On Scalable to Lossless Audio Coding

Li Te

A thesis submitted to the

School of Electrical & Electronic Engineering,

Nanyang Technological University

in fulfilment of the requirement for the degree of

Doctor of Philosophy

2008

Acknowledgements

I would like to express my profound appreciation and sincere gratitude to my supervisors, Associate Professor Susanto Rahardja and Professor Koh Soo Ngee, for their invaluable guidance, advices and support throughout my research. The things I have learnt from them are extremely precious to me and I believe that they will benefit my whole life.

I am deeply grateful to my parents, Mr. Li Xuedong and Ms Zhang Yuwen, for their endless love, care and encouragements all along, though we are separated in two countries with thousands of miles apart.

Finally, my sincere thanks go to the staff members from the Signal Processing Department of Institute for Infocomm Research, A*STAR, for the inspirational discussions I had with them and the enthusiastic help so generously given to me throughout my research.

Contents

Summary

With advances in broadband network and storage technologies, more and more digital audio applications are ready to deliver high sampling rate and high-resolution lossless audio. On the other hand, there are also applications that require highly compressed audio such as those found in wireless communications. To deal with these various demands, a scalable audio coding technology that supports both lossy and lossless audio compression is thus desirable. MPEG-4 audio *scalable lossless* (SLS) coding was published as an international standard in June 2006. It allows the scaling up of a perceptually coded audio to a fully lossless audio with a wide range of intermediate bitrate representations. The main technologies adopted in SLS include the latest integer transform, namely *integer modified discrete cosine transform* (IntMDCT), and a new entropy coder that is based on *bit-plane Golomb code* (BPGC).

As a relatively new coding structure, SLS is far from perfect with room for improvements. There are several critical issues which directly affect the wide adoption and application of the SLS codec. This dissertation aims to provide answers to all these critical issues.

Firstly, the effect of the rounding errors introduced by IntMDCT under the perceptual (lossy) audio coding scenario is studied. Based on intensive test results,

it is concluded that MDCT and IntMDCT filterbanks are interchangeable in a lossy coding scenario. This finding justifies the use of the low-complexity SLS structure.

Secondly, perceptually enhanced prioritized bit-plane audio coding algorithms are proposed for the non-core and low-core-bitrate mode of SLS based on the energy distributions in different frequency regions. By using only a single bit in each frame to indicate one of the two coding models to be used, considerable perceptual quality enhancement is achieved for a wide range of bitrates.

Thirdly, efficient bit allocation schemes for stereo channels in both the SLS encoder and truncator are proposed. By allocating bits according to the energy level, significant improvement in quality can be achieved by the proposed algorithm for signal (such as speech) that is highly correlated for the left and right channels.

Lastly, a "smart" function is designed for SLS. With a low quality audio format and its original inputs, the proposed smart enhancing process enables a scalable encoder to automatically encode the minimum amount of enhancement for the low quality audio to attain a "transparent quality" that is the same as the CD quality. This function facilitates the application of SLS in multi-quality online music sales.

With these proposed solutions, the MPEG-4 SLS coder has been enhanced resulting in a much better perceptual quality and more robust features. The users can benefit from the convenience of the universality, as well as the excellent performance in terms of both the quality and compression, of this codec.

Finally, several interesting research topics for scalable lossless coding are also recommended for future research.

List of Figures

List of Tables

List of Abbreviations and Symbols

A

AAC	advanced audio coding
AE	allocation entropy

B

BPGC	bit-plane Golomb code
BSAC	bit-sliced arithmetic code

C

CBAC	context based arithmetic code
CfP	call for proposal

D

DCT-IV	type–IV discrete cosine transform

E

EZW	embedded zerotrees wavelet

F

FGS	fine granular scalability

I

IEC	International Electrotechnical Commission
I^2R	Institute for Infocomm Research

IntMDCT	integer modified discrete cosine transform
ISO	International Standardization Organization
J	
JTC	joint technical committee
K	
kbps	kilo bits per second
L	
LEMC	low energy mode code
LLE	lossless enhancement
LSB	least significant bit
M	
M/S	mid/side
MDL	multi-dimensional lifting
MLD	masking level difference
MPEG	Moving Picture Experts Group
MSB	most significant bit
N	
NMR	noise to mask ratio
O	
ODG	objective difference grade
OSF	oversampling factor
P	
PCM	pulse code modulation
pdf	probability density function
PE	perceptual entropy
PEAQ	perceptual evaluation of audio quality

PSD	power spectrum density
Q	
QoS	quality of service
R	
RM	reference model
S	
SAAC	scalable advanced audio coding
SAC	scalable audio coding
SC	sub committee
SDK	software development kit
SDL	single dimensional lifting
sfb	scalefactor band
sfbs	scalefactor bands
SLS	scalable lossless
SMR	signal to mask ratio
SNR	signal to noise ratio
SPIHT	set partitioning in hierarchical trees
T	
TWIN-VQ	transform-domain weighted interleave vector quantization
W	
WG	work group

Chapter 1

Introduction

1.1 Emergence of Scalable Lossloss Audio Coding

In the past several decades, digital audio has essentially replaced its analog counterpart due to its unprecedented advantages. However uncompressed digital audio, i.e., CD audio sampled at 44.1 kHz and ecoded at 16 bits per sample which results in 705.6 kbps per channel, is a heavy burden in several early digital sound applications. The expectation of CD quality at low rates motivated research towards the compression schemes that can satisfy simultaneously the conflicting demands of high compression and transparent quality [1–10]. Nowadays audio coders can deliver CD-quality stereo audio at bitrates from 64 – 192 kbps, or 21 – 7 times compression, and the quality degradation introduced are perceptually negligible in most cases. Most of these audio coders are designed using perceptual audio coding technique [11] which optimizes the perceptual quality of the compressed audio by exploiting the masking properties of the human auditory system. Perceptual coding technique is applied via a rate distortion loop to control the distortions

introduced during the coding process according to the *just noticeable distortion* (JND) threshold estimated from a psychoacoustic model.

With advances in broadband networking and storage technologies, the capacities of more and more digital audio applications are quickly approaching those for delivery of high sampling rate and high-resolution digital audio at lossless quality. Moreover, there are applications, such as audio archival, which require every bit of the original audio waveform to be preserved. On the other hand, there are also applications that require highly compressed audio such as wireless devices. To deal with the various demands, a scalable audio coding technology that supports both lossy and lossless audio compression *simultaneously* is thus desirable. Such an audio coder is expected to include the following features:

- **Lossy to lossless scalability**

  The lossy to lossless scalability in the bitstream level should be provided. The audio contents thus only need to be encoded once to adapt to various demands from different devices, as indicated in Figure 1.1.

- **Fine granular scalability (FGS)**

  FGS is desirable in an universal multimedia access [12] paradigm so that scalability with sufficiently fine step size is provided to cater for the large diversity of the network bandwidth, device capabilities and user preferences.

- **Perceptual scalability**

  This is a natural requirement for a scalable audio coder that better audio quality should always be furnished when more bits are consumed.

