC	CELP-MPE	CELP-RPE
Sampling frequency (kHz)	8	8, 16	16
Bit rate (kb/s)	1.4–4	3.85–23.8 58 Bit rates	10.9–23.8 30 Bit rates
Frame size (ms)	10–40	10–40	10–20
Delay (ms)	33.5–56	~15–45	~20–25
Features	Multi-bit-rate coding, bit-rate scalability	Multi-bit-rate coding, bit-rate scalability, bandwidth scalability	Multi-bit-rate coding, bit-rate scalability

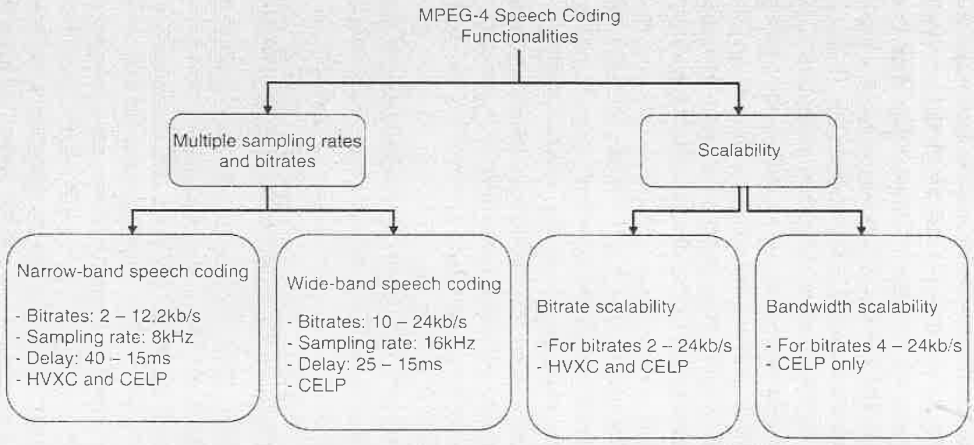


Figure 10.17. MPEG-4 speech coder functionalities.

rates, 15–25 ms delay is common. The *bit-rate scalability* feature allows a wide range of bit rates (2–24 kb/s) in step sizes of as low as 100 b/s. Both HVXC and CELP tools can be used to realize bit-rate scalability by employing a core layer and a series of enhancement layers at the encoder. When the HVXC encoding is used, one enhancement layer is preferred, while three bit-rate scalable enhancement layers may be used for the CELP codec [Gril97]. *Bandwidth scalability* improves the audio quality by adding an additional coding layer that extends the transmitted audio bandwidth. Only CELP-RPE and CELP-MPE schemes allow bandwidth scalability in the MPEG-4 audio. Furthermore, only one bandwidth scalable enhancement layer is possible [Nomu98] [Herr00a].

10.4.4.8 MPEG-4 Structured Audio Coding Structured audio (SA), introduced by Vercoe *et al.*, [Verc98] presents a new dimension to MPEG-4 audio, primarily due to its ability to represent and encode efficiently the synthetic audio and multimedia content. The MPEG-4 SA tool [Sche98a] [Sche98c] [Sche99a] [Sche99b] was developed based on a synthesizer-description language called the *Csound* [Verc95], developed by Vercoe at the MIT Media Labs. Moreover, the MPEG-4 SA tool inherits features from "Netsound" [Case96], a structured audio experiment carried out by Casey *et al.* based on the Csound synthesis language. Instead of specifying a synthesis method, the MPEG-4 SA describes a *special language* that defines synthesis methods. In particular, the MPEG-4 SA tool defines a set of syntax and semantic rules corresponding to the synthesis-description language called the Structured Audio Orchestra Language (SAOL) [Sche98d]. A control (score) language called the Structured Audio Score Language (SASL) was also defined to describe the details of the SAOL code compaction. Another component, namely, the Structured Audio Sample Bank Format (SASBF) is used for the transmission of data samples in blocks. These blocks contain sample data as well as details of the parameters used for selecting optimum wave-table synthesizers and facilitate algorithmic modifications. A theoretical basis for the SA coding was established in [Sche01] based on the Kolmogorov complexity theory. Also, in [Sche01], Scheirer proposed a new paradigm called the *generalized audio coding* in which SA encompasses all other audio coding techniques. Furthermore, treatment of structured audio in view of both lossless coding and perceptual coding is also given in [Sche01].

The SA bitstream available at the MPEG-4 SA decoder (Figure 10.18) consists of a header, sample data, and score data. The *SAOL decoder* block acts as an interpreter and reads the header structure. It also provides the information required to reconfigure the synthesis engine. The header carries descriptions of several instruments, synthesizers, control algorithms, and routing instructions. The *Event List and Data* block obtains the actual stream of data samples, and parameters controlling algorithmic modifications. In particular, the bitstream data consists of access units that primarily contain the list of events. Furthermore, each event refers to an instrument described (e.g., in the orchestra chunk) in the header [Sche01]. The *SASL decoder* block compiles the score data from the SA bitstream and provides control sequences and signals to the synthesis engine via a *run-time scheduler*. This control information determines the time at

which the events (or commands) are to be dispatched in order to create notes (or instances) of an instrument. Each note produces some sound output. Finally, all these sound outputs (corresponding to each note) are added, in order to create the overall orchestra output. In Figure 10.18, we represented the *run-time scheduler* and *reconfigurable synthesis engine* blocks separately, however, in practice they are usually combined into one block.

As mentioned earlier, the structured audio tool and the text-to-speech (TTS) fall in the synthetic audio coding group. Recall that the structured audio tools convert structured representation into synthetic sound, while the TTS tools translate text to synthetic speech. In both these methods, the particular synthesis method or implementation is not defined by the MPEG-4 audio standard; however, the input-output relation for SA and the TTS interface are standardized. The next question that arises is how the natural and synthetic audio content can be mixed. This is typically carried out based on a special format specified by the MPEG-4 namely, the Audio Binary Format for Scene Description (AudioBIFS) [Sche98e]. AudioBIFS enables sound mixing, grouping, morphing, and effects like echo (delay), reverberation (feedback delay), chorus, etc.

10.4.4.9 MPEG-4 Low-Delay Audio Coding Significantly large algorithmic delays (of the order of 100–200 ms) in the MPEG-4 GA coding tool (discussed in Section 10.4.4.4) hinder its applications in two-way, real-time communication. These algorithmic delays in the GA coder can be attributed primarily to the analysis/synthesis filter bank window, the look-ahead, the bit-reservoir, and the frame length. In order to overcome large algorithmic delays, a simplified version of the GA tool, i.e., the MPEG-4 low-delay (LD) audio coder has been proposed [Herr98c] [Herr99]. One of the main reasons for the wide proliferation of this tool is the low algorithm delay requirements in voice-over Internet protocol (VoIP) applications. In contrast to the ITU-T G.728 speech standard that is based

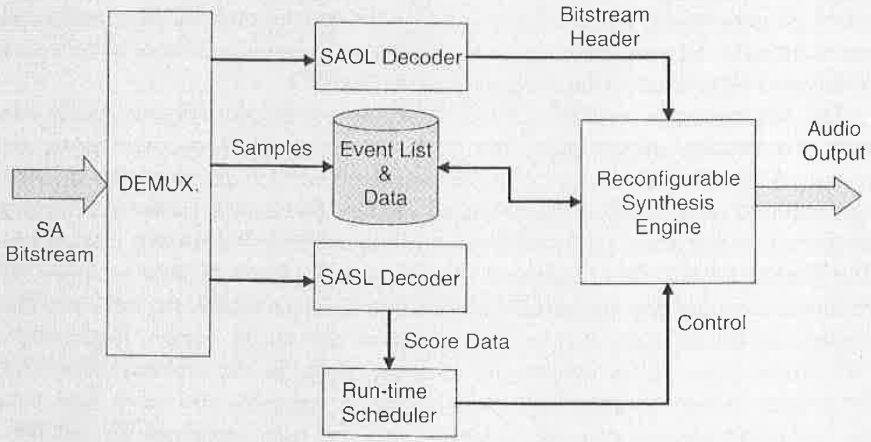


Figure 10.18. MPEG-4 SA decoder (after [Sche98a]).

on the LD-CELP [G728], the MPEG-4 LD audio coder [Alla99] is derived from the GA coder and MPEG-2 AAC. The ITU-T G.728 LD-CELP algorithm operates on speech frames of 2.5 ms (20 samples) at a sampling rate of 8 kHz and results in an algorithmic delay of 0.625 ms (5 samples). On the other hand, the MPEG-4 LD audio coding tool operates on 512 or 480 samples at a sampling rate of up to 48 kHz with an overall algorithmic delay of 20 ms. Recall that the GA tool that is based on the MPEG-2 AAC operates on frames of 1024 or 960 samples.

The delays due to the analysis/synthesis filter-bank window can be reduced by employing shorter windows. The look-ahead delays can be avoided by not employing the block switching. To reduce pre-echo distortions (Sections 6.9 and 6.10), TNS is employed in conjunction with window shape adaptation. In particular, for nontransient parts of the signal, a sine window is used, while a so-called low-overlap window is used in case of transient signals to achieve optimum TNS performance [Purn99b] [ISOI00]. Although most algorithms are fixed rate, the instantaneous bit rates required to satisfy masked thresholds on each frame are in fact time-varying. Thus, the idea behind a bit reservoir is to store surplus bits during periods of low demand, and then to allocate bits from the reservoir during localized periods of peak demand, resulting in a time-varying instantaneous bit rate but at the same time a fixed average bit rate. However, in MPEG-4 LD audio codec, the use of the bit reservoir is minimized in order to reach the desired target delay.

Based on the results published in [Alla99] [Herr99] [Purn99b] [ISOI00], the MPEG-4 LD audio codec performs relatively well compared to the MP3 coder at a bit rate of 64 kb/s/channel. It can also be noted from the MPEG-4 version 2 audio verification test [ISOI00], the quality measures of MPEG-2 AAC at 24 kb/s and MPEG-4 LD audio codec at 32 kb/s can be favorably compared. Moreover, the MPEG-4 LD audio codec [Herr98c] [Alla99] [Herr99] outperformed the ITU-T G.728 LD CELP [G728] for the case of coding both music and speech signals. However, as expected, the coding efficiency in the case of MPEG-4 LD codec is slightly reduced compared to its predecessors, MPEG-2 AAC and MPEG-4 GA. It should be noted that this reduction in the coding efficiency is attributed to the low coding delay achieved.

10.4.4.10 MPEG-4 Audio Error Robustness Tool One of the key issues in achieving reliable transmission over noisy and fast time-varying channels is the bit-rate scalability feature (discussed in Section 10.4.4.5). The bit-rate scalability enables flexible selection of coding features and dynamically adapts to the channel conditions and the varying channel capacity. However, the bit-rate scalability feature alone is not adequate for reliable transmission. The error resilience and error protection tools are also essential to obtain high quality audio. To this end, the MPEG-4 audio version 2 is fitted with codec-specific error robustness techniques [Purn99b] [ISOI00]. In this subsection, we will review the error robustness and equal and unequal error protection (EEP and UEP) tools in the MPEG-4 audio version 1. In particular, we discuss the *error resilience* [Sper00] [Sper02], *error protection* [Purn99b] [Mein01], and *error concealment* [Sper01] functionalities that are primarily designed for mobile applications.

The main idea behind the error resilience and protection tools is to provide better protection to sensitive and priority (important) bits. For instance, the audio frame header requires maximum error robustness; otherwise, transmission errors in the header will seriously impair the entire audio frame. The codewords corresponding to these priority bits are called the priority codewords (PCW). The error resilience tools available in the MPEG-4 audio version 2 are classified into three groups: the Huffman codeword reordering (HCR), the reversible variable length coding (RVLC), and the virtual codebooks (VCB11). In the HCR technique, some of the codewords, e.g., the PCWs, are sorted in advance and placed at known positions. First, a presorting procedure is employed that reorders the codewords based on their priority. The resulting PCWs are placed such that an error in one codeword will not affect the subsequent codewords. This can be achieved by defining segments of known length (L_{SEG}) and placing the PCWs at the beginning of these segments. The non-PCWs are filled into the gaps left by the PCWs, as shown in Figure 10.19.

The various applications of reversible variable length codes (RVLC) [Taki95] [Wen98] [Tsai01] in image coding have inspired researchers to consider them in error-resilient techniques for MPEG-4 audio. RVLC codes are used instead of Huffman codes for packing the scale factors in an AAC bitstream. The RVLC codes are (symmetrically) designed to enable both forward and backward decoding without affecting the coding efficiency. In particular, RVLCs allow instantaneous decoding in both directions that provides error robustness and significantly reduces the effects of bit errors in delay-constrained real-time applications. The next important tool employed for error resilience is the virtual codebook 11 (VCB11). Virtual codebooks are used to detect serious errors within spectral data [Purn99b] [ISOI00]. The error robustness techniques are codec specific (e.g., AAC and BSAC bitstreams). For example, AAC supports the HCR, the RVLC, and the VCB11 error-resilient tools. On the other hand, BSAC supports segmented binary arithmetic coding [ISOI00] to avoid error propagation within spectral data.

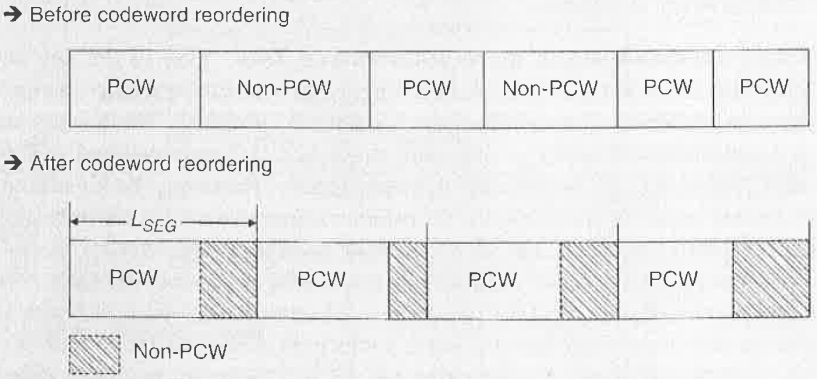


Figure 10.19. Huffman codeword reordering (HCR) algorithm to minimize error propagation in spectral data.