- **Coding efficiency and low complexity**

  It is understandable that scalability will usually entail certain overheads in

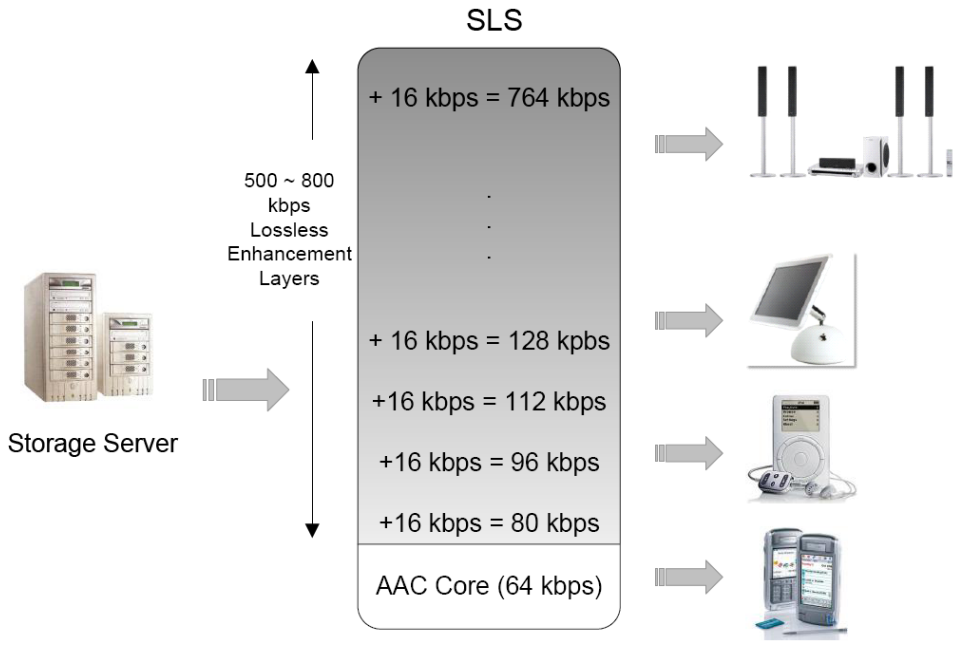


Figure 1.1: Feature of scalable lossless coding: encode once, adapt to all.

terms of coding efficiency and complexity. Such overheads should be minimized as much as possible.

Responding to this need, the international standard organization ISO/IEC JTC1 SC29 WG11, also known as *Moving Picture Experts Group* (MPEG) has conducted a market survey and issued a *call for proposal* (CfP) [13] in October 2002. This CfP solicited state-of-the-art lossless audio coding technologies that offer good performance, flexible quality scalability and backward compatibility to the MPEG *advanced audio coding* (AAC) technology [14]. The *reference model* (RM) for MPEG-4 audio *scalable lossless* (SLS) coding [15] was established in July 2003. It allows the scaling up of a perceptually coded representation such as AAC to a fully lossless representation with a wide range of intermediate bitrate representations. The main technologies adopted in SLS include the *integer modified discrete cosine transform* (IntMDCT) [16] and *bit-plane Golomb code* (BPGC) [17]. Subsequently, SLS has been further enhanced by several input technologies which include

a more efficient IntMDCT algorithm [18], noise shaping [19], *low energy mode code* (LEMC) and *context-based arithmetic code* (CBAC) [20]. MPEG-4 SLS was published by ISO in June 2006 as one of the standard MPEG-4 audio tools.

1.2 Motivation

Standardized MPEG-4 SLS achieves excellent compression performance which is comparable to that of a state-of-the-art non-scalable lossless audio coding system and fine granular scalability. However, as a relatively new codec structure, SLS (especially the reference encoder) is still far from being optimized and mature enough. Several vital issues still exist in its major components. These directly affect the wide adoption and application of the SLS codec. The most critical issues are summarized as follows.

1. IntMDCT, a relatively new integer transform, is adopted in the reference SLS structure to enable efficient lossless reconstruction. On the other hand, there is an MDCT filterbank which is inherent to the AAC core that is embedded in the SLS codec. The presence of two filterbanks has undoubtedly increased the complexity of the implementation and it is for this reason that the MDCT is disabled and the IntMDCT is then the only type of filterbank that is employed in SLS for both lossy and lossless operations. Because of the rounding operations in the IntMDCT, there is a concern as to whether the use of IntMDCT for perceptual audio coding will eventually degrade the fidelity of the audio codec.

2. The bit-plane coding in SLS is implemented in a sequential way from low frequency to high frequency, *most significant bit* (MSB) to *least significant bit* (LSB) without any reference to perceptual information. For some cases

where full scalability or near-full scalability is desired, the bitrate occupied by the perceptual core may be zero (non-core mode of SLS) or very low. In this scenario, the spectral shape of the residual signal for bit-plane coding roughly follows the shape of the original signal spectrum. This is far from the optimum for sequential bit-plane coding and will directly result in non-optimal perceptual quality of output audio at intermediate bitrates.

3. In SLS, *mid/side* (M/S) coding is applied. In RM, the bitrate for the two channels is equally allocated regardless of the possible level difference between the M and S channels. This is especially inefficient for the non-core mode of SLS, in which the perceptual core is absent.

4. One significant application of SLS is online music store. By using the SLS *software development kit* (SDK) released by the *Institute for Infocomm Research* (I^2R), various versions of compressed music at different bitrates with the corresponding prices are currently provided by Asia's largest mobile music store, www.soundbuzz.com. However, the quality of each track at a certain bitrate actually varies. Thus, a more accurate and desirable sales model, *multi-quality* instead of multi-bitrate, is still not available due to the lack of a clear link between scalable audio and its perceptual quality, i.e., the level of perceptual quality can be achieved at a certain scalable bitrate.

In this thesis, we provide solutions to all these critical issues. The research that leads to these solutions aims to equip MPEG-4 SLS with much better perceptual quality and more robust features, so that the users can benefit from the convenience of the universality as well as the excellent performance in terms of both the quality and compression of this codec.

1.3 Original Contributions

The major contributions of this dissertation are summarized as follows:

1. Thorough statistical analysis and testings are conducted on the influence of the rounding errors introduced by IntMDCT under the perceptual (lossy) audio coding scenario. Based on the results, it is found that the rounding errors of IntMDCT will not degrade the perceptual quality of decoded audio under standard playback circumstances. It is therefore concluded that MDCT and IntMDCT filterbanks are interchangeable in the lossy coding scenario. This conclusion proves the validity of using IntMDCT only in a scalable lossless audio coder.

2. Perceptually enhanced prioritized bit-plane audio coding algorithms are proposed for the non-core and low-core-bitrate mode of SLS. The bit-planes are prioritized with optimized parameters according to the energy distribution in different frequency regions. Based on the statistical modelling of the frequency spectrum, a much simplified implementation of prioritized bit-plane coding is integrated in SLS structure by replacing the sequential bit-plane coding in the enhancement layer. The main feature of this enhancement is that the complexity is extremely low. There is no modification to the standard SLS decoder and no extra side information is needed. Extensive experimental results show that the perceptual quality of SLS with non-core and very low core bitrate is improved significantly in a wide range of bitrate combinations by using the proposed method. Fully scalable audio coding up to lossless with much enhanced perceptual quality is thus achieved.

3. A perceptually enhanced stereo bit allocation algorithm for fully scalable audio coding is presented. According to the bit-plane levels in different channels, the bitrate is allocated in a much more efficient manner. Experimental results show that the proposed method significantly improves the perceptual quality of the fully scalable audio at various bitrates without introducing any new side information.

4. A "smart" function is designed for SLS, which is recently deployed by online music stores for multi-quality music sales. With a low quality audio format and its original format inputs, the proposed smart enhancing process enables a scalable encoder to automatically encode the minimum amount of enhancement for the low quality audio to obtain a *transparent quality* that is same as the CD quality. In addition, a bitrate estimation model is proposed. The model enables the estimation of the enhancing bitrate from two parameters extracted from the original and the low quality perceptual audio formats without real encoding. This estimation model can be applied to truncate a lossless audio format into the transparent quality lossy format. Evaluation results show that the smart enhancing process achieves an average of 19.8% bitrate saving compared to the traditional fixed bitrate setting without quality degradation. It is also shown that the estimation model proposed is able to accurately predict the necessary enhancing bitrate without real encoding.

1.4 Thesis Organization

The rest of the thesis is organized as follows.

In the next Chapter, an overview on scalable audio coding, especially MPEG-4

SLS, is given. The basic features, applications and the key technologies are introduced. The performance of SLS in terms of the perceptual quality and compression rate is evaluated as well.

The following four chapters include the main contributions of this thesis, which intend to solve the key issues of SLS. Chapter 3 addresses the concern on integer filterbank by analyzing the performance of the IntMDCT filterbank deployed by SLS in a lossy coding scenario. In Chapter 4, perceptually enhanced bit-plane coding algorithms are proposed and evaluated. A stereo bitrate allocation algorithm is described in Chapter 5 and a smart function for SLS in online music store application is elaborated in Chapter 6.

Finally, Chapter 7 presents the conclusions and recommendations for future research.

Chapter 2

Scalable Lossless Audio Coding

2.1 History of Scalable Audio Coding

Scalable audio coding (SAC) allows an encoder to compress data at a high/lossless bitrate and a receiver to decode the compressed signal at different fidelity levels according to the specific applications and user context requirements as well as available device capabilities. This scalability is very important for the co- and inter-operability of today's myriad of multimedia applications spanning across various media platforms including cell phone networks, local and wide area computer networks, and television and radio broadcasting networks; each one often requiring a different range of coding bitrates. Moreover, for heterogenous networks such as the Internet, audio coding with FGS maintains uninterrupted service in the often highly-congested and unstable channel traffic, with capability to demonstrate the efficient use of channel bandwidth. Scalability is also very useful in archiving where all the data are stored in a single, uniform format with a high/lossless bitrate, rather than with varying encoding bitrates.

The essential idea of the initial approaches [21–27] in SAC is basically a structure with perceptual or speech core layer and several enhancement layers in terms

of higher sampling rate or finer quantization. The first approach of SAC is brought in [21] and later implemented in [22], with a combined structure of three perceptual codecs operating at different sampling frequencies and bitrates. Specifically, the input of the second and third stage are the residual signals between the input and output of the previous stage. This is further improved in [23] by adopting a speech core coder and up-sampled perceptual enhancement layers with a *frequency selective switch*. This algorithm is later standardized in [14] as *scalable advanced audio coding* (SAAC) tool. Besides signal to noise ratio and bandwidth, the scalability in numbers of channels is implemented in [24]. With the development of *transform–domain weighted interleave vector quantization* (TWIN-VQ) in [25], the TWIN-VQ coder is used as the core coder in [26] and [27]. All these scalable coders have the characteristic of coarse granular scalability.

FGS, where increases in coding quality can be achieved with small increments in bitrate, is desirable in numerous applications. The most common way of implementing FGS in audio coding is through bit-plane coding. There are other approaches, for example that which uses a two-dimensional transform [28] where the data is progressively ordered in the bitstream according to perceptual relevance. The state-of-the-art FGS scalable audio codec is *bit-sliced arithmetic coding* (BSAC) [29] which adopts bit-plane coding on quantized spectral data. Scalability of 1kbps per channel can be achieved with sufficient coverage on the statistics of bit-slices through arithmetic coding models. As a result of the increased computational complexity arising from an increasing number of layers, BSAC is mainly used for scalability in a relatively limited bitrate range of 16 to 64kbps.

With the belief that more efficient bit-plane coding can be achieved by assigning different priorities to the bit-planes, numerous approaches have been proposed to implement the idea of *prioritized* bit-plane coding. These approaches can be

grouped into three basic categories. The first category covers the approaches inspired by tree-based significance mapping techniques including *embedded zerotrees wavelet* (EZW) [30] and *set partitioning in hierarchical trees* (SPIHT) [31] in wavelet image compression. In conjunction with wavelet packet transform based audio coding [32–34], ZTW and SPIHT are applied to achieve the scalable audio coding format in [35–37]. Besides the tree-based structure, the psychoacoustic information of a signal is another attractive characteristic that can be used to enhance the performance of bit-plane coding. Such information are being utilized in prioritized bit-plane coding approaches of the second category. Here, the priorities of bit-plane coefficients are assigned according to the psychoacoustic information [38], as done in [39] where the scalefactors from the AAC quantization are used as a guide to coding orders. The third group is composed of approaches that combine both psychoacoustic information and significant map structure together, as have been adopted in [40–42].

The scalable audio coding schemes mentioned above mainly focus on the scalability of low bitrate perceptual audio. With the proliferation of broadband access and the decline of storage costs, there is a trend in providing consumers with a rich media experience of extremely high fidelity. In the realm of audio, this is achieved by employing lossless formats with high resolutions and/or high sampling rates. It is in this context that the ISO/MPEG audio standardization group began exploring technologies for lossless and near lossless coding of audio signals by issuing a CfP for relevant technology in 2002 [13]. The key outcome of this CfP, the MPEG-4 SLS [15], was released as a standard audio coding tool in June 2006. It allows the scaling up of a perceptually coded representation such as the MPEG-4 AAC [14] to a fully lossless representation with a wide range of intermediate bitrate representations.

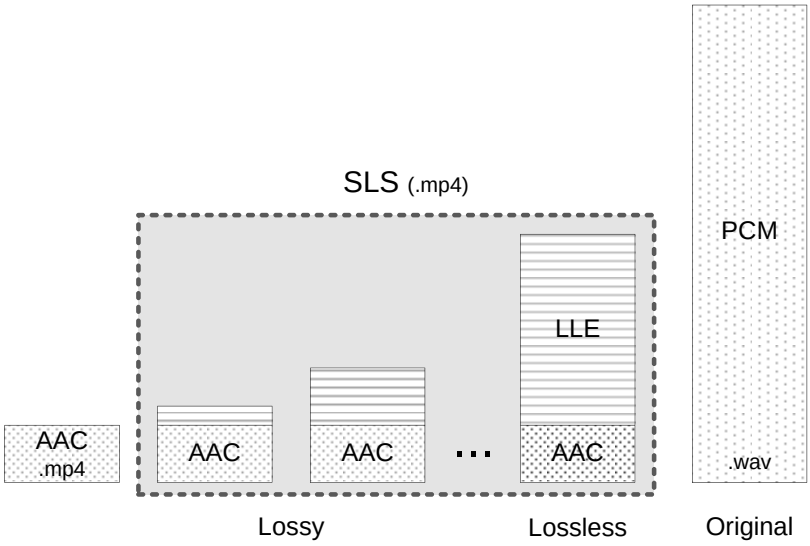


Figure 2.1: SLS format.

In December 2007, as the first realized application, SLS was deployed by Asia's largest online music store Soundbuzz Pte Ltd.

2.2 Basic Structure

The structure of SLS combines the scalable coding approaches of "perceptual core + enhancement layers" scheme and bit-plane coding to achieve backward compatibility with the MPEG perceptual audio coder and FGS. The format of SLS compared with other common formats, including AAC and PCM are shown in Figure 2.1. Basically, the core layer in SLS can be MPEG AAC codec or another perceptual scalable codec such as the scalable AAC and BSAC. *Lossless enhancement* (LLE) are achieved through sequential bit-plane coding of the residual signals between the original and the AAC encoded spectrum. Note that the AAC core coder can be turned off in the *non-core* mode of SLS where full scalability can be achieved through sequential bit-plane coding.