The MPEG-4 audio error protection tools include cyclic redundancy check (CRC), forward error correction (FEC), and interleaving. Note that these tools are inspired by some of the error correcting/detecting features inherent in the convolutional and block codes that essentially provide the controlled redundancy desired for error protection. Unlike the error-resilient tools that are limited only to the AAC and BSAC bitstreams, the error protection tools can be used in conjunction with a variety of MPEG-4 audio tools, namely, General Audio Coder (LTP and TwinVQ), Scalable Audio Coder, parametric audio coder (HILN), CELP, HVXC, and low-delay audio coder. Similar to the error-resilient tools, the first step in the EP tools is to reorder the bits based on their priority and error sensitiveness. The bits are sorted and grouped into different classes (usually 4 or 5) according to their error sensitivities. For example, consider that there are four error-sensitive classes (ESC), namely, ESC-0, ESC-1, ESC-2, and ESC-3. Usually, header bitstream or other very important bits that control the syntax and the global gain are included in the ESC-0. While the scale factors and spectral data (spectral envelope) are grouped in ESC-1 and ESC-2, respectively. The remaining side information and indices of MDCT coefficients are classified in ESC-3. After reordering (grouping) the bits, each error sensitive class receives a different error protection depending on the overhead allowed for each configuration. CRC and systematic rate-compatible punctured convolutional code (SRCPC) enable error detection and forward error correction (FEC). The SRCPC codes also aid in adjusting the redundancy rates in small steps. An interleaver is employed typically to deconcentrate or spread the burst errors. Shortened Reed-Solomon (SRS) codes are used to protect the interleaved data. Details on the design of Reed-Solomon codes for MPEG AAC are given in [Huan02]. For an in-depth treatment on the error correcting and error detecting codes refer to [Lin82] [Wick95].

10.4.4.11 MPEG-4 Speech Coding Tool Versus ITU-T Speech Standards It is noteworthy to compare the MPEG-4 speech coding tool against the ITU-T speech coding standards. While the latter applies source-filter configuration to model the speech parameters, the former employs a variety of techniques in addition to the traditional parametric representation. The MPEG-4 speech coding tool allows bit-rate scalability and real-time processing as well as applications related to storage media. The MPEG-4 speech coding tool incorporates algorithms such as the TwinVQ, the BSAC, and the HVXC/CELP. The MPEG-4 speech coding tool also accommodates multiple sampling rates. Error protection and error resilient techniques are provided in the MPEG-4 speech coding tool to obtain improved performance over error-prone channels. Other important features that distinguish the MPEG-4 tool from the ITU-T speech standards are the content-based interactivity and the ability to represent the audiovisual content as a set of objects.

10.4.4.12 MPEG-4 Audio Applications The MPEG-4 audio standard finds applications in low-bit-rate audio/speech compression, individual coding of natural and synthetic audio objects, low-delay coding, error-resilient transmission,

and real-time audio transmission over packet-switching networks such as the Internet [Diet96] [Liu99]. MPEG-4 tools allow parameterization of the acoustical properties of an audio scene, with features such as immersive audiovisual rendering (virtual 3-D environments [Kau98]), room acoustical modeling, and enhanced 3-D sound presentation. MPEG-4 finds interesting applications in remote robot control system design [Kim02b]. Streaming audio codecs have also been proposed as a result of the MPEG-4 standardization efforts.

Applications of MPEG-4 audio in DRM digital narrowband broadcasting (DNB) and digital multimedia broadcasting (DMB) are given in [Diet00] and [Grub01], respectively. The general audio coding tool provides the necessary infrastructure for the design of error-robust scalable coders [Mori00b] and delivers improved speech/audio quality [Moori00a]. The "bit rate scalability" and "error resilience/protection" tools of the MPEG-4 audio standard dynamically adapt to the channel conditions and the varying channel capacity. Other important application-oriented features of MPEG-4 audio include low-delay bi-directional audio transmission, content-based interactivity, and object-based representation. Real-time implementation of the MPEG-4 audio is reported in [Hilp00] [Mesa00] [Pena01].

10.4.4.13 Spectral Band Replication and Parametric Stereo Spectral band replication (SBR) [Diet02] and parametric stereo (PS) [Schu04] are the two new compression techniques recently added to the MPEG 4 audio standard [ISOI03c]. The SBR technique is used in conjunction with a conventional coder such as the MP3 or the MPEG AAC. The audio signal is divided into low- and high-frequency bands. The underlying core coder operates at a reduced sampling rate and encodes the low-frequency band. The SBR technique operates at the original sampling rate to estimate the spectral envelope associated with the input audio. The spectral envelope along with a set of control parameters are encoded and transmitted to the decoder. The control parameters contain information regarding the gain and the spectral envelope level adjustment of the high frequency components. At the decoder, the SBR reconstructs the high frequencies based on the transposition of the lower frequencies.

aacPlus v1 is the combination of AAC and SBR and is standardized as the MPEG 4 high-efficiency (HE)-AAC [ISOI03c] [Wolt03]. Relative to the conventional AAC, the MPEG 4 HE-AAC results in bit rate reductions of about 30% [Wolt03]. The SBR has also been used to enhance the performance of MP3 [Zieg02] and the MPEG layer 2 digital audio broadcasting systems [Gros03].

aacPlus v2 [Purn03] adds the parametric stereo coding to the MPEG 4 HE-AAC standard. In the PS encoding [Schu04], the stereo signal is represented as a monaural signal plus ancillary data that describe the stereo image. The stereo image is described using four different PS parameters, i.e., inter-channel intensity differences (IID), inter-channel phase differences (IPD), inter-channel coherence (IC), and overall phase difference (OPD). These PS parameters can capture the perceptually relevant spatial cues at bit rates as low as 10 kb/s [Bree04].

10.4.5 MPEG-7 Audio (ISO/IEC 15938-4)

MPEG-7 audio standard targets content-based multimedia applications [ISOI01b]. MPEG-7 audio supports a broad range of applications [ISOI01d] that include multimedia indexing/searching, multimedia editing, broadcast media selection, and multimedia digital library sorting. Moreover, it provides ways for efficient audio file retrieval and supports both text-based and context-based queries. It is important to note that MPEG-7 will not replace MPEG-1, MPEG-2 BC/LSF, MPEG-2 AAC, or MPEG-4. It is intended to provide complementary functionality to these MPEG standards. If MPEG-4 is considered as the first object-based multimedia representation standard, then MPEG-7 can be regarded as the first content-based standard that incorporates multimedia interfaces through *descriptions*. These descriptions are the means of linking the audio content features and attributes with the audio itself. Figure 10.20 presents an overview of the MPEG-7 audio standard. This figure depicts the various audio tools, features, and profiles associated with the MPEG-7 audio. Publications on the MPEG-7 Audio Standard include [Lind99] [Nack99a] [Nack99b] [Lind00] [ISOI01b] [ISOI01e] [Lind01] [Quac01] [Manj02].

Motivated by the need to exchange multimedia content through the World Wide Web, in 1996, the ISO/IEC MPEG workgroup worked on a project called "Multimedia Content Description Interface" (MCDI) – MPEG-7. A working draft was formed in December 1999 followed by a final committee draft in February 2001. Seven months later, MPEG-7 ISO/IEC 15938: Part 4 Audio, an international standard (IS) for *content-based multimedia applications* was published along with seven other parts of the MPEG-7 standard (Figure 10.20). Figure 10.20 shows a summary of various features, applications, and profiles specified by the MPEG-7 audio coding standard.

10.4.5.1 MPEG-7 Parts MPEG-7 defines the following eight parts [MPEG] (Figure 10.20): MPEG-7 Systems, MPEG-7 DDL, MPEG-7 Visual, MPEG-7 Audio, MPEG-7 MDS, MPEG-7 Reference Software (RS), MPEG-7 Conformance Testing (CT), and MPEG-7 Extraction and use of Descriptions.

MPEG-7 Systems (Part I) specifies the binary format for encoding MPEG-7 Descriptions; MPEG-7 DDL (Part II) is the language for defining the syntax of the Description Tools. MPEG-7 Visual (Part III) and MPEG-7 Audio (Part IV) deal with the visual and audio descriptions, respectively. MPEG-7 MDS (Part V) defines the structures for multimedia descriptions. MPEG-7 RS (Part VI) is a unique software implementation of certain parts of the MPEG-7 Standard with noninformative status. MPEG-7 CT (Part VII) provides the essential guidelines/procedures for conformance testing of MPEG-7 implementations. Finally, the eighth part, addresses the use and formulation of a variety of description tools that we will discuss later in this section.

In our discussion on MPEG-7 Audio, we refer to MPEG-7 DDL and MPEG-7 MDS parts quite regularly, mostly due to their interconnectivity within the MPEG-7 Audio Framework. Therefore, it is necessary that we introduce these two parts first, before we move on to the MPEG-7 Audio Description Tools.

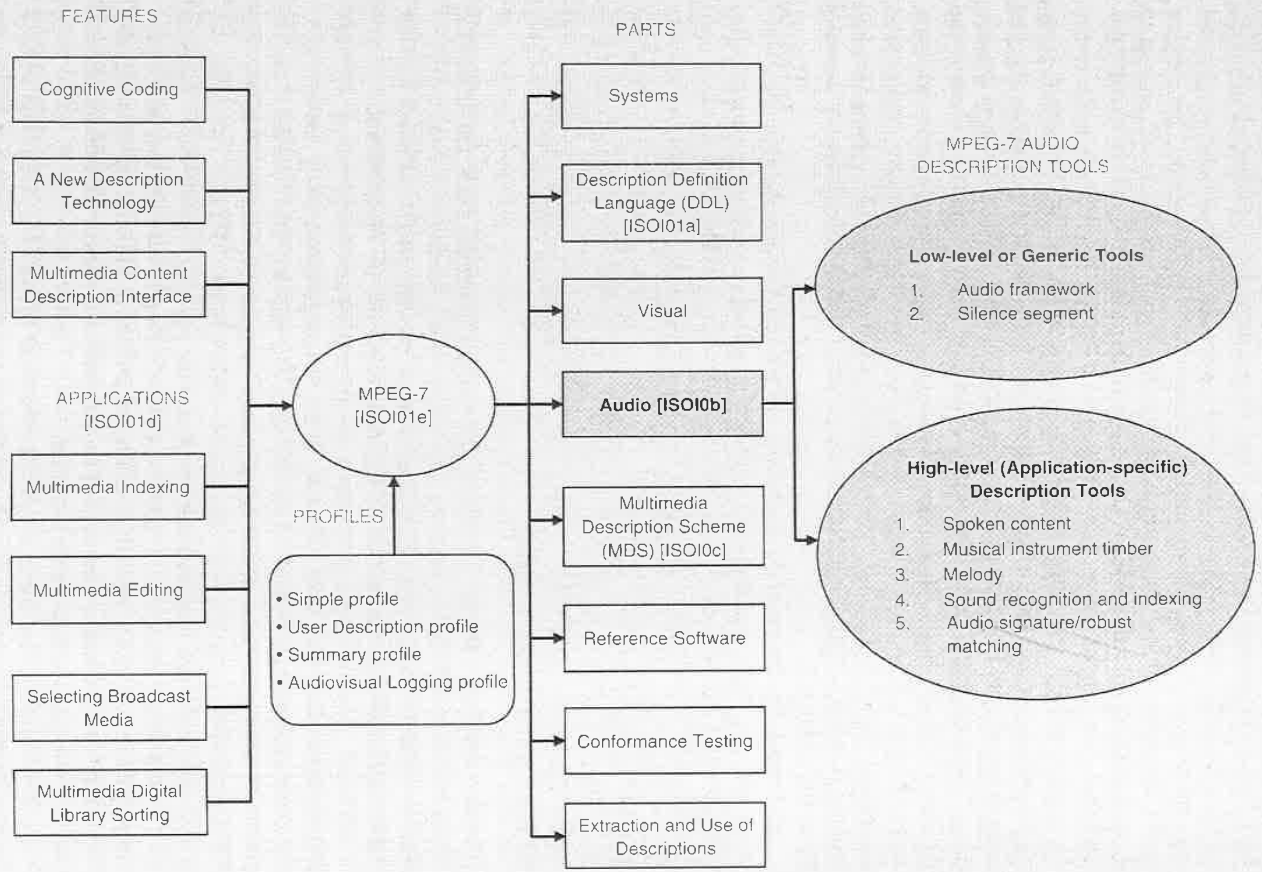
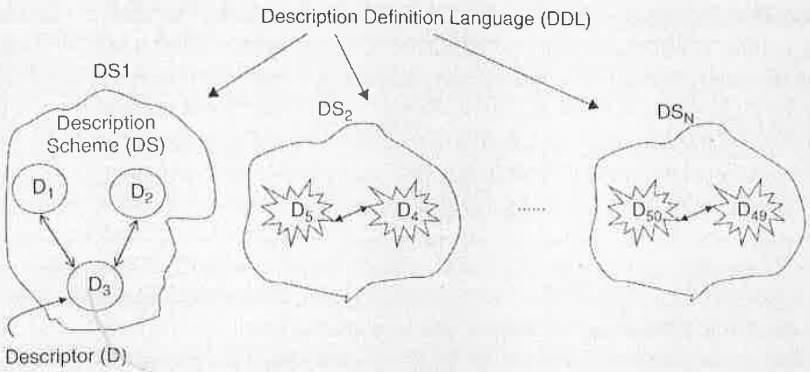


Figure 10.20. An overview of the MPEG-7 standard.



NOTE:
1. Descriptors (Ds) are the features and attributes associated with an audio waveform
2. The structure and the relationships among the descriptors are defined by a Description Scheme (DS)
3. Description Definition Language (DDL) defines the syntax necessary to create, extend, and combine avariety of DSs and Ds

Figure 10.21. Some essential building blocks of MPEG-7 standard: descriptors (Ds), description schemes (DSs), and description definition language (DDL).

MPEG-7 Description Definition Language (DDL) – Part II. We mentioned earlier that MPEG-7 incorporates multimedia interfaces through *descriptors*. These descriptors are the features and attributes associated with the audio. For example, descriptors in the case of MPEG-7 Visual part describe the visual features such as color, resolution, contour, mapping techniques, etc. A group of descriptors related in a manner suitable for a specific application, forms a *description scheme* (DS). The standard [ISOI01e] defines the description scheme as one that specifies a structure for the descriptors and semantics of their relationships.