The block diagram of SLS codec is depicted in Figure 2.2. In the SLS encoder,

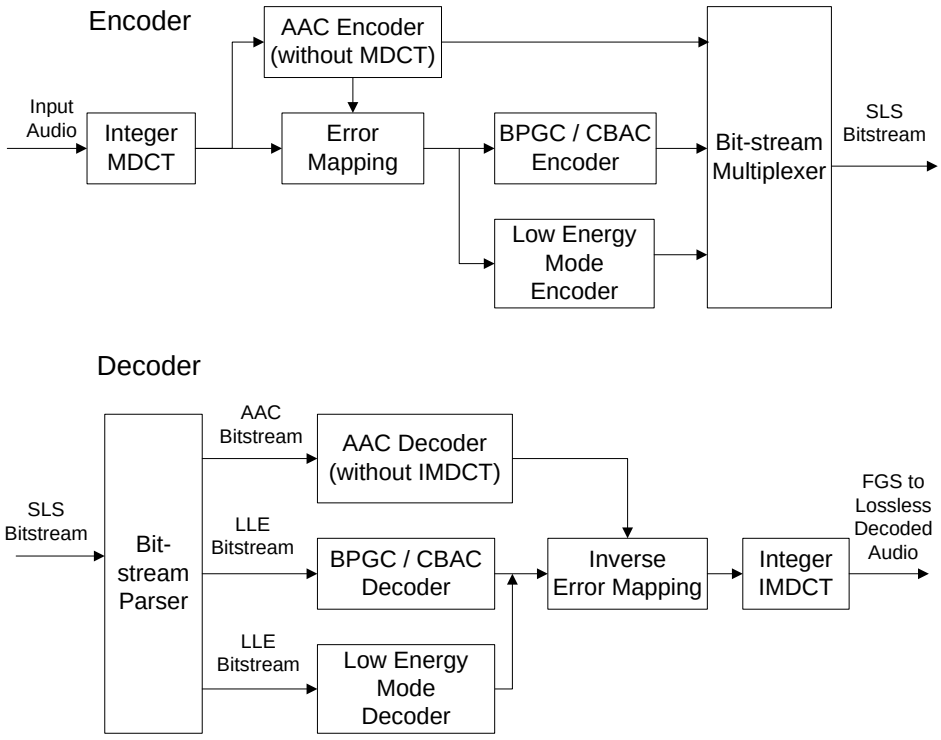


Figure 2.2: Structure of SLS encoder and decoder.

the input audio in integer PCM format is losslessly transformed into the frequency domain by using IntMDCT [16] [18] which is a lossless integer to integer transform that approximates the normal MDCT transform. If the Mono IntMDCT is used for the left and the right channel, the integer M/S [43] processing has to be applied to the *scalefactor bands* (sfbs) where the M/S flag is set to 1. The Stereo IntMDCT delivers by default an M/S spectrum. The resulting coefficients are then passed on to the AAC encoder to generate the core layer AAC bitstream. In the AAC encoder, transformed coefficients are first grouped into sfbs which are generally fixed at 49 for a sampling rate of 48 or 44.1kHz. The coefficients are then quantized with a non-uniform quantizer, usually with different quantization steps in different sfbs to shape the quantization noise so that it can be best masked.

In order to efficiently utilize the information of the spectral data that has been

carried in the core layer bitstream, error-mapping procedure is employed to generate the residual spectrum coded in the LLE layer. This is done by subtracting the AAC quantized spectrum from the original spectrum. For $k = \{0, 1..., N-1\}$ where N is the dimension of IntMDCT, the residual signal $e[k]$ is computed by

$$e[k] = \begin{cases} c[k] & i[k] = 0 \\ c[k] - \lfloor thr(i[k]) \rfloor & i[k] \neq 0 \end{cases} . \tag{2.1}$$

Here $c[k]$ is the IntMDCT coefficient, $i[k]$ is the quantized data vector produced by the AAC quantizer, $\lfloor \cdot \rfloor : \mathbb{R} \rightarrow \mathbb{Z}$ is the flooring operation that rounds off a floating-point value to its nearest integer with a smaller amplitude and $thr\,(i[k])$ is the low boundary (towards-zero side) of the quantization interval corresponding to $i[k]$, which is calculated as

$$thr(i[k]) = \begin{cases} \mathrm{sgn}(i[k]) \left[\sqrt[4]{2^{scale\_factor(s)}} (|i[k]| - C)^{4/3} \right] , & i[k] \neq 0 \\ 0 , & i[k] = 0 \end{cases} \tag{2.2}$$

where $scale\_factor[s]$ is the scale factor that determines the quantization step size for sfb s, and the rounding offset is given by $C = 0.4054$.

The residual spectrum is then coded using BPGC [17] combined with CBAC and LEMC [20] to generate the scalable LLE layer bitstream. In the final step of the encoding process, the output of LLE bitstream is multiplexed with the core AAC bitstream to produce the final SLS bitstream.

Basic knowledge of perceptual audio coding can be found in [11] [45] and some references of noiseless coding include [46–49].

2.3 Key Technologies

Basically, the key technologies of SLS include the IntMDCT transform and the bit-plane coding (includes entropy coding) block which comprises BPGC, CBAC and LEMC.

2.3.1 Integer MDCT

The *modified discrete cosine transform* (MDCT) [50, 51] has been widely used in transform audio coders such as *MPEG–1 layer III* (MP3) and the MPEG AAC. It is a Fourier-related transform based on the *type–IV discrete cosine transform* (DCT-IV), with the additional property of being lapped: it is designed to be performed on consecutive blocks of a larger data set, where subsequent blocks are overlapped so that the last half of one block coincides with the first half of the next block. This overlapping analysis windowing is very attractive to audio coding since it significantly mitigates the blocking artifacts. To achieve critical sampling, a subsampling operation is performed in the frequency domain, and the aliasing resulting from this subsampling operation is subsequently cancelled in the time domain by an "overlap and add" technology, which is called *time-domain aliasing cancellation* (TDAC) [50]. Another attractive feature of the MDCT is the availability of *fast Fourier transform* (FFT)–based fast algorithms such as those in [52, 53] that make it viable for practical applications.

Like most transforms, the MDCT produces floating point spectral values even for integer input samples. For this reason, the MDCT is not directly applicable for lossless coding, as it will generally result in data expansion instead of compression if those floating point spectral values are directly coded to a precision that ensures lossless decoding.

The IntMDCT transform, introduced in [16] and later significantly improved in [18,54–56], provides a lossless integer to integer transform that approximates the normal MDCT transform. IntMDCT is obtained by factorization of the MDCT into a cascade of Givens rotations, which are then implemented by using the "lifting step" [57] or "ladder network" [58] that has been applied to construct a number of integer transforms such as integer FFT [59] and the integer DCT [60]. The lifting scheme produces integer outputs if the inputs are integers.

A description on the IntMDCT implementation based on the lifting scheme will be provided in this section. Improved methods which are achieved by extending the idea of lifting scheme to the multi–dimensional case will be discussed as well.