MPEG-7 in its entirety has been built around these descriptors (Ds) and description schemes (DSs), and most importantly on a language called the description definition language (DDL). The DDL defines the syntax necessary to create, extend, and combine a variety of DSs and Ds. In particular, the DDL forms "the core part" of the MPEG-7 standard and will also be invoked by other parts (i.e., Visual, Audio, and MDS) to create new Ds and DSs. The DDL follows a set of programming rules/structure similar to the ones employed in the eXtensible Markup Language (XML). It is important to note that the DDL is not a modeling language but a schema language that is based on the WWW Consortium's XML schema [XML] [ISOI01e]. Several modifications were needed before adopting the XML schema language as the basis for the DDL. We refer to [XML] [ISOI01a] [ISOI01e] for further details on the XML schema language and its liaison with MPEG-7 DDL [ISOI01a].

MPEG-7 Multimedia Description Schemes (MDS) – Part V. Recall that a description scheme (DS) specifies structures for descriptors; similarly, a multimedia

description scheme (MDS) [ISOI01c] provides details on the structures for describing multimedia content (in particular audio, visual, and textual data). MPEG-7 MDS defines two classes of description tools, namely, the basic (or low-level) and multimedia (or high-level) tools [ISOI01c]. Figure 10.22 shows the classification of MDS elements. The *basic tools* specified by the MPEG-7 MDS are the generic entities, usually associated with simple descriptors, such as the basic data types, textual database, etc. On the other hand, the *high-level multimedia tools* deal with the content-specific entities that are complex and involve signal structures, semantics, models, efficient navigation, and access. The high-level (complex) tools are further subdivided into five groups (Figure 10.22), i.e., content description, content management, content organization, navigation and access, and user interaction.

Let us consider an example to better understand the concepts of DDL and MDS framework. Suppose that an audio signal, $s(n)$, is described using three descriptors, namely, spectral features D_1, parametric models D_2, and energy D_3. Similarly, visual $v(i, j)$ and textual content can also be described as shown in Table 10.5. We arbitrarily chose four description schemes (DS_1 through DS_4) that link these multimedia features (audio, visual, and textual) in a structured manner. This linking mechanism is performed through DDL, a schema language designed specifically for MPEG-7. From Table 10.5, the descriptors D_2, D_8, D_9 are related using the description scheme DS_2. The melody descriptor D_8 provides the melodic information (e.g., rhythmic, high-pitch, etc.), and the timbre descriptor D_9 represents some perceptual features (e.g., pitch/loudness details, bass/treble adjustments in audio, etc.). The parametric model descriptor D_2 describes the audio encoding model and related encoder details (e.g., MPEG-1 layer III, sampling rates, delay, bit rates, etc.). While the descriptor D_2 provides details on the encoding procedure, the descriptors D_8 and D_9 describe audio morphing, echo/reverberation, tone control, etc.

MPEG-7 Audio – Part IV. MPEG-7 Audio represents part IV of the MPEG-7 standard and provides structures for describing the audio content. Figure 10.23 shows the organization of MPEG-7 audio framework.

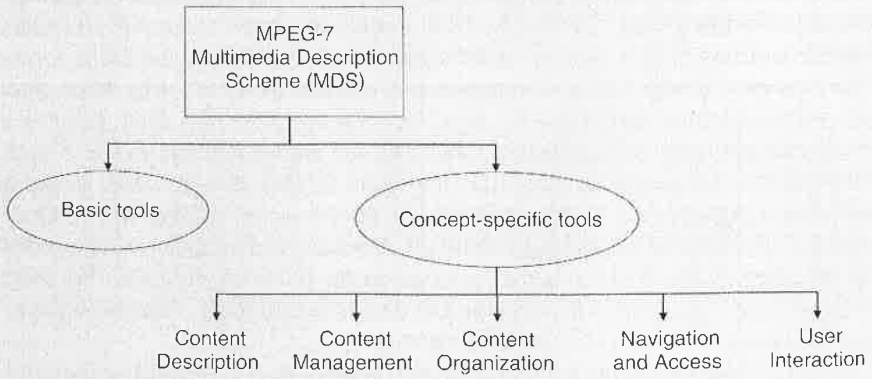


Figure 10.22. Classification of multimedia description scheme (MDS) tools.

Table 10.5. A hypothetical example that gives a broader perspective on multimedia descriptors; i.e., audio, visual, and textual features to describe a multimedia content.

Group	Descriptors	Description schemes
Audio content, $s(n)$	D_1: Spectral features D_2: Parametric models D_3: Energy of the signal	DS_1: D_1, D_3
Visual content, $v(i, j)$	D_4: Color	DS_2: D_2, D_8, D_9
	D_5: Shape	DS_3: DS_2, D_4, D_5
Textual descriptions	D_6: Title of the clip	DS_4: DS_1, D_6, D_7
	D_7: Author information D_8: Melody details D_9: Timbre details	

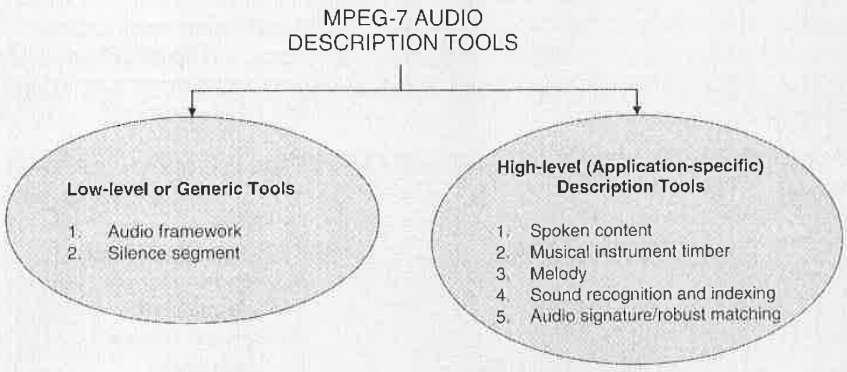


Figure 10.23. MPEG-7 audio description tools.

10.4.5.2 MPEG-7 Audio Versions and Profiles New extensions (Amendment 1) for the existing MPEG-7 Audio are being considered. Some of the extensions are in the areas of application-specific spoken content, tempo description, and specification of precision for low-level data types. This new amendment will be standardized as MPEG-7 Audio Version 2 (Final drafts of International Standard (FDIS) for Version 2 were finalized in March 2003).

Although many description tools are available in MPEG-7 audio, it is not practical to implement all of them in a particular system. MPEG-7 Version 1 therefore defines four complexity-ranked profiles (Figure 10.20) intended to help system designers in the task of tool subset selection. These include simple profile, user description profile, summary profile, and audiovisual logging profile.

10.4.5.3 MPEG-7 Audio Description Tools The MPEG-7 Audio framework comprises two main categories; namely, *generic* tools and a set of *application-specific* tools (see Figure 10.20 and Figure 10.23).

10.4.5.3.1. Generic Tools The generic toolset consists of 17 low-level audio descriptors and a silence segment descriptor (Table 10.6).

MPEG-7 Audio Low-level Descriptors. MPEG-7 audio [ISOI01b] defines two ways of representing the low-level audio features, i.e., segmenting and sampling. In segmentation, usually, common datatypes or scalars are grouped together (e.g., energy, power, bit rate, sampling rate, etc.). On the other hand, sampling enables discretization of audio features in a vector form (e.g., spectral features, excitation samples, etc.). Recently, a unified framework called the *scalable series* [Lind99] [Lind00] [ISOI01b] [Lind01] has been proposed to manipulate these discretized values. This is somewhat similar to MPEG-4 scalable audio coding that we discussed in Section 10.4.4. A list of low-level audio descriptors defined by the MPEG-7 Audio standard [ISOI01b] is summarized in Table 10.6. These

Table 10.6. Low-level audio descriptors (17 in number) and the silence descriptor supported by the MPEG-7 generic toolset [ISOI01b].

	Generic toolset	Descriptors
Low-level audio descriptors group	1. Basic	D_1: Audio waveform
	2. Basic spectral	D_2: Power
		D_3: Spectrum envelope
		D_4: Spectral centroid
		D_5: Spectral spread
		D_6: Spectral flatness
	3. Signal parameters	D_7: Harmonicity
		D_8: Fundamental frequency
	4. Spectral basis	D_9: Spectrum basis
		D_{10}: Spectrum projection
	5. Timbral spectral	D_{11}: Harmonic spectral centroid
		D_{12}: Harmonic spectral deviation
		D_{13}: Harmonic spectral spread
		D_{14}: Harmonic spectral variation
		D_{15}: Spectral centroid
	6. Timbral temporal	D_{16}: Log attack time
		D_{17}: Temporal centroid
Silence	7. Silence segment	D_{18}: Silence descriptor

descriptors can be classified into the following groups: basic, basic spectral, signal parameters, spectral basis, timbral spectral, and timbral temporal.

MPEG-7 Silence Segment. The MPEG-7 silence segment attaches a semantic of silence to an audio segment. The silence descriptor provides ways to specify threshold levels (e.g., the level of silence).

10.4.5.3.2. High-Level or Application-Specific MPEG-7 Audio Tools Besides the aforementioned generic toolset, the MPEG-7 audio standard describes five specialized high-level tools (Table 10.7). These application-specific description tools can be grouped as spoken content, musical instrument, melody, sound recognition/indexing, and robust audio matching.

Spoken Content Description Tool (SC-DT). The SC-DT provides descriptions of spoken words in an audio clip, thereby enabling speech recognition and speech parameter indexing/searching. Spoken content lattice and spoken content header are the two important parts of the SC-DT (see Table 10.7). While the SC header carries the lexical information (i.e., wordlexicon, phonelexicon, ConfusionInfo, and SpeakerInfo descriptors), the SC-lattice DS represents lattice-structures to connect words or phonemes chosen from the corresponding lexicon. The idea of using *lattice structures* in the SC-lattice DS is similar to the one employed in a typical continuous automatic speech recognition scenario [Rabi89] [Rabi93].

Musical Instrument Timbre Description Tool (MIT-DT). The MIT-DT describes the timbre features (i.e., perceptual attributes) of sounds from musical instruments. Timbre can be defined as the collection of perceptual attributes that make two

Table 10.7. Application-specific audio descriptors and description schemes [ISOI01b].

High-level descriptor toolset		Descriptor details
SC-DT	1. SC-header	D_1: Word lexicon
		D_2: Phone lexicon
		D_3: Confusion info
		D_4: Speaker info
	2. SC-lattice DS	Provides structures to connect or link the words/phonemes in the lexicon.
MIT-DT	3. Timbre (perceptual) features of musical instruments	D_1: Harmonic Instrument Timbre
		D_2: Percussive Instrument Timbre
M-DT	4. Melody features	DS_1: Melody contour
		DS_2: Melody sequence
SRI-DT	5. Sound recognition and indexing application	D_1: Sound Model State Path
		D_2: Sound Model State Histogram
		DS_1: Sound model
		DS_2: Sound classification model
AS-DT	6. Robust audio identification	DS_1: Audio signature DS

audio clips having the same pitch and loudness sound different [ISOI01b]. Musical instrument sounds, in general, can be classified as harmonic-coherent-sustained, percussive-nonsustained, nonharmonic-coherent-sustained, and non-coherent-sustained. The standard defines descriptors for the first two classes of musical sounds (Table 10.7). In particular, MIT-DT defines two descriptors, namely, the harmonic instrument timbre (HIT) descriptor and the percussive instrument timbre (PIT) descriptor. The HIT descriptor was built on the four harmonic low-level descriptors (i.e., D_{11} through D_{14} in Table 10.6) and the Logattacktime descriptor. On the other hand, the PIT descriptor is based on the combination of the timbral temporal low-level descriptors (i.e., Logattacktime and Temporalcentroid) and the spectral centroid descriptor.

Melody Description Tool (M-DT). The M-DT represents the melody details of an audio clip. The melodycontourDS and the melodysequenceDS are the two schemes included in M-DT. While the former scheme enables simple and robust melody contour representation, the latter approach involves detailed and expanded melody/rhythmic information.

Sound Recognition and Indexing Description Tool (SRI-DT). The SRI-DT is on automatic sound identification/recognition and indexing. Recall that the SC-DT employs lexicon descriptors (Table 10.7) for SC recognition in an audio clip. In the case of SRI, classification/indexing of sound tracks are achieved through sound models. These models are constructed based on the spectral basis low-level descriptors, i.e., spectral basis (D_9) and spectral projection (D_{10}), listed in Table 10.6. Two descriptors, namely the sound model state path descriptor and the sound model state histogram descriptor, are defined to keep track of the active paths in a trellis.

Robust Audio Identification and Matching. Robust matching and identification of audio clips is one of the important applications of MPEG-7 audio standard [ISOI01d]. This feature is enabled by the low-level spectral flatness descriptor (Table 10.6). A description scheme, namely, the Audio Signature DS defines the semantics and structures for the spectral flatness descriptor. Hellmuth *et al.* [Hell01] proposed an advanced audio identification procedure based on content descriptions.

10.4.5.4 MPEG-7 Audio Applications Being the first metadata standard, MPEG-7 audio provides new ideas for audio-content indexing and archiving [ISOI01d]. Some of the applications are in the areas of multimedia searching, audio file indexing, sharing and retrieval, and media selection for digital audio broadcasting (DAB). We discussed most of these applications while addressing the high-level audio descriptors and description schemes. A summary of these applications follows. Unlike in an automatic speech recognition scenario where word or phoneme lattices (based on feature vectors) are employed for identifying speech, in MPEG-7 these lattice structures are denoted as Ds and DSs. These description data enable *spoken content retrieval*. MPEG-7 audio version 2 includes new tools and specialized enhancements to spoken content search. *Musical instrument timbre search* is another important application that targets content-based editing. *Melody search* enables query by humming [Quac01]. *Sound recognition/indexing* and *audio identification/fingerprinting* form two other important

applications of the MPEG-7. We will next address the concepts of "interoperability" and "universal multimedia access" (UMA) in the context of the new work initiated by the ISO/IEC MPEG workgroup in June 2000, called the *Multimedia Framework – MPEG* 21 [Borm03].

10.4.6 MPEG-21 Framework (ISO/IEC-21000)

Motivated by the need for a standard that enables multimedia content access and distribution, the ISO/IEC MPEG workgroup addressed the 21st Century Multimedia Framework – MPEG-21: ISO/IEC 21000 [Spen01] [ISOI02a] [ISOI03a] [ISOI03b] [Borm03] [Burn03]. This multimedia standard should be interoperable and highly automated [Borm03]. The MPEG-21 multimedia framework envisions creating a platform that encompasses a great deal of functionalities for both content-users and content-creators/providers. Some of these functions include the multimedia resource delivery to a wide range of networks and terminals (e.g., personal computers (PCs), PDAs and other digital assistants, mobile phones, third-generation cellular networks, digital audio/video broadcasting (DAB/DVB), HDTVs, and several other home entertainment systems); protection of intellectual property rights through digital rights management (DRM) systems.

Content creators and service providers face several challenging tasks in order to satisfy simultaneously the conflicting demands of "interoperability" and "intellectual property management and protection" (IPMP). To this end, MPEG-21 defines a multimedia framework that comprises seven important parts [ISOI02a], as shown in Table 10.8. Recall that the MPEG-7 ISO/IEC-15938 standard defines a fundamental unit called "Descriptors" (Ds) to define/declare the features and attributes of multimedia content. In a manner analogous to this, MPEG-21 ISO/IEC-21000: Part 1 defines a basic unit called the "Digital Item" (DI). Besides DI, MPEG-21 specifies another entity called the "User" interaction [ISOI02a] [Burn03] that provides details on how each "User" interacts with other users via objects called the "Digital Items." Furthermore, MPEG-21 Parts 2 and 3 define the declaration and identification of the DIs, respectively (see Table 10.8). MPEG-21 ISO/IEC-21000 Parts 4 through 6 enables interoperable digital content distribution and transactions that take into account the IPMP requirements. In particular, a machine-readable language called the Rights Expression Language (REL) is specified in MPEG-21 ISO/IEC-21000: Part 5 that defines the rights and permissions for the access and distribution of multimedia resources across a variety of heterogeneous terminals and networks. MPEG-21 ISO/IEC-21000: Part 6 defines a dictionary called the Rights Data Dictionary (RDD) that contains information on content protection and rights.

MPEG-7 and MPEG-21 standards provide an open framework on which one can build application-oriented interfaces or tools that satisfy a specific criterion (e.g., a query, an audio file indexing, etc.). In particular, the MPEG-7 standard provides an *interface* for indexing, accessing, and distribution of multimedia content; and the MPEG-21 defines an *interoperable framework* to access the multimedia content.

Table 10.8. MPEG-21 multimedia framework and the associated parts [ISOI02a].

Parts in the MPEG-21: ISO/IEC 21000 Standard [ISOI02a]		Details
Part 1	Vision, technologies, and strategy	Defines the vision, requirements, and applications of the standard; and provides an overview of the multimedia framework. Introduces two new terms, i.e., *digital item* (DI) and *user interaction*.
Part 2	Digital item declaration	Defines the relationship between a variety of multimedia resources and provides information regarding the declaration of DIs.
Part 3	Digital item identification	Provides ways to identify different types of digital items (DIs) and descriptors/description schemes (Ds/DSs) via uniform resource identifiers (URIs).
Part 4	IPMP	Defines a framework for the intellectual property management and protection (IPMP) that enables interoperability.
Part 5	Rights expression language	A syntax language that enables multimedia content distribution in a way that protects the digital content. The rights and the permissions are expressed or declared based on the terms defined in the rights data dictionary.

Table 10.8. (*continued*)

Parts in the MPEG-21: ISO/IEC 21000 Standard [ISOI02a]		Details
Part 6	Rights data dictionary	A database or a dictionary that contains the information regarding the rights and permissions to protect the digital content.
Part 7	Digital item adaptation	Defines the concept of an adapted digital item.

Until now, our focus was primarily on ISO/IEC MPEG Audio Standards. In the next few sections, we will attend to company-oriented perceptual audio coding algorithms, i.e., the Sony Adaptive Transform Acoustic Coding (ATRAC), the Lucent Technologies Perceptual Audio Coder (PAC), the Enhanced PAC (EPAC), the Multichannel PAC (MPAC), Dolby Laboratories AC-2/AC-2A/AC-3, Audio Processing Technology (APT-x100), and the Digital Theater Systems (DTS) Coherent Acoustics (CA) encoder algorithms.

10.4.7 MPEG Surround and Spatial Audio Coding

MPEG spatial audio coding began receiving attention during the early 2000s [Fall01] [Davis03]. Advances in joint stereo coding of multichannel signals [Herr04b], binaural cue coding [Fall01], and the success of the recent low complexity parametric stereo encoding in MPEG 4 HE-AAC/PS standard [Schu04] generated interest in the MPEG surround and spatial audio coding [Herr04a] [Bree05]. Unlike the discrete 5.1-channel encoding as used in Dolby Digital or DTS, the MPEG spatial audio coding, captures the "spatial image" of a multichannel audio signal. The spatial image is represented using a compact set of parameters that describe the perceptually relevant differences among the channels. Typical parameters include the interchannel level difference (ICLD), the interchannel coherence (ICC), and the interchannel time difference (ICTD). The multichannel signal is first downmixed to a stereo signal and then a conventional MP3 coder is used. Spatial image parameters are computed using the binaural cue coding (BCC) technique and are transmitted to the decoder as side information [Herr04a]. At the decoder, a one-to-two (OTT) or two-to-three (TTT) channel mapping is used to synthesize the multichannel surround sound.

10.5 ADAPTIVE TRANSFORM ACOUSTIC CODING (ATRAC)

The ATRAC algorithm, developed by Sony for use in its rewriteable MiniDisc system [Yosh94], combines subband and transform coding to achieve nearly CD-quality coding of 44.1 kHz 16-bit PCM input data at a bit rate of 146 kb/s per

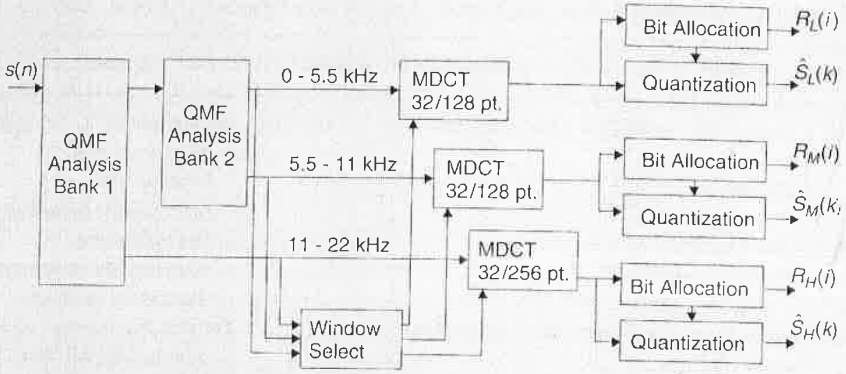


Figure 10.24. Sony ATRAC (embedded in MiniDisc, SDDS).

channel [Tsut98]. Using a tree-structured QMF analysis bank (Section 6.5), the ATRAC encoder (Figure 10.24) first splits the input signal into three subbands of 0–5.5 kHz, 5.5–11 kHz, and 11–22 kHz. Like MPEG-1 layer III, the ATRAC QMF bank is followed by signal-adaptive MDCT analysis in each subband. Next, a window-switching scheme is employed that can be summarized as follows. During steady-state input periods, high-resolution spectral analysis is attained using 512 sample blocks (11.6 ms). During input attack or transient periods, short block sizes of 1.45 ms in the high-frequency band and 2.9 ms in the low- and mid-frequency bands are used for pre-echo cancellation.

After MDCT analysis, spectral components are clustered into 52 nonuniform subbands (block floating units or BFUs) according to critical band spacing. The BFUs are block-companded, quantized, and encoded according to a psychoacoustically derived bit allocation. For each analysis frame, the ATRAC encoder transmits quantized MDCT coefficients, subband window lengths, BFU scalefactors, and BFU word lengths to the decoder. Like the MPEG family, the ATRAC architecture decouples the decoder from psychoacoustic analysis and bit-allocation details. Evolutionary improvements in the encoder bit allocation strategy are therefore possible without modifying the decoder structure. An added benefit of this architecture is asymmetric complexity, which enables inexpensive decoder implementations.

Suggested bit allocation techniques for ATRAC are of lower complexity than those found in other standards since ATRAC is intended for low-cost, battery-powered devices. One proposed method distributes bits between BFUs according to a weighted combination of fixed and adaptive bit allocations [Tsut96]. For the k-th BFU, bits are allocated according to the relation

$$r(k) = \alpha r_a(k) + (1 - \alpha) r_f(k) - \beta, \tag{10.10}$$

where $r_f(k)$ is a fixed allocation, $r_a(k)$ is a signal-adaptive allocation, the parameter β is a constant offset computed to guarantee a fixed bit rate, and the parameter α is a tonality estimate ranging from 0 (noise-like) to 1 (tone-like). The fixed

allocations, $r_f(k)$, are the same for all inputs and concentrate more bits at the lower frequencies. The signal adaptive bit allocations, $r_a(k)$, assign bits according to the strength of the MDCT components. The effect of Eq. (10.10) is that more bits are allocated to BFUs containing strong peaks for tonal signals. For noise-like signals, bits are allocated according to a fixed allocation rule, with low bands receiving more bits than high bands.

Sony Dynamic Digital Sound (SDDS). In addition to providing near CD-quality on a MiniDisc medium, the ATRAC algorithm has also been deployed as the core of Sony's digital cinematic sound system, SDDS. SDDS integrates eight independent ATRAC modules to carry the program information for the left (L), left center (LC), center (C), right center (RC), right (R), subwoofer (SW), left surround (LS), and right surround (RS) channels typically present in a modern theater. SDDS data is recorded using optical black and white dot-matrix technology onto two thin strips along the right and left edges of the film, outside of the sprocket holes. Each edge contains four channels. There are 512 ATRAC bits per channel associated with each movie frame, and each optical data frame contains a matrix of 52 × 192 bits [Yama98]. SDDS data tracks do not interfere with or replace the existing analog sound tracks. Both Reed-Solomon error correction and redundant track information are delayed by 18 frames and employed to make SDDS robust to bit errors introduced by run-length scratches, dust, splice points, and defocusing during playback or film printing. Analog program information is used as a backup in the event of uncorrectable digital errors.

10.6 LUCENT TECHNOLOGIES PAC, EPAC, AND MPAC

The pioneering research contributions on perceptual entropy [John88b], monophonic PXFM [John88a], stereophonic PXFM [John92a], and ASPEC [Bran91] strongly influenced not only the MPEG family architecture but also evolved at AT&T Bell Laboratories into the Perceptual Audio Coder (PAC). The PAC algorithm eventually became property of Lucent. AT&T, meanwhile, became active in the MPEG-2 AAC research and standardization. The low-complexity profile of AAC became the AT&T coding standard.

Like the MPEG coders, the Lucent PAC algorithm is flexible in that it supports monophonic, stereophonic, and multiple channel modes. In fact, the bit stream definition will accommodate up to 16 front side, 7 surround, and 7 auxiliary channel pairs, as well as 3 low-frequency effects (LFE or subwoofer) channels. Depending upon the desired quality, PAC supports several bit rates. For a modest increase in complexity at a particular bit rate, improved output quality can be realized by enabling enhancements to the original system. For example, whereas 96 kb/s output was judged to be adequate with stereophonic PAC, near CD quality was reported at 56–64 kb/s for stereophonic enhanced PAC [Sinh98a].

10.6.1 Perceptual Audio Coder (PAC)

The original PAC system described in [John96c] achieves very-high-quality coding of stereophonic inputs at 96 kb/s. Like the MPEG-1 layer III and the ATRAC,

the PAC encoder (Figure 10.25) uses a signal-adaptive MDCT filter bank to analyze the input spectrum with appropriate frequency resolution. A long window of 2048 points (1024 subbands) is used during steady-state segments, or else a series of short 256-point windows (128 subbands) is applied for segments containing transients or sharp attacks. In contrast to MPEG-1 and ATRAC, however, PAC relies on the MDCT alone rather than a hybrid filter-bank structure, thus realizing a complexity reduction. As noted previously [Bran88a] [Mahi90], the MDCT lends itself to compact representation of stationary signals, and a 2048-point block size yields sufficiently high-frequency resolution for most sources. This segment length was also associated with the maximum realizable coding gain as a function of block size [Sinh96]. Filter-bank resolution switching decisions are made on the basis of PE (high complexity) or signal energy (low complexity) criteria.

The PAC perceptual model derives noise masking thresholds from filter-bank output samples in a manner similar to MPEG-1 psychoacoustic model recommendation 2 [ISOI92] and the PE calculation in [John88b]. The PAC model, however, accounts explicitly for both simultaneous and temporal masking effects. Samples are grouped into 1/3 critical band partitions, tonality is estimated in each band, and then time and frequency spreading functions are used to compute a masking threshold that can be related to the filter-bank outputs. One can observe that PAC realizes some complexity reduction relative to MPEG by avoiding parallel frequency analysis structures for quantization and perceptual modeling. The masking thresholds are used to select one of 128 exponentially distributed quantization step sizes in each of 49 or 14 coder bands (analogous to ATRAC BFUs) in high-resolution and low-resolution modes, respectively. The coder bands are quantized using an iterative rate control loop in which thresholds are adjusted to satisfy simultaneously bit-rate constraints and an equal loudness criterion that attempts to shape quantization noise such that its absolute loudness is constant relative to the masking threshold. The rate control loop allows time-varying instantaneous bit rates so that additional bits are available in times of peak demand, much like the bit reservoir of MPEG-1 layer III. Remaining statistical redundancies are removed from the stream of quantized spectral samples prior to bit stream formatting using eight structured, multidimensional Huffman codebooks. These codebooks are applied to DPCM-encoded quantizer outputs. By clustering coder bands into sections and selecting only one codebook per section, the system minimizes the overhead.