2.3.1.1 MDCT by Windowing and DCT–IV

The MDCT is performed on a block of time domain samples with length $2N$ to produce N transform coefficients. As two succeeding analysis blocks are 50% overlapped, only N new time domain samples are taken for each transform block. Given an input block , the MDCT coefficients are calculated as:

$$X[m] = \sqrt{\frac{2}{N}} \sum_{k=0}^{2N-1} w[k]x[k] \cos \frac{(2k+1+N)(2m+1)\pi}{4N}, \quad m = 0, ..., N-1, \quad (2.3)$$

where $w[k]$, $k = 0, ..., 2N-1$ is a window function for a smooth overlapping of blocks. The inverse MDCT is given by:

$$y[k] = w[k] \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X[m] \cos \frac{(2k+1+N)(2m+1)\pi}{4N}, \quad k = 0, ..., 2N-1. \quad (2.4)$$

After the forward and inverse MDCT operation a time domain aliasing is introduced in the restored time domain signal $y[k]$. This aliasing error is cancelled by

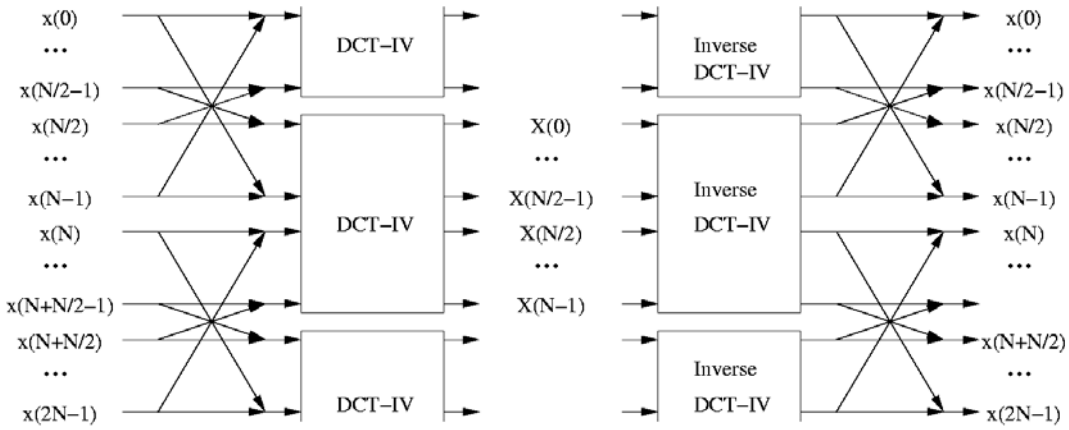


Figure 2.3: MDCT and inverse MDCT by windowing and DCT–IV [18].

adding the outputs of the inverse MDCT of two adjacent blocks $t-1$ and t as

$$x'_t[k] = y_t[k] + y_{t+1}[N+k], \quad k = 0, ..., N-1. \tag{2.5}$$

To achieve perfect reconstruction, $w[n]$ should further be subjected to the following constraints:

$$w[2N-1-k] = w[k] \tag{2.6}$$

and

$$w^2[k] + w^2[k+N] = 1 \tag{2.7}$$

for $k = 0, ..., N-1$. A typical example of the window function that fulfills the constraints is the sine window defined as:

$$w[k] = \sin\left[\left(k + \frac{1}{2}\right)\frac{\pi}{2N}\right]. \tag{2.8}$$

It has previously been shown in [16, 51] that the MDCT can be factorized into a window modulation stage and DCT–IV [61] (as shown in Figure 2.3). To elaborate, the time domain modulated signal $\widetilde{x}_t[k]$ is defined from an input signal block $x_t[k]$

as

$$\widetilde{x}_t[k] = w\left[\frac{N}{2}+k\right] x_t\left[\frac{N}{2}+k\right] - w\left[\frac{N}{2}-1-k\right] x_t\left[\frac{N}{2}-1-k\right], \tag{2.9}$$

$$\widetilde{x}_t[N-1-k] = w\left[\frac{3N}{2}+k\right] x_t\left[\frac{3N}{2}+k\right] + w\left[\frac{3N}{2}-1-k\right] x_t\left[\frac{3N}{2}-1-k\right], \tag{2.10}$$

$$k = 0, ..., \frac{N}{2}-1.$$

The MDCT is now reduced to:

$$X[m] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} -\widetilde{x}_t[N-1-k] \cos\left[(2k+1)(2m+1)\frac{\pi}{4N}\right], \tag{2.11}$$
$$m = 0, ..., N-1$$

which is in fact the application of a length N DCT–IV to $\widetilde{x}_t[N-1-k]$, $k = 0, ..., N-1$. Meanwhile we observe that due to the overlap nature of MDCT, the signal $x_t[k]$, $k = 0, ..., N-1$ is also used for MDCT block $t-1$. From Eqn.(2.10) we have

$$\begin{aligned}\widetilde{x}_{t-1}[N-1-k] &= w\left[\frac{3N}{2}+k\right] x_t\left[\frac{N}{2}+k\right] + w\left[\frac{3N}{2}-1-k\right] x_t\left[\frac{N}{2}-1-k\right], \\ &= w\left[\frac{N}{2}-1-k\right] x_t\left[\frac{N}{2}+k\right] + w\left[\frac{N}{2}+k\right] x_t\left[\frac{N}{2}-1-k\right],\end{aligned} \tag{2.12}$$

$$k = 0, ..., \frac{N}{2}-1.$$

Combining this with Eqn.(2.9) we observe that the time domain modulated signal can be prepared from the time domain signal by:

$$\begin{pmatrix} \widetilde{x}_t[k] \\ \widetilde{x}_{t-1}[N-1-k] \end{pmatrix} = \begin{pmatrix} w[\frac{N}{2}+k] & -w[\frac{N}{2}-1-k] \\ w[\frac{N}{2}-1-k] & w[\frac{N}{2}+k] \end{pmatrix} \begin{pmatrix} x_t[\frac{N}{2}+k] \\ x_t[\frac{N}{2}-1-k] \end{pmatrix}. \tag{2.13}$$

From the TDAC condition in Eqn.(2.7),

$$w^2\left[\frac{N}{2}+k\right] + w^2\left[\frac{N}{2}-1-k\right] = 1. \tag{2.14}$$

Therefore the window modulation process in Eqn.(2.13) can be written as an application of Givens rotation [62]

$$\begin{pmatrix} \cos\alpha_k & \sin\alpha_k \\ \sin\alpha_k & \cos\alpha_k \end{pmatrix} \tag{2.15}$$

where the angles α_k are given by

$$\alpha_k = \arctan\frac{w[\frac{N}{2}-1-k]}{w[\frac{N}{2}+k]}, \quad k = 0, ..., \frac{N}{2}-1. \tag{2.16}$$

The inverse MDCT can be obtained by reversing the procedure described above, where the inverse DCT-IV is the DCT–IV itself, and the window modulation is reverted by applying Givens rotation with angles $-\alpha_k$, $k = 0, ..., \frac{N}{2}-1$.

The coefficients of the DCT–IV with length N produces an orthogonal $N \times N$ matrix $\mathbf{C}_N^{IV}$ which is given by

$$\mathbf{C}_N^{IV} = \sqrt{\frac{2}{N}}\cos\left[\frac{\pi}{4}(2k+1)(2m+1)\right], \tag{2.17}$$

$$m, k = 0, ..., N-1.$$

It is well-known that an $N \times N$ orthonormal matrix can be factorized into $\frac{N(N-1)}{2}$ Givens rotations. It is possible, however, to reduce the number of the Givens rotations to the magnitute of $O(N \log_2 N)$ by using some fast algorithms for DCT–IV. Two examples using fast algorithms include the complex FFT based fast DCT–IV algorithm described in [51] and the direct factorization method proposed in [61].