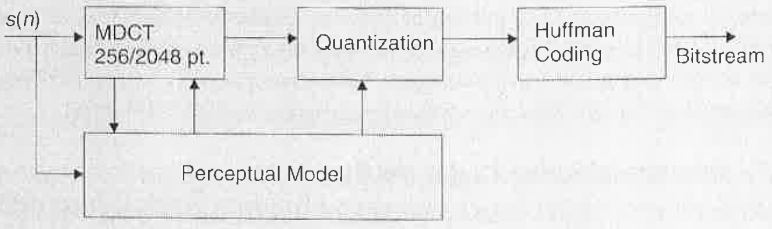


Figure 10.25. Lucent Technologies Perceptual Audio Coder (PAC).

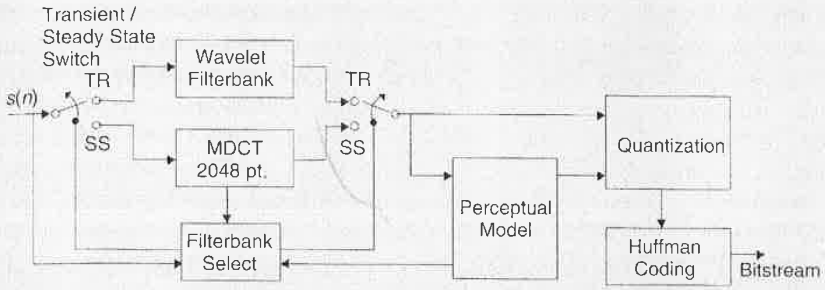


Figure 10.26. Lucent Technologies Enhanced Perceptual Audio Coder (EPAC).

10.6.2 Enhanced PAC (EPAC)

In an effort to enhance PAC output quality at low bit rates, Sinha and Johnston [Sinh96] introduced a novel signal-adaptive MDCT/WP[1] switched filter bank scheme that resulted in nearly transparent coding for CD-quality source material at 64 kb/s per stereo pair. EPAC (Figure 10.26) is unique in that it switches between two distinct filter banks rather than relying upon hybrid [Tsut98] [ISOI92] or nonuniform cascade [Prin95] structures.

A 2048-point MDCT decomposition is applied normally, during "stationary" periods. EPAC switches to a tree-structured wavelet packet (WP) decomposition matched to the auditory filter bank during sharp transients. Switching decisions occur every 25 ms, as in PAC, using either PE or energy criteria. The WP analysis offers the dual advantages of more compact signal representation during transient periods than MDCT, as well as improved time resolution at high frequencies for accurate estimation of the time/frequency distribution of masking power contained in sharp attacks. In contrast to the uniform time-frequency tiling associated with MDCT window-length switching schemes (e.g., [ISOI92] [Bran94a]), the EPAC WP transform (tree-structured QMF bank) achieves a nonuniform time-frequency tiling. For a suitably designed analysis wavelet and tree-structure, an improvement in time resolution is restricted to the high-frequency regions of interest, while good-frequency resolution is maintained in the low-frequency subbands. The EPAC WP filter bank was specifically designed for time-localized impulse responses at high frequencies to minimize the temporal spread of quantization noise (pre-echo). Novel start and stop windows are inserted between analysis frames during switching intervals to mitigate boundary effects associated with the MDCT-to-WP and WP-to-MDCT transitions. Other than the enhanced filter bank, EPAC is identical to PAC. In subjective tests involving 12 expert and nonexpert listeners with difficult castanets and triangle test signals, EPAC outperformed PAC for a 64 kb/s per stereo pair by an average of 0.4–0.6 on a five-point quality scale.

10.6.3 Multichannel PAC (MPAC)

Like the MPEG, AC-3, and SDDS systems, the PAC algorithm also extends its monophonic processing capabilities into stereophonic and multiple-channel

[1] See Chapter 8, Sections 8.2 and 8.3, for descriptions on wavelet filter bank and WP transforms.

modes. Stereophonic PAC computes individual masking thresholds for the left, right, mono, and stereo (L, R, M = L + R, and S = L − R) signals using a version of the monophonic perceptual model that has been modified to account for binary-level masking differences (BLMD), or binaural unmasking effects [Moor77]. Then, monaural PAC methods encode either the signal pairs (L, R) or (M, S). In order to minimize the overall bit rate, however, a LR/MS switching procedure is embedded in the rate-control loop such that different encoding modes (LR or MS) can be applied to the individual coder bands on the same analysis frame.

In the MPAC 5-channel configuration, composite coding modes are available for the front side left, center, right, and left and right surround (L, C, R, Ls, and Rs) channels. On each frame, the composite algorithm works as follows: First, appropriate window-switched filter-bank frequency resolution is determined separately for the front, side, and surround channels. Next, the four signal pairs LR, MS, LsRs, and MsSs (Ms = Ls + Rs, Ss = Ls − Rs) are generated. The MPAC perceptual model then computes individual BLMD-compensated masking thresholds for the eight LR and MS signals, as well as the center channel, C. Once thresholds have been obtained, a two-step coding process is applied. In step 1, a minimum PE criterion is first used to select either MS or LR coding for the front, side, and surround channel groups in each coder band. Then, step 2 applies interchannel prediction to the quantized spectral samples. The prediction residuals are quantized such that the final quantization noise satisfies the original masking thresholds for each channel (LR or MS). The interchannel prediction schemes are summarized in [Sinh98a]. In pursuit of a minimum bit rate, the composite coding algorithm may elect to utilize either step 1 or step 2, both step 1 and step 2, or neither step 1 nor step 2. Finally, the composite perceptual model computes a global masking threshold as the maximum of the five individual thresholds, minus a safety margin. This threshold is phased in gradually for joint coding when the bit reservoir drops below 20% [Sinh98a]. The safety margin magnitude depends upon the bit reservoir state. Composite modes are applied separately for each coder band on each analysis frame. In terms of performance, the MPAC system was found to produce the best quality at 320 kb/s for 5 channels during a recent ISO test of multichannel algorithms [ISOII94].

Applications. Both 128 and 160 kb/s stereophonic versions of PAC were considered for standardization in the U.S. Digital Audio Radio (DAR) project. In an effort to provide graceful degradation and extend broadcast range in the presence of heavy fading associated with fringe reception areas, perceptually motivated unequal error protection (UEP channel coding) schemes were examined in [Sinh98b]. The proposed scheme ranks bit stream elements into two classes of perceptual importance. Bit stream parameters associated with center channel information and certain mid-frequency subbands are given greater channel protection (class 1) than other parameters (class 2). Subjective tests revealed a strong preference for UEP over equal error protection (EEP), particularly when bit error rates (BER) exceeded 2×10^{-4}. For network applications, acceptable PAC output quality at bit rates as low as 12–16 kb/s per channel in conjunction with the availability of JAVA PAC decoder implementations are

reportedly increasing PAC deployment among suppliers of Internet audio program material [Sinh98a]. MPAC has also been considered for cinematic and advanced television applications. Real-time PAC and EPAC decoder implementations have been demonstrated on 486-class PC platforms.

10.7 DOLBY AUDIO CODING STANDARDS

Since the late 1980s, Dolby Laboratories has been active in perceptual audio coding research and standardization, and Dolby researchers have made numerous scientific contributions within the collaborative framework of MPEG audio. On the commercial front, Dolby has developed the AC-2 and the AC-3 algorithms [Fiel91] [Fiel96].

10.7.1 Dolby AC-2, AC-2A

The AC-2 [Davi90] [Fiel91] is a family of single-channel algorithms operating at bit rates between 128 and 192 kb/s for 20 kHz bandwidth input sampled at 44.1 or 48 kHz. There are four available AC-2 variants, all of which share an architecture in which the input is mapped to the frequency domain by an evenly stacked TDAC filter bank [Prin86] with a novel parametric Kaiser-Bessel analysis window (Section 6.7) optimized for improved stop-band attenuation relative to the sine window. The evenly stacked TDAC differs from the oddly stacked MDCT in that the evenly stacked low-band filter is half-band, and its magnitude response wraps around the fold-over frequency (see Chapter 6). A unique mantissa-exponent coding scheme is applied to the TDAC transform coefficients. First, sets of frequency-adjacent coefficients are grouped into blocks (subbands) of roughly critical bandwidth. For each block, the maximum is identified and then quantized as an exponent in terms of the number of left shifts required until overflow occurs. The collection of exponents forms a stair-step spectral envelope having 6 dB (left shift = multiply by 2 = 6.02 dB) resolution, and normalizing the transform coefficients by the envelope generates a set of mantissas. The envelope approximates the short-time spectrum, and therefore a perceptual model uses the exponents to compute both a fixed and a signal-adaptive bit allocation for the mantissas on each frame.

As far as details on the four AC-2 variants are concerned, two versions are designed for low-complexity, low-delay applications, and the other two for higher quality at the expense of increased delay or complexity. In version 1, a 128-sample (64-channel) filter bank is used, and the coder operates at 192 kb/s per channel, resulting in high-quality output with only 7-ms delay at 48 kHz. Version 2 is also for low-delay applications with improved quality at the same bit rate, and it uses the same filter bank but exploits time redundancy across block pairs, thus increasing delay to 12 ms. Version 3 achieves similar quality with the reduced rate of 128 kb/s per channel at the expense of longer delay (45 ms) by using a 512-sample (256 channel) filter bank to improve steady-state coding gain. Finally, version 4 (the AC-2A [Davi92] algorithm) employs a switched 128/512-point TDAC filter bank to improve quality for transient signals while maintaining

high coding gain for stationary signals. A 320-sample bridge window preserves PR filter bank properties during mode switching, and a transient detector consisting of an 8-kHz Chebyshev highpass filter is responsible for switching decisions. Order of magnitude peak level increases between 64-sample sub-blocks at the filter output are interpreted as transient events. The Kaiser window parameters used for the KBD windows in each of the AC-2 algorithms appeared in [Fiel96]. For all four algorithms, the AC-2 encoder multiplexes spectral envelope and mantissa parameters into an output bitstream, along with some auxiliary information. Byte-wide Reed-Solomon ECC allows for correction of single byte errors in the exponents at the expense of 1% overhead, resulting in good performance up to a BER of 0.001.

One AC-2 feature that is unique among the standards is that the perceptual model is backward adaptive, meaning that the bit allocation is not transmitted explicitly. Instead, the AC-2 decoder extracts the bit allocation from the quantized spectral envelope using the same perceptual model as the AC-2 encoder. This structure leads to a significant reduction of side information and induces a symmetric encoder/decoder complexity, which was well suited to the original AC-2 target application of single point-to-point audio transport. An example single point-to-point system now using low-delay AC-2 is the DolbyFAX™, a full-duplex codec that carries simultaneously two channels in both directions over four ISDN "B" links for film and TV studio distance collaboration. Low-delay AC-2 codecs have also been installed on 950-MHz wireless digital studio transmitter links (DSTL). The AC-2 moderate delay and AC-2A algorithms have been used for both network and wireless broadcast applications such as cable and direct broadcast satellite (DBS) television. The AC-2A is the predecessor to the now popular multichannel AC-3 algorithm. As the next section will show, the AC-3 coder has inherited and enhanced several facets of the AC-2/AC-2A architecture. In fact, the AC-2 encoder is nearly identical to (one channel of) the simplified AC-3 encoder shown in Figure 10.27, except that AC-2 does not transmit explicitly any perceptual model parameters.

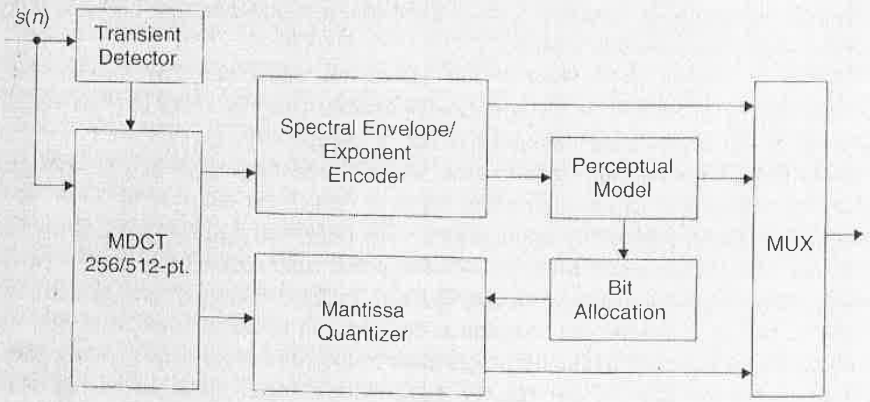


Figure 10.27. Dolby Laboratories AC-3 encoder.

10.7.2 Dolby AC-3/Dolby Digital/Dolby SR · D

The 5.1-channel "surround" format that had become the *de facto* standard in most movie houses during the 1980s was becoming ubiquitous in home theaters of the 1990s that were equipped with matrixed multichannel sound (e.g., Dolby ProLogic™). As a result of this trend, it was clear that emerging applications for perceptual coding would eventually minimally require stereophonic or even multichannel surround-sound capabilities to gain consumer acceptance. Although single-channel algorithms such as the AC-2 can run on parallel independent channels, significantly better performance can be realized by treating multiple channels together in order to exploit interchannel redundancies and irrelevancies. The Dolby Laboratories AC-3 algorithm [Davis93] [Todd94] [Davi98], also known as "Dolby Digital" or "SR · D," was developed specifically for multichannel coding by refining all of the fundamental AC-2 blocks, including the filter bank, the spectral envelope encoding, the perceptual model, and the bit allocation. The coder carries 5.1 channels of audio (left, center, right, left surround, right surround, and a subwoofer), but at the same time it incorporates a flexible downmix strategy at the decoder to maintain compatibility with conventional monaural and stereophonic sound reproduction systems. The ".1" channel is usually reserved for low-frequency effects, and is lowpass bandlimited below 120 Hz. The main features of the AC-3 algorithm are as follows:

- Sample rates: 32, 44.1, and 48 kHz
- Bit rates: 32–640 kb/s, variable
- High-quality output at 64 kb/s per channel
- Delay roughly 100 ms
- MDCT filter bank (oddly stacked TDAC [Prin87]), KBD prototype window
- MDCT coefficients quantized and encoded in terms of exponents, mantissas
- Spectral envelope represented by exponents
- Signal-adaptive exponent strategy with time-varying time-frequency resolution
- Hybrid forward-backward adaptive perceptual model
- Parametric bit allocation
- Uniform quantization of mantissas according to signal-adaptive bit allocation
- Perceptual model improvements possible at the encoder without changing decoder
- Multiple channels processed as an ensemble
- Frequency-selective intensity coding, as well as LR, MS
- Robust decoder downmix functionality from 5.1 to fewer channels
- Integral dynamic range control system
- Board-level real-time encoders available
- Chip-level real-time decoders available.