The Givens rotations can be implemented as a reversible integer to integer mapping by using the lifting scheme. In a lifting scheme, the Givens rotation is first factorized into three lifting steps as given in the following formula:

$$\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & cos\alpha \end{pmatrix} = \begin{pmatrix} 1 & \frac{\cos\alpha - 1}{\sin\alpha} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \sin\alpha & 1 \end{pmatrix} \begin{pmatrix} 1 & \frac{\cos\alpha - 1}{\sin\alpha} \\ 0 & 1 \end{pmatrix}. \quad (2.18)$$

In order to have a reversible integer to integer mapping, in each of the lifting steps a rounding operation $\mathbf{Round} : \mathbb{R} \rightarrow \mathbb{Z}$ is used after coefficient multiplication.

The three lifting steps that implemented the Givens rotation is shown in Figure 2.4. In the case that the rounding operation, **Round**, is odd symmetric its inverse transform is simply the lifting scheme for Givens rotation with angle $-\alpha$. The implementation of the IntMDCT is now straightforward as MDCT has been factorized into a cascade of Givens rotations. This is achieved by replacing each Givens rotation with the lifting scheme described above. The IntMDCT will generate integer outputs for integer input values, and the whole process can be losslessly inverted by applying the inverse lifting scheme in the reverse order in the reverse transform.

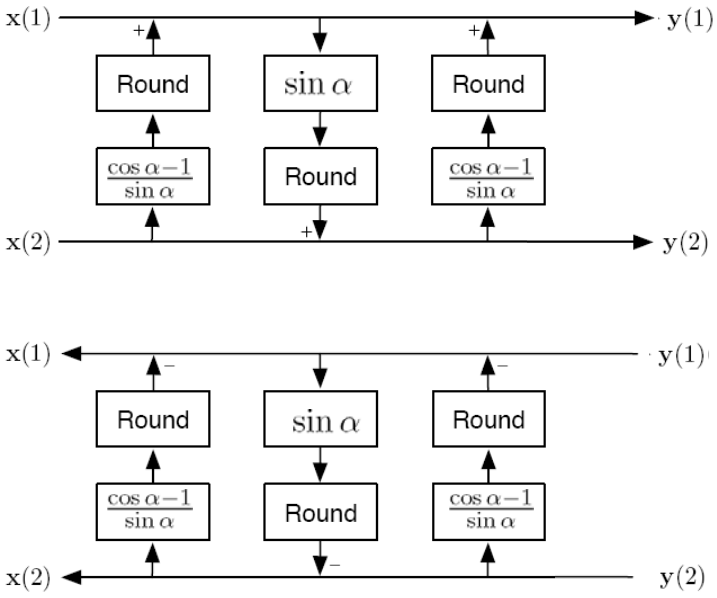


Figure 2.4: Lifting steps.

2.3.1.2 Multi-Dimensional Lifting Scheme

The lifting steps can be extended to *multi-dimensional lifting* (MDL) by replacing the inputs for each lifting step with vector values. The resultant multi-dimensional lifting steps are shown as follows:

$$\begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_N & 0 \\ \mathbf{S} & \mathbf{I}_N \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix}, \tag{2.19}$$

where $\mathbf{I}_N$ is an $N \times N$ identity matrix, $\mathbf{S}$ is an $N \times N$ matrix and $\mathbf{x}$s, $\mathbf{y}$s are N dimensional vectors. Similarly, integer approximation for this multi-dimensional lifting step can be obtained as follows:

$$\mathbf{y}_1 = \mathbf{x}_1, \tag{2.20}$$

$$\mathbf{y}_2 = \mathbf{x}_2 + (\mathbf{S} \cdot \mathbf{x}_1), \tag{2.21}$$

where $r : \mathbb{R}^N \rightarrow \mathbb{Z}^N$ is the rounding operation that maps a floating-point vector to an integer one. Clearly, an integer implementation for a transform $\mathbf{T}$ can be achieved in this manner if it can be factorized into a cascade of multi-dimensional lifting steps.

The main advantage of the MDL based approaches is that it significantly reduces the number of rounding operations from the magnitude of $O(N \log_2 N)$ as in the conventional lifting scheme described in previous section (which will be referred to as the *single dimensional lifting* (SDL) scheme hereafter) to the magnitude of $O(N)$. This helps to improve the accuracy of the integer transforms in particular for those with large orders.

The IntMDCT algorithm can be further simplified if we allow the IntMDCT to be performed simultaneously on more than one data block. It is observed from the fact that for any $N \times N$ invertible matrix $\mathbf{T}_N$ the following factorization holds (as shown in Figure 2.5):

$$\begin{pmatrix} 0 & \mathbf{T}_N \\ \mathbf{T}_N^{-1} & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_N & 0 \\ -\mathbf{T}_N^{-1} & \mathbf{I}_N \end{pmatrix} \begin{pmatrix} \mathbf{I}_N & \mathbf{T}_N \\ 0 & \mathbf{I}_N \end{pmatrix} \begin{pmatrix} \mathbf{I}_N & 0 \\ \mathbf{T}_N^{-1} & \mathbf{I}_N \end{pmatrix}. \tag{2.22}$$

Since $\left(\mathbf{C}_N^{IV}\right)^{-1} = \mathbf{C}_N^{IV}$, it is straightforward to derive the following factorization for a cascade of two DCT–IV matrix:

$$\begin{pmatrix} 0 & \mathbf{C}_N^{IV} \\ \mathbf{C}_N^{IV} & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_N & 0 \\ -\mathbf{C}_N^{IV} & \mathbf{I}_N \end{pmatrix} \begin{pmatrix} \mathbf{I}_N & \mathbf{C}_N^{IV} \\ 0 & \mathbf{I}_N \end{pmatrix} \begin{pmatrix} \mathbf{I}_N & 0 \\ \mathbf{C}_N^{IV} & \mathbf{I}_N \end{pmatrix} \tag{2.23}$$

for which the MDL scheme is directly applicable. The main disadvantage of this approach, however, is that the additional delay introduced by this algorithm as two data blocks instead of one are needed to be buffered for each IntMDCT operation.

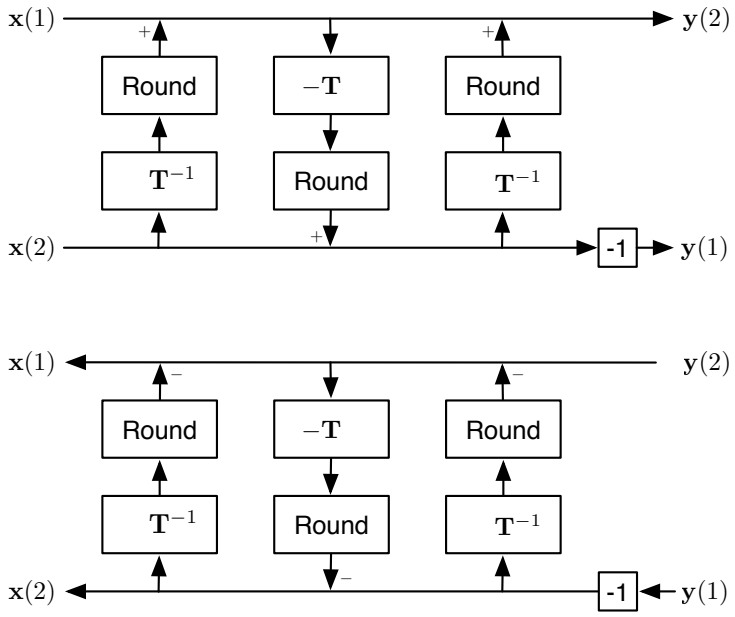


Figure 2.5: IntMDCT based on DCT matrix pair [18].