The AC-3 works in the following way. A signal-adaptive MDCT filter bank with a customized KBD window (Section 6.7) maps the input to the frequency domain. Long windows are applied during steady-state segments, and a pair of short windows is used for transient segments. The MDCT coefficients are quantized and encoded by an exponent/mantissa scheme similar to AC-2. Bit allocation for the mantissas is performed according to a perceptual model that estimates the masked threshold from the quantized spectral envelope. Like AC-2, an identical perceptual model resides at both the encoder and decoder to allow for backward adaptive bit allocation on the basis of the spectral envelope, thus reducing the burden of side information on the bitstream. Unlike AC-2, however, the perceptual model is also forward adaptive in the sense that it is parametric. Model parameters can be changed at the encoder and the new parameters transmitted to the decoder in order to affect modified masked threshold calculations. Particularly at lower bit rates, the perceptual bit allocation may yield insufficient bits to satisfy both the masked threshold and the rate constraint. When this happens, mid-side and intensity coding ("channel coupling" above 2 kHz) reduce the demand for bits by exploiting, respectively, interchannel redundancies and irrelevancies. Ultimately, exponents, mantissas, coupling data, and exponent strategy data are combined and transmitted to the receiver.

The remainder of this section provides details on the major functional blocks of the AC-3 algorithm, including the filter bank, exponent strategy, perceptual model, bit allocation, mantissa quantization, intensity coding, system-level functions, complexity, and applications and standardization activities.

10.7.2.1 Filter Bank Although the high-level AC-3 structure (Figure 10.27) resembles that of AC-2, there are significant differences between the two algorithms. Like AC-2, the AC-3 algorithm first maps input samples to the frequency domain using a PR cosine-modulated filter bank with a novel KBD window (Section 6.7 parameters given in [Fiel96]). Unlike AC-2, however, AC-3 is based on the oddly stacked MDCT. The AC-3 also handles window switching differently than AC-2A. Long, 512-sample (93.75 Hz resolution @ 48 kHz) windows are used to achieve reasonable coding gain during stationary segments. During transients, however, a pair of 256-sample windows replaces the long window to minimize pre-echoes. Also in contrast to the MPEG and AC-2 algorithms, the AC-3 MDCT filter bank retains PR properties during window switching without resorting to bridge windows by introducing a suitable phase shift into the MDCT basis vectors (Chapter 6, Eq. (6.38a) and (6.38b); see also [Shli97]) for one of the two short transforms. Whenever a scheme similar to the one used in AC-2A detects transients, short filter-bank windows may activate independently on any one or more of the 5.1 channels.

10.7.2.2 Exponent Strategy The AC-3 algorithm uses a refined version of the AC-2 exponent/mantissa MDCT coefficient representation, resulting in a significantly improved coding gain. In AC-3, the MDCT coefficients corresponding to 1536 input samples (six transform blocks) are combined into a single frame.

Then, a frame-processing routine optimizes the exponent representation to exploit temporal redundancy, while at the same time representing the stair-step spectral envelope with adequate frequency resolution. In particular, spectral envelopes are formed from partitions of either one, two, or four consecutive MDCT coefficients on each of the six MDCT blocks in the frame. To exploit time-redundancy, the six envelopes can be represented individually, or any or all of the six can be combined into temporal partitions. As in AC-2, the exponents correspond to the peak values of each time-frequency partition, and each exponent is represented with 6 dB of resolution by determining the number of left shifts until overflow. The overall exponent strategy is selected by evaluating spectral stability. Many strategies are possible. For example, all transform coefficients could be transmitted for stable spectra, but time updates might be restricted to 32-ms intervals, i.e., an envelope of single-coefficient partitions might be repeated five times to exploit temporal redundancy. On the other hand, partitions of two or four components might be encoded for transient signals, but the time-partition might be smaller, e.g., updates could occur for every 5.3-ms MDCT block. Regardless of the particular strategy in use for a given frame, exponents are differentially encoded across frequency. Differential coding of exponents exploits knowledge of the filter-bank transition band characteristics, thus avoiding slope overload with only a five-level quantization strategy. The AC-3 exponent strategy exploits in a signal-dependent fashion the time- and frequency-domain redundancies that exist on a frame of MDCT coefficients.

10.7.2.3 Perceptual Model A novel parametric forward-backward adaptive perceptual model estimates the masked threshold on each frame. The forward-adaptive component exists only at the encoder. Given a rate constraint, this block interacts with an iterative rate control loop to determine the best set of perceptual model parameters. These parameters are passed to the backward-adaptive component, which estimates the masked threshold by applying the parameters from the forward-adaptive component to a calculation involving the quantized spectral envelope. Identical backward-adaptive model components are embedded in both the encoder and decoder. Thus, model parameters are fixed at the encoder after several threshold calculations in an iterative rate control process, and then transmitted to the decoder. The decoder only needs to perform one threshold calculation given the parameter values established at the encoder.

The backward-adaptive model component works as follows. First, the quantized spectral envelope exponents are clustered into 50, 0.5-Bark-width subbands. Then, a spreading function is applied (Figure 10.28a) that accounts only for the upward spread of masking. To compensate for filter-bank leakage at low frequencies, spreading is disabled below 200 Hz. Also, spreading is not enabled between 200 and 700 Hz for frequencies below the occurrence of the first significant masker. The absolute threshold of hearing is accounted for after the spreading function has been applied. Unlike other algorithms, AC-3 neglects the downward spread of masking, assumes that masking power is nonadditive, and makes no explicit assumptions about the relationship between tonality and the

skirt slopes on the spreading function. Instead, these characteristics are captured in a set of parameters that comprise the forward-adaptive model component. Masking threshold calculations at the decoder are controlled by a set of parameters transmitted by the encoder, creating flexibility for model improvements at the encoder such that improved estimates of the masked threshold can be realized without modifying the embedded model at the decoder.

For example, a parametric (upwards only) spreading function is defined (Figure 10.28a) in terms of two slopes, S_i, and two level offsets, L_i, for $i \in [1, 2]$. While the parameters S_1 and L_1 can be uniquely specified for each channel, the parameters S_2 and L_2 are applied to all channels. The parametric spreading function is advantageous in that it allows the perceptual model at the encoder

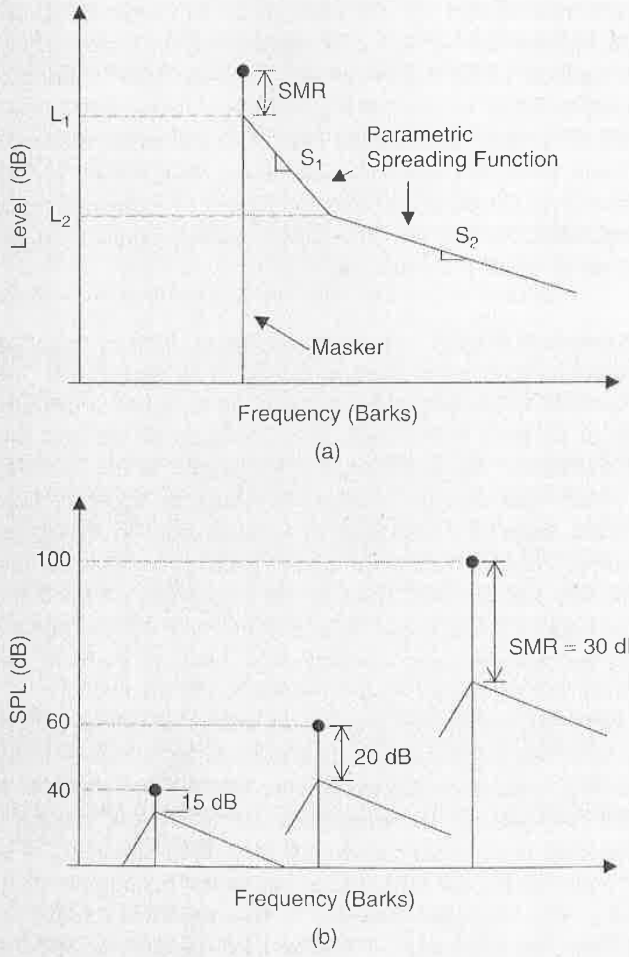


Figure 10.28. Dolby AC-3 parametric perceptual model: (a) prototype spreading function, (b) masker SMR level-dependence.

to account for tonality or dynamic masking patterns without the need to alter the decoder model. A range of values is available for each parameter. With units of dB per 1/2 Bark, the slopes are defined to be within the ranges $-2.95 \leqslant S_1 \leqslant -5.77$, and $-0.7 \leqslant S_2 \leqslant -0.98$. With units of dB SPL, the levels are defined to be within the ranges $-6 \leqslant L_1 \leqslant -48$ and $-49 \leqslant L_2 \leqslant -63$. Ultimately, there are 512 unique spreading function shapes to choose from. The acoustic-level dependence of masking thresholds is also modeled in AC-3. It is in general true that the signal-to-mask ratio (SMR) increases with increasing stimulus level (Figure 10.28b), i.e., the threshold moves closer to the stimulus as the stimulus intensity decreases. In the AC-3 parametric perceptual model, this phenomenon is captured by adding a positive bias to the masked thresholds when the spectral envelope is below a threshold level. Acoustic level threshold biasing is applied on a band-by-band basis. The decision threshold for the biasing is one of the forward adaptive parameters transmitted by the encoder. This function can also be disabled altogether. The parametric perceptual model also provides a convenient upgrade path in the form of a bit allocation delta parameter.

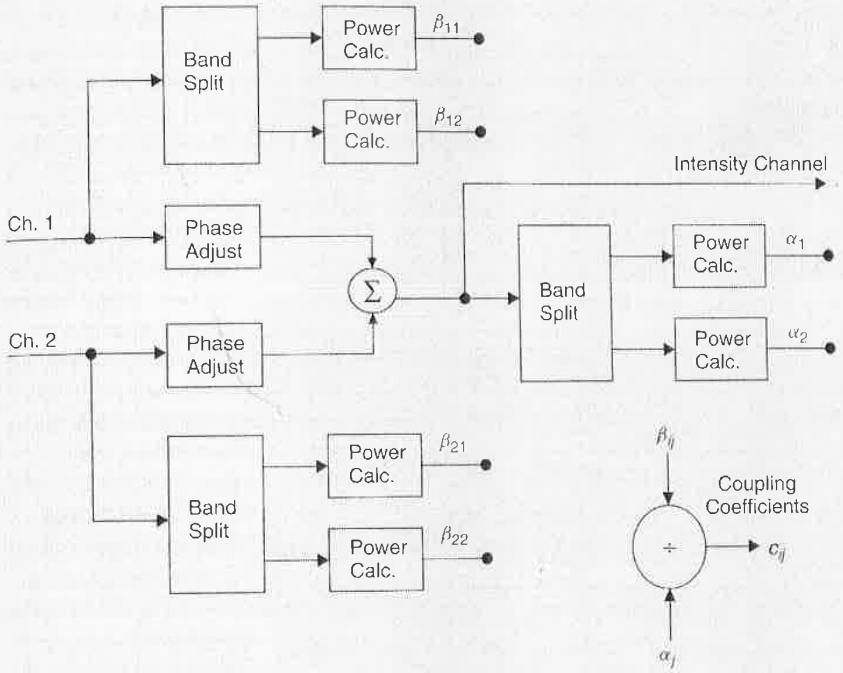
It was envisioned that future, more sophisticated AC-3 encoders might run in parallel two perceptual models, with one being the original reference model, and the other being an enhanced model with more accurate estimates of masked threshold. The delta parameter allows the encoder to transmit a stair-step function for which each tread specifies a masking level adjustment for an integral number of 1/2-Bark bands. Thus, the masking model can be incrementally improved without alterations to the existing decoders. Other details on the hybrid backward-forwards AC-3 perceptual model can be found in [Davi94].

10.7.2.4 Bit Allocation and Mantissa Quantization A bit allocation is determined at the encoder for each frame of mantissas by an iterative procedure that adjusts the mantissa quantizers, the multichannel coding strategies (below), and the forward-adaptive model parameters to satisfy simultaneously the specified rate constraint and the masked threshold. Within the rate-control loop, threshold partitions are formed on the basis of a bit allocation frequency resolution parameter, with coefficient partitions ranging in width between 94 and 375 Hz. In a manner similar to MPEG-1, quantizers are selected for the set of mantissas in each partition based on an SMR calculation. Sufficient bits are allocated to ensure that the SNR for the quantized mantissas is greater than or equal to the SMR. The quantization noise is thus rendered inaudible, below masked threshold. Uniform quantizers are selected from a set of 15 having 0, 3, 5, 7, 11, and 15 levels symmetric about 0, and conventional 2's-complement quantizers having 32, 64, 128, 256, 512, 1024, 2048, 4096, 16384, or 65536 levels. Certain quantizer codewords are group-encoded to make more efficient usage of available bits. Dithering can be enabled optionally on individual channels for 0-bit mantissas. If the bit supply is insufficient to satisfy the masked threshold, then SNRs can be reduced in selected threshold partitions until the rate is satisfied, or intensity coding and MS transformations are used in a frequency-selective fashion to reduce the bit demand. Two variable-rate methods are also available to

satisfy peak-rate demands. Within a frame of six MDCT coefficient blocks, bits can be distributed unevenly, such that the instantaneous bit rate is variable but the average rate is constant. In addition, bit rates are adjustable, and a unique rate can be specified for each frame of six MDCT blocks. Unlike some of the other standardized algorithms, the AC-3 does not include an explicit lossless coding stage for final redundancy reduction after quantization and encoding.