In the case of jointly stereo audio coding this additional delay can be avoided by performing the transform simultaneously on data blocks from two channels in parallel.

2.3.2 Bit-Plane Coding

In SLS, the IntMDCT spectral data of the input audio are first coded with an MPEG-4 AAC encoder to generate an embedded AAC bit-stream, while the residual spectrum between the IntMDCT spectral data and the their quantized values by the embedded AAC encoder is subsequently coded with BPGC [17] to produce the fine granular scalable lossy to lossless portion of the final lossless bit-stream. As the frequency assignment rule of BPGC is derived from the Laplacian *probability density function* (pdf), BPGC only delivers excellent compression performance when the sources are Laplacian or near-Laplacian distributed. It has been shown

in [44] that in most cases, the IntMDCT spectral data of audio are closely approximated by the Laplacian distribution. However, it is also found that for some music items, there always exist some "silence" time/frequency (T/F) regions where the IntMDCT spectral data are in fact dominated by the rounding errors of the IntMDCT algorithm. In order to improve the coding efficiency of SLS, LEMC is also adopted for coding IntMDCT spectral data from these "silence" T/F regions. It is possible to improve the coding efficiency of BPGC by further incorporating more sophisticated probability assignment rules that take into account the dependencies of the distribution of IntMDCT spectral data to several contexts such as their frequency locations, or the amplitudes of adjacent spectral lines. Existence of these dependencies will lead to a small variance in the probability distribution of the bit-plane symbols of IntMDCT spectral data, which can be effectively captured by using CBAC.

2.3.2.1 Bit-Plane Golomb Code

It is mentioned in [44] that both the IntMDCT coefficients and the error signal generated from the error mapping of SLS can be approximated as Laplacian distribution. The BPGC coding process is basically a bit-plane coding scheme where the bit-plane symbols are arithmetic coded with a structural frequency assignment rule, based on the assumption that the source is Laplacian distributed. Each element $e[k]$ in Eqn.(2.2) is first represented in a binary format as

$$e[k] = (2s[k] - 1) \sum_{j=0}^{M-1} b[k, j] \cdot 2^j, \quad k = 0, ..., N-1, \tag{2.24}$$

where M is the MSB for $e[k]$ that satisfies $2^{M-1} \leq \max\{|e[k]|\} < 2^M$, $k = 0, ..., N-1$; $b[k, j] \in \{0, 1\}$ is the bit-plane symbol and j is the bit-plane number. $s[k]$ denotes

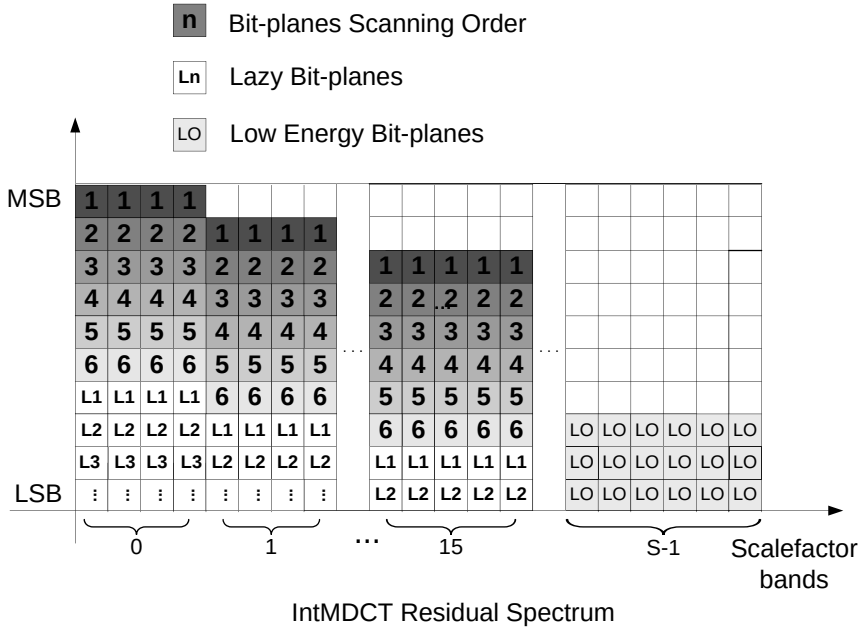


Figure 2.6: Bit-plane scan process in SLS.

the sign symbol, i.e.,

$$s[k] \triangleq \begin{cases} 1 & e[k] \geqslant 0 \\ 0 & e[k] < 0 \end{cases}, \quad k = 0, ..., N-1 \, . \tag{2.25}$$

Here, The bit-plane symbols are then scanned and coded (see Figure 2.6) from the MSB to the LSB over all the elements, and coded by using arithmetic code with a structural frequency assignment Q_j^L given by

$$Q^L[j] = \begin{cases} \frac{1}{1+2^{2^{j-L}}} & j > L \\ \frac{1}{2} & j < L \end{cases} \tag{2.26}$$

where the Lazy plane parameter L can be selected using the adaptation rule

$$L = \min\{L' \in \mathbb{Z} | 2^{L'+1} N \geq A\}. \tag{2.27}$$

Here A is the absolute sum of the data vector $\mathbf{e}$. It enters a "lazy mode" (since coding of binary symbol with probability assignment $\frac{1}{2}$ can be implemented by directly outputting input symbols to a compressed bitstream) for bit-planes $j < L$.

2.3.2.2 Context-Based Arithmetic Code

The CBAC coding method used in SLS introduces three types of the contexts. These include the *frequency band* (FB) context, the *distance to lazy* (D2L) context, and the *significant state* (SS) context. The criterion for selecting these contexts is to find those that are most "relevant" to the distribution of the bit-plane symbols. The detailed context assignments are summarized as follows:

- **Frequency Band**

  It was found in [20] that the probability distribution of bit-plane symbols of IntMDCT varies for different frequency bands. Therefore, in CBAC the IntMDCT spectral data are classified into three different FB contexts, namely, low band ($0 \sim 4$ kHz, FB = 0), mid band (4 kHz $\sim$ 11 kHz, FB = 1) and high band (above 11 kHz, FB = 2).

- **Distance to Lazy**

  The D2L context is defined as the distance of the current bit-plane j to the BPGC Lazy plane parameter L, as defined in the following equation.

$$D2L = \begin{cases} 3 - j + L & j - L \geq -2 \\ 6 & else \end{cases} \tag{2.28}$$

  The rationale follows the BPGC frequency assignment rule (2.26), which is based on the fact that the skewness of the probability distribution of the bit-plane symbols from a source with Laplacian or near-Laplacian distribution

tends to decrease as the number of D2L decreases. To reduce the total number of the D2L context, all the bit-planes with $D2L < -2$ are grouped into one context where all the bit-plane symbols are coded with probability 0.5.

- **Significant State**

  Due to the leakage of the IntMDCT filterbank, the amplitudes of adjacent spectral lines are correlated. In addition, the amplitude of the IntMDCT spectrum is also highly correlated with the quantization interval of the SLS core quantizer if it is present. The SS context is designed to capture these correlations. The detailed configuration of the SS context can be found at [15].

2.3.2.3 Low Energy Mode Coding

For some low energy regions, the IntMDCT spectral data are in fact dominated by the rounding errors accumulated from the rounding operation in the IntMDCT algorithm with distributions far away from the Laplacian distribution. In order to improve the coding efficiency, the BPGC/CBAC coding process is replaced with LEMC.