10.7.2.5 Multichannel Coding When bit demand imposed by multiple independent channels exceeds the bit budget, the AC-3 ensemble processing of 5.1 channels exploits interchannel redundancies and irrelevancies, respectively, by making frequency-selective use of mid-side (MS) and intensity coding techniques. Although the MS and intensity functions can be simultaneously active on a given channel, they are restricted to nonoverlapping subbands. The MS scheme is carefully controlled [Davi98] to maintain compatibility between AC-3 and matrixed surround systems such as Dolby ProLogic. Intensity coding, also known as "channel coupling," is a multichannel irrelevancy reduction coding technique that exploits properties of spatial hearing. There is considerable experimental evidence [Blau74] suggesting that the interaural time difference of a signal's fine structure has negligible influence on sound localization above a certain frequency. Instead, the ear evaluates primarily energy envelopes. Thus, the idea behind intensity coding is to transmit only one envelope in place of two or more sufficiently correlated spectra from independent channels, together with some side information. The side information consists of a set of coefficients that is used to recover individual spectra from the intensity channel.

A simplified version of the AC-3 intensity coding scheme is illustrated in Figure 10.29. At the encoder (Figure 10.29a), two or more input spectra are added together to form a single intensity channel. Prior to the addition, an optional adjustment is applied to prevent phase cancellation. Then, groups of adjacent coefficients are partitioned into between 1 and 18 separate intensity subbands on both the individual and the intensity channels. A set of coupling coefficients is computed, c_{ij}, that expresses the fraction of energy contributed by the i-th individual channel to the j-th band of the intensity envelope, i.e., $c_{ij} = \beta_{ij}/\alpha_j$, where β_{ij} is the power contained in the j-th band of the i-th channel, and α_j is the power contained in the j-th band of the intensity channel. Finally, the intensity spectrum is quantized, encoded, and transmitted to the decoder. The coupling coefficients, c_{ij}, are transmitted as side information. Once the intensity channel has been recovered at the decoder (Figure 10.29b), the intensity subbands are scaled by the coupling coefficients, c_{ij}, in order to recover an appropriate fraction of intensity energy in the j-th band of the i-th channel. The intensity-coded coefficients are then combined with any remaining uncoupled transform coefficients and passed through the synthesis filter bank to reconstruct the individual channel. The AC-3 coupling coefficients have a dynamic range that spans -132 to $+18$ dB, with quantization step sizes between 0.28 and 0.53 dB. Intensity coding is applied in a frequency-selective manner, parameterized by a start frequency of 3.42 kHz or higher, and a bandwidth expressed in multiples of 1.2 kHz for



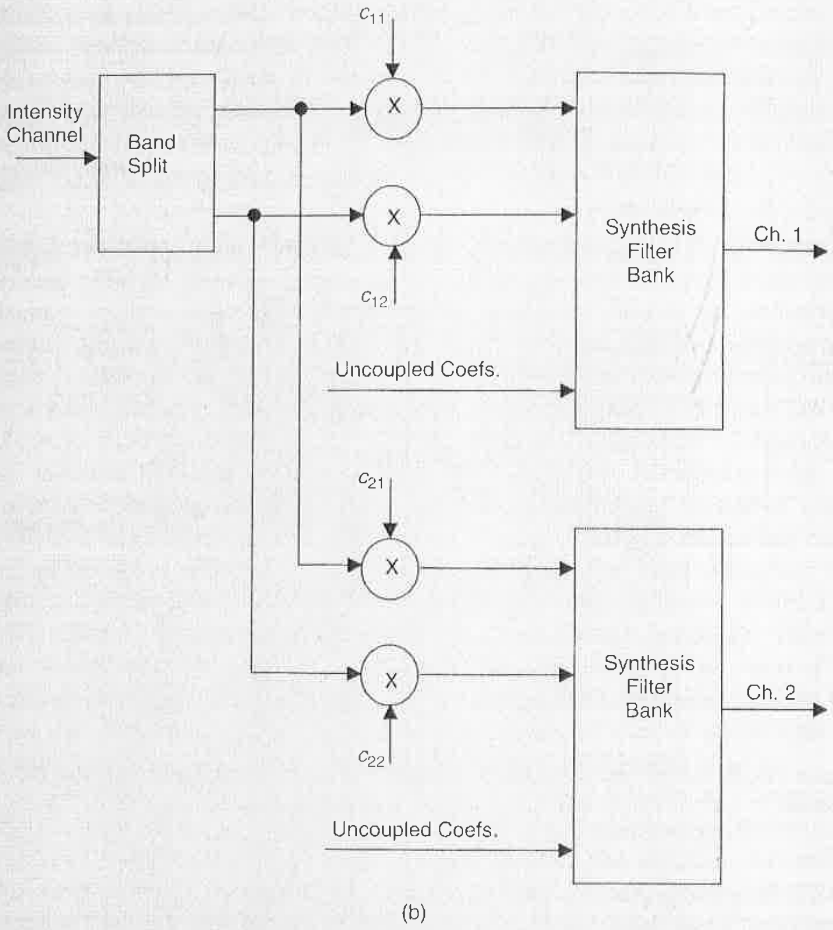
(a)

Figure 10.29. Dolby AC-3 intensity coding ("channel coupling"): (a) encoder, (b) decoder.

a 48-kHz system [Davi98]. Note that unlike the simplified system shown in the figure, the actual AC-3 intensity coding scheme may couple the spectra from as many as five channels.

10.7.2.6 System-Level Functions At the system level, AC-3 provides mechanisms for channel down-mixing and dynamic range control. Down-mix capability is essential for the 5.1-channel system since the majority of potential playback systems are still monaural or, at best, stereophonic. Down-mixing is performed at the decoder in the frequency domain rather than the time-domain to reduce complexity. This is possible because of the filter-bank linearity. The bit stream carries some down-mix information since different listening situations call for different down-mix weighting. Dialog-level normalization is also available at the decoder. Finally, the bit stream has available facilities to handle other control and ancillary user information such as copyright, language, production, and time-code data [Davis94].

10.7.2.7 Complexity Assuming the standard HDTV configuration of 384 kb/s with a 48 kHz sample rate and implementation using the Zoran ZR38001 general-purpose DSP instruction set, the AC-3 decoder memory requirements and complexity



(b)

Figure 10.29. (*continued*).

are as follows: 6.6 kbytes RAM, 5.4 kbytes ROM, 27.3 MIPS for 5.1 channels, and 3.1 kbytes RAM, 5.4 kbytes ROM, and 26.5 MIPS for 2 channels [Vern95]. Note that complexity estimates are processor-dependent. For example, on a Motorola DSP56002, 45 MIPS are required for a 5.1-channel decoder. Encoder complexity varies between two and five times decoder complexity depending on the encoder sophistication [Vern95]. Numerous real-time encoder and decoder implementations have been reported. Early on, for example, a single-chip decoder was implemented on a Zoran DSP [Vern93]. More recently, a DP561 AC-3 encoder (5.1 channels, 44.1- or 48-kHz sample rate) for DVD mastering was implemented in real-time on a PC host with a plug-in DSP subsystem. The computational requirements were handled by an Ariel PC-Hydra DSP array of eight Texas Instruments TMS 320C44 floating-point DSP devices clocked at 50 MHz [Terr96]. Current information on real-time AC-3 implementations is also available online from Dolby Laboratories.

10.7.2.8 Applications and Standardization The first popular AC-3 application was in the cinema. The "Dolby Digital" or "SR D" AC-3 information is interleaved between sprocket holes on one side of the 35-mm film. The AC-3 was first deployed in only three theaters for the film *Star Trek VI* in 1991, after which the official rollout of Dolby SR D occurred in 1992 with *Batman Returns*. By 1997 April, over 900 film soundtracks had been AC-3 encoded. Nowadays, the AC-3 algorithm is finding use in digital versatile disc (DVD), cable television (CATV), and direct broadcast satellite (DBS). Many hi-fidelity amplifiers and receiver units now contain embedded AC-3 decoders and accept an AC-3 digital rather than an analog feed from external sources such as DVD.

In addition, the DP504/524 version of the DolbyFAX system (Section 10.7.1) has added AC-3 stereo and MPEG-1 layer II to the original AC-2-based system. Film, television, and music studios use DolbyFAX over ISDN links for automatic dialog replacement, music collaboration, sound effects delivery, and remote videotape audio playback. As far as standardization is concerned, the United States Advanced Television Systems Committee (ATSC) has adopted the AC-3 algorithm as the A/52 audio compression standard [USAT95b] and as the audio component of the A/52 Digital Television (DTV) Standard [USAT95a]. The United States Federal Communications Commission (US FCC) in 1996 December adopted the ATSC standard for DTV, including the AC-3 audio component. On the international standardization front, the Digital Audio-Visual Council (DAVIC) selected AC-3 and MPEG-1 layer II for the audio component of the DAVIC 1.2 specification [DAVC96].

10.7.2.9 Recent Developments –The Dolby Digital Plus A Dolby digital plus system or the enhanced AC-3 (E-AC-3) [Fiel04] was recently introduced to extend the capabilities of the Dolby AC-3 algorithm. While remaining backward compatible with the Dolby AC-3 standard, the Dolby digital plus provides several enhancements. Some of the extensions include flexibility to encode up to 13.1 channels, extended data rates up to 6.144 Mb/s. The AC-3 filterbank is supplemented with a second stage DCT to exploit the stationary characteristics in the audio. Other coding tools include spectral extension, enhanced channel coupling, and transient pre-noise processing. The E-AC-3 is used in cable and satellite television set-top boxes and broadcast distribution transcoding devices. For a detailed description on the Dolby digital plus refer to [Fiel04].

10.8 AUDIO PROCESSING TECHNOLOGY APT-x100

Without exception, all of the commercial and international audio coding standards described thus far couple explicit models of auditory perception with classical quantization techniques in an attempt to distribute quantization noise over the time-frequency plane such that it is imperceptible to the human listener. In addition to irrelevancy reduction, most of these algorithms simultaneously seek to reduce statistical redundancies. For the sake of comparison and perhaps to better assess the impact of perceptual models on realizable coding gain, it is instructive

to next consider a commercially available audio coding algorithm that relies only upon redundancy removal without any explicit regard for auditory perception.

We turn to the Audio Processing Technology APT-x100 algorithm, which has been reported to achieve nearly transparent coding of CD-quality 44.1 kHz 16-bit PCM input at a compression ratio of 4:1, or 176.4 kb/s per monaural channel [Wyli96b]. Like the ITU-T G.722 wideband speech codec [G722], the APT-x100 encoder (Figure 10.30) relies upon subband signal decomposition followed by independent ADPCM quantization of the decimated subband output sequences. Codewords from four uniform bandwidth subbands are multiplexed onto the channel and sent to the decoder where the ADPCM and filter-bank operations are inverted to generate an output. As shown in the figure, a tree-structured QMF filter bank splits the input signal into four subbands. The first and second filter stages have 64 and 32 taps, respectively. Backward adaptive prediction is applied to the four subband output sequences. The resulting prediction residual is quantized with a backward-adaptive Laplacian quantizer. Backward adaptation in the prediction and quantization steps eliminates side information but increases sensitivity to fast transients. On the other hand, both prediction and adaptive quantization were found to significantly improve coding gain for a wide range of test signals [Wyli96b]. Adaptive quantization attempts to track signal dynamics and tends to produce constant SNR in each subband during stationary segments.

Unlike the other algorithms reviewed in this document, APT-x100 contains no perceptual model or rate control loop. The ADPCM output codewords are of fixed resolution (1 bit per sample), and therefore with four subbands the output bit rate is reduced 4:1. A comparison between APT-x100 quantization noise and

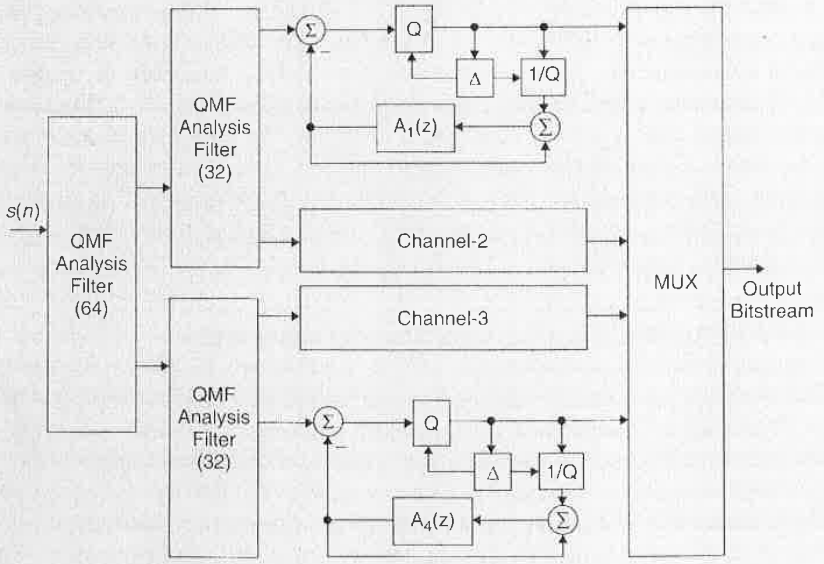


Figure 10.30. Audio Processing Technology APT-x100 encoder.

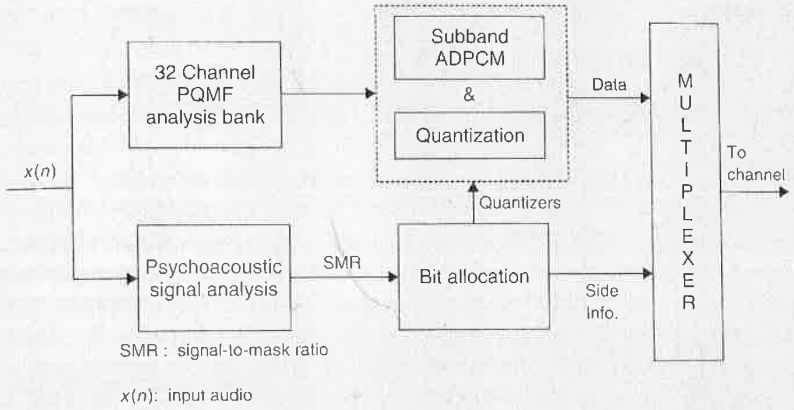


Figure 10.31. DTS-coherent acoustics (DTS-CA) encoding scheme.

noise masking thresholds computed as in [John88a] for a variety of test signals from the SQAM test CD [SQAM88] revealed two trends in the APT-x100 noise floor. First, as expected, it is flat rather than shaped. Second, the noise is below the masking threshold in most critical bands for most stationary test signals, but tends to exceed the threshold in some critical bands for transient signals. In [Wyli96b], however, the fast step-size adaptation in APT-x100 is reported to exploit temporal masking effects and mitigate the audibility of unmasked quantization noise. While the lack of a perceptual model results in an inefficient flat noise floor, it also affords some advantages including reduced complexity, reduced frequency resolution requirements, and low delay of only 122 samples or 2.77 ms at 44.1 kHz.

Several other relevant facts on APT-x100 quality and robustness were also reported in [Wyli96b]. Objective output quality was evaluated in terms of average subband SNRs, which were 30, 15, 10, and 7 dB, respectively, for the lowest to highest subbands, and the authors stated that the algorithm outperformed NICAM [NICAM] in an informal subjective comparison [Wyli96b]. APT-x100 was robust to both random bit errors and tandem encoding. Errors were inaudible for a bit error rate (BER) of 10^{-4}, and speech remained intelligible for a BER of 10^{-1}. In one test, 10 stages of synchronous tandeming reduced output SNR from 45 dB to 37 dB. An auxiliary channel that accommodates up to 1/4 kb/s of the sample rate in buried data (e.g., 24 kb/s for 48-kHz stereo samples) by bit stealing from one of the subbands had a negligible effect on output quality. Finally, real-time APT-x100 encoder and decoder modules were implemented on a single AT&T DSP16A masked ROM DSP. As far as applications are concerned, APT-x100 has been deployed in digital studio-transmitter links, audio storage products, and cinematic surround sound applications. A cursory performance comparison of the nonperceptual algorithms versus the perceptually based algorithms (e.g., NICAM or APT-x100 vs. MPEG or PAC, etc.) confirms that some awareness of peripheral auditory processing is necessary to achieve high-quality coding of CD-quality audio for compression ratios in excess of 4:1.

10.9 DTS – COHERENT ACOUSTICS

The performance comparison of the nonperceptual algorithms versus the perceptually based algorithms (e.g., APT-x100 vs. MPEG or PAC, etc.) given in the earlier section, highlights that some awareness of *peripheral auditory processing* is necessary to achieve high-quality encoding of digital audio for compression ratios in excess of 4:1. To this end, DTS employs an audio compression algorithm based on the principles of "coherent acoustics encoding" [Smyt96] [Smyt99] [DTS]. In coherent acoustics, both ADPCM-subband filtering and psychoacoustic analysis are employed to compress the audio data. The main emphasis in DTS is to improve the precision (and, hence, the quality) of the digital audio. The DTS encoding algorithm provides a resolution of up to 24 bits per sample and at the same time can deliver compression rates in the range of 3 to 40. Moreover, DTS can deliver up to eight discrete channels of multiplexed audio at sampling frequencies of 8–192 kHz and at bit rates of 8–512 kb/s per channel. Table 10.9 summarizes the various bit rates, sampling frequencies, and the bit resolutions employed in the four configurations supported by the DTS-coherent acoustics.

10.9.1 Framing and Subband Analysis

The DTS-CA encoding algorithm (Figure 10.31) operates on 24-bit linear PCM signals. The audio signals are typically analyzed in blocks (frames) of 1024 samples, although frame sizes of 256, 512, 2048, and 4096 samples are also supported depending on the bit rates and sampling frequencies used (Table 10.10). For example, if operating at bit rates of 1024–2048 kb/s and sampling frequencies of 32 or 44.1 or 48 kHz; then the maximum number of samples allowed per frame is 1024. Next, the segmented audio frames are decomposed into 32 critically subsampled subbands using a polyphase realization of a pseudo QMF (PQMF) bank (Chapter 6). Two different PQMF filter-bank structures, namely, perfect reconstructing (PR) and nonperfect reconstructing (NPR) are provided in DTS-coherent acoustics. In the example that we considered above, a frame size of 1024 samples results in 32 PCM samples per subband (i.e., 1024/32). The channels are equally spaced such that a 32 kHz input signal is split into 500 Hz subbands (i.e., 16 kHz/32), with the subbands being decimated at the ratio 32:1.

Table 10.9. A list of encoding parameters used in DTS-coherent acoustics after [Smyt99].

Bit rates (kb/s/channel)	Sampling rates (kHz)	Bit resolution per sample
8–32	≤24	16
32–96	≤48	20
96–256	≤96	24
256–512	≤192	24

Table 10.10. Maximum frame sizes allowed in DTS-CA (after [Smyt99]).

	Sampling frequency-set, $f_s = \{8/11.05/12\}$ (kHz)				
Bit rates (kb/s)	f_s	$2f_s$	$4f_s$	$8f_s$	$16f_s$
0–512	Max. 1024	Max. 2048	Max. 4096	N/A	N/A
512–1024	N/A	Max. 1024	Max. 2048	N/A	N/A
1024–2048	N/A	N/A	Max. 1024	Max. 2048	N/A
2048–4096	N/A	N/A	N/A	Max. 1024	Max. 2048

10.9.2 Psychoacoustic Analysis

While the subband filtering stage minimizes the statistical dependencies associated with the input PCM signal, the psychoacoustic analysis stage eliminates the perceptually irrelevant information. Since we have already established the necessary background on psychoacoustic analysis in Chapters 5, we will not elaborate on these steps. However, we describe next the advantages of combining the differential subband coding techniques (e.g., ADPCM) with the psychoacoustic analysis.

10.9.3 ADPCM – Differential Subband Coding

A block diagram depicting the steps involved in the differential subband coding in DTS-CA is shown in Figure 10.32. A fourth-order forward linear prediction is performed on each subband containing 32 PCM samples. From the above example, we have 32 subbands and 32 PCM samples per subband. Recall that in LP we predict the current time-domain audio sample based on a linearly weighted combination of previous samples. From the LPC analysis corresponding to the i-th subband, we obtain predictor coefficients, $a_{i,k}$ for $k = 0, 1, \ldots, 4$ and the residual error, $e_i(n)$ for $n = 0, 1, 2, \ldots, 31$ samples. The predictor coefficients are usually vector quantized in the line spectral frequency (LSF) domain.

Two stages of ADPCM modules are provided in the DTS-CA algorithm, i.e., the *ADPCM estimation stage* and the *real ADPCM stage*. ADPCM utilizes the redundancy in the subband PCM audio by exploiting the correlation between adjacent samples. First, the "estimation ADPCM" module is used to determine the degree of prediction achieved by the fourth-order linear prediction filter (Figure 10.32). Depending upon the statistical features of audio, a decision to enable or disable the second "real ADPCM" stage is made.

A predictor mode flag, "PMODE" = 1 or 0, is set to indicate if the "real ADPCM" module is active or not, respectively.

$$s_{i,pred}(n) = \sum_{k=0}^{4} a_{i,k} s_i(n-k), \text{ for } n = 0, 1, \ldots, 31 \tag{10.11}$$

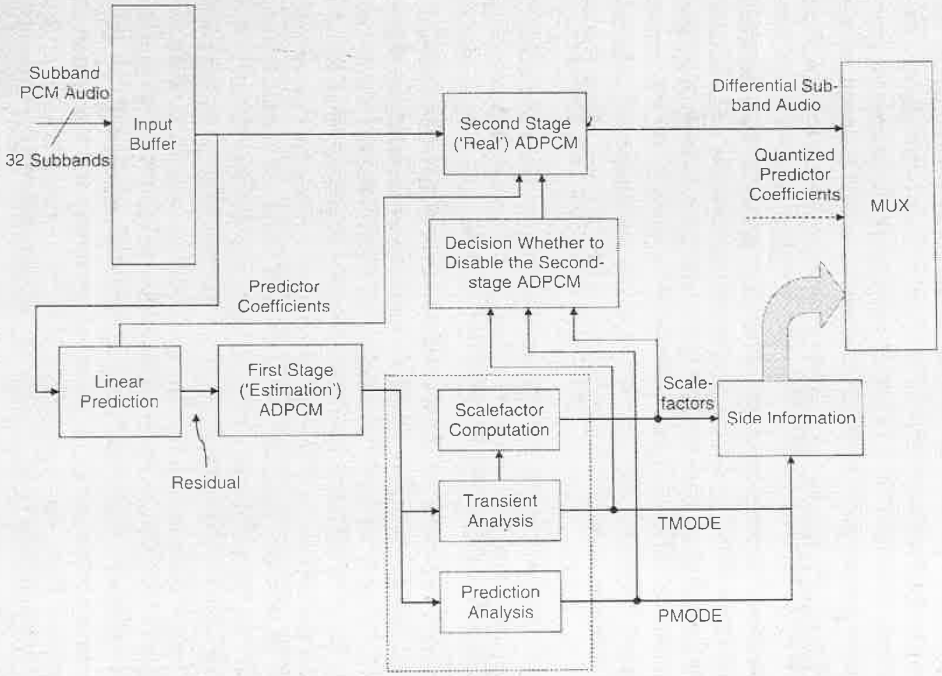


Figure 10.32. Differential subband (ADPCM) coding in the DTS-CA encoder.

$$e_i(n) = s_i(n) - s_{i,pred}(n)$$
$$= s_i(n) - \sum_{k=0}^{4} a_{i,k} s_i(n-k) \tag{10.12}$$

While the "prediction analysis" block computes the PMODE flag based on the prediction gain, the "transient analysis" module monitors the transient behavior of the error residual. In particular, when a signal with a sharp attack (i.e., rapid transitions) begins near the end of a transform block and immediately following a region of low energy, pre-echoes occur. Several pre-echo control strategies have been developed (Chapter 6, Section 6.10). These include window switching, gain modification, switched filter banks, including the bit reservoir, and temporal noise shaping. In DTS-CA, the pre-echo artifacts are controlled by dividing the subband analysis buffer into four sub-buffers. A transient mode, "TMODE" = 0, 1, 2, or 3, is set to denote the beginning of a transient signal in sub-buffers 1, 2, 3, or 4, respectively. In addition, two scale factors are computed for each subband (i.e., before and after the transition) based on the peak magnitude of the residual error, $e_i(n)$. A 64-level nonuniform quantizer is usually employed to encode the scale factors in DTS-CA.

Note that the PMODE flag is a "Boolean" and the TMODE flag has four values. Therefore, a total of 15 bits (i.e., 12 bits for two scale factors, 1 bit for the "PMODE" flag, and 2 bits for the "TMODE" flag) are sufficient to encode the entire *side information* in the DTS-CA algorithm. Next, based on the predictor mode flag (1 or 0), the second-stage ADPCM is used to encode the differential subband PCM audio as shown in Figure 10.32. The optimum number of bits (in the sense of minimizing the quantization noise) required to encode the differential audio in each subband is estimated using a bit allocation procedure.

10.9.4 Bit Allocation, Quantization, and Multiplexing

Bit allocation is determined at the encoder for each frame (32 subbands) by an iterative procedure that adjusts the scale-factor quantizers, the fourth-order linear predictive model parameters, and the quantization levels of the differential subband audio. This is done in order to satisfy simultaneously the specified rate constraint and the masked threshold. In a manner similar to MPEG-1, quantizers are selected in each subband based on an SMR calculation. A sufficient number of bits is allocated to ensure that the SNR for the quantized error is greater than or equal to the SMR. The quantization noise is thus rendered inaudible, i.e., below the masked threshold. Recall that the main emphasis in DTS-CA is to improve precision and hence the quality of the digital audio, while giving relatively less importance to minimizing the data rate. Therefore, the DTS-CA bit reservoir will almost always meet the bit demand imposed by the psychoacoustic model. Similar to some of the other standardized algorithms (e.g., MPEG codecs, lossless audio coders), the DTS-CA includes an explicit lossless coding stage for final redundancy reduction after quantization and encoding. A data multiplexer merely packs the differential subband data, the side information, the synchronization details,

and the header syntax into a serial bitstream. Details on the structure of the "output data frame" employed in the DTS-CA algorithm are given in [Smyt99] [DTS].

As an extension to the current coherent acoustics algorithm, Fejzo *et al.* proposed a new enhancement that delivers 96 kHz, 24-bit resolution audio quality [Fejz00]. The proposed enhancement makes use of both "core" and "extension" data to reproduce 96-kHz audio bitstreams. Details on the real-time implementation of the 5.1-channel decoder on a 32-bit floating-point processor are also presented in [Fejz00]. Although much work has been done in the area of encoder/decoder architectures for the DTS-CA codecs, relatively little has been published [Mesa99].

10.9.5 DTS-CA Versus Dolby Digital

The DTS-Coherent Acoustics and the Dolby AC-3 algorithms were the two competing standards during the mid-1990s. While the former employs an adaptive differential linear prediction (ADPCM–subband coding) in conjunction with a perceptual model, the latter employs a unique exponent/mantissa MDCT coefficient encoding technique in conjunction with a parametric forward-backward adaptive perceptual model.

PROBLEMS

10.1. List some of the primary differences between the DTS, the Dolby digital, and the Sony ATRAC encoding schemes.

10.2. Using a block diagram, describe how the ISO/IEC MPEG-1 layer I codec is different from the ISO/IEC MPEG-1 layer III algorithm.

10.3. What are the enhancements integrated into MPEG-2 AAC relative to the MP3 algorithm. State key differences in the algorithms.

10.4. List some of the distinguishing features of the MP4 audio format over the MP3 format. Give bitrates and cite references.

10.5. What is the main idea behind the scalable audio coding? Explain using a block diagram. Give examples.

10.6. What is structured audio coding and parametric audio coding?

10.7. How is ISO/IEC MPEG-7 standard different from the other MPEG standards.

COMPUTER EXERCISE

10.8. The audio files *Ch10aud2L.wav*, *Ch10aud2R.wav*, *Ch10aud2C.wav*, *Ch10aud2Ls.wav*, and *Ch10aud2Rs.wav* correspond to left, right, center, left-surround, and right-surround, respectively, of a 3/2-channel configuration. Using the matrixing technique obtain a stereo output.