LEMC is used for sfb for which the BPGC parameter L is smaller or equal to 0. In LEMC, the amplitude of the residual spectral data $e[k]$ is first converted into unitary binary string $\mathbf{b} = \{b[0], b[1], ..., b[pos], ...\}$ as illustrated in Table 2.1, with M being the maximum bit-plane. It can be seen that the probability distributions of these symbols depend on the position (pos), and the distribution of $e[k]$:

$$Pr\{b[pos] = 1\} = Pr\{|e[k]| > pos \mid |e[k]| \geq pos\},\ 0 \leq pos < 2^M. \qquad (2.29)$$

$b[pos]$ is then arithmetic coded conditioned on its position pos and the BPGC

Table 2.1: Binarization of IntMDCT error spectrum at low energy mode. From [20].

Amplitude of $e[k]$	Binary string $\{b[pos]\}$
0	0
1	1 0
2	1 1 0
...	...
$2^M - 2$	1 1 ... ... 1 0
$2^M - 1$	1 1 ... ... 1 1
pos	0 1 2 3 ...

parameter L with a trained frequency table.

2.4 Performance

The performance of SLS is summarized in this section.

Table 2.2: Test items (stereo)

no.	Items (.wav)	no.	Items (.wav)	no.	Items (.wav)
1	avemaria	6	cymbal	11	haffner
2	blackandtan	7	dcymbals	12	mfv
3	broadway	8	etude	13	unfo
4	cherokee	9	flute	14	violin
5	clarinet	10	fouronsix	15	waltz

Firstly, the performance of SLS in terms of lossless compression ratio for standard lossless audio test sequences (as listed in TABLE 2.2) is evaluated. The compression ratio is defined as

$$\text{Compression Ratio} = \frac{\text{Original Size}}{\text{Compressed Size}} \tag{2.30}$$

Four sets of formats including 48kHz/16bit, 48kHz/24bit, 96kHz/24bit and 192kHz/24bit at different *oversamping factor* (osf, see [15]) are tested and the results are shown in TABLE 2.3.

The compression performances of SLS (osf = 1) using BPGC and CBAC are compared in Table 2.4. An average of 1% improvement can be achieved by using CBAC.

Table 2.3: Lossless compression ratio performance of MPEG-4 SLS

	osf = 1	osf = 2	osf = 3	osf = 4
48kHz/16bit	2.15	2.17	2.19	2.20
48kHz/24bit	1.56	1.57	1.58	1.58
96kHz/24bit	2.08	2.08	2.12	2.13
192kHz/24bit	2.52	2.54	2.60	2.63
Overall	2.08	2.08	2.11	2.12

Table 2.4: Compression improvement of CBAC comparing with BPGC

	BPGC	CBAC	% Improvement
48kHz/16bit	2.12	2.15	1.44
48kHz/24bit	1.55	1.56	0.23
96kHz/24bit	2.05	2.08	1.07
192kHz/24bit	2.49	2.52	1.26
Overall	2.05	2.08	1.00

The computational complexity for SLS is evaluated by counting the total numbers of standard instructions (multiplications, additions, bit-shifts, comparisons, memory transfers, etc) required for performing the decoding process on a generic 32-bit fixed-point CPU. For the purpose of comparison we compute a weighted average number (with weights: 14.0 - multiplications, 56.0 - divisions, 4.0 - shifts, and 0.5 - for additions, comparisons, and memory-access operations) corresponding to the estimated latencies of instructions in Intel Pentium processors [63]. The complexity of deterministic algorithms is evaluated exactly, while the complexity of arithmetic coding engines is estimated as the upper bound for the average complexity, assuming it achieves compression ratio of 2:1.

The complexity in terms of combined pentium latency per decoded audio sample of the SLS decoder is listed in Table 2.5 [64]. To facilitate DSP and other

Table 2.5: Complexity of SLS decoder in terms of combined pentium latency (cycle/sample)

	SLS with 128 kbps AAC core	SLS non-core
48kHz/16bit	642.29	677.575
48kHz/24bit	642.29	677.575
96kHz/24bit	724.42	677.575
192kHz/24bit	729.66	677.575

Table 2.6: ROM requirement of SLS decoder

	SLS + AAC core	SLS non-core
ROM	45K bytes	4K bytes

potential embedded implementation of SLS, only the estimated maximum value of the complexity is reported here.

Memory requirements of an SLS decoder are summarized in Table 2.6. The detailed analysis can be found in [64].

The perceptual quality of SLS with different configurations is evaluated and compared with that of AAC-LC by using objective tests (some subjective tests are shown in Chapter 4). Only the 48kHz/16bit testing set (most commonly used) are presented. The configuration of the evaluation is listed as follows:

- The overall test bitrate range is from 64 to 384kbps (in total for stereo channels), with step size of 32kbps in the range of 64–256kbps and 64kbps in the range of 256–384kbps.

- The total bitrate comprises the core bitrate and enhancement (bit-plane coding) bitrate.

- The perceptual quality of the decoded audio is measured in terms of *objective difference grade* (ODG) scores using OPERA voice/quality analyzer [65] which performs ITU-R BS.1387 *perceptual evaluation of audio quality*

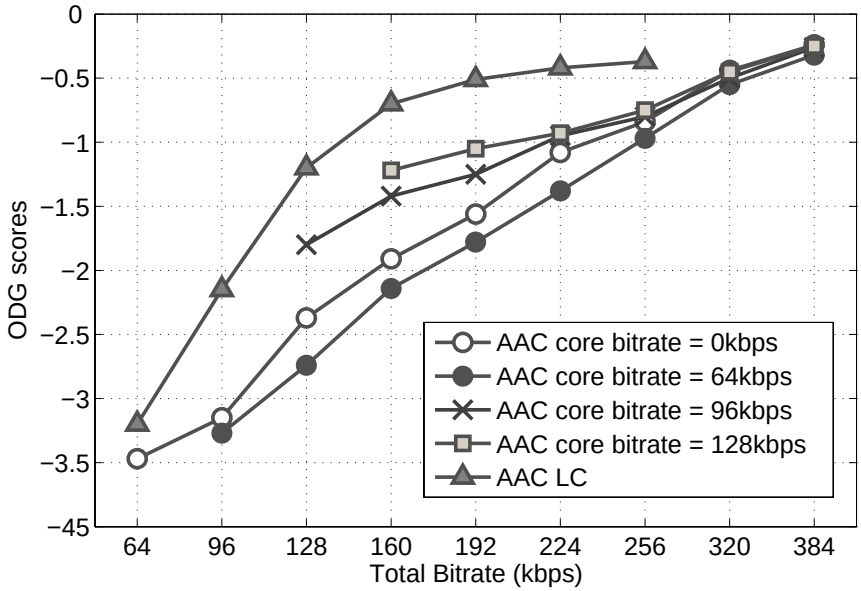


Figure 2.7: Perceptual quality performance of SLS at variable bitrate combinations.

(PEAQ) [66] test (the basic version). The PEAQ test makes use of a computational model of the human auditory system to compare the perceptual difference between the processed (coded) audio signal and the original one, and outputs the ODG score that mimics the listening tests rating. The grading scale of ODG ranges from -4 ("very annoying") to 0 ("imperceptible difference").

- For each bitrate combination, the ODG result is the average value of the scores for 15 test sequences as listed in TABLE 2.2.

The evaluation result is shown in Figure 2.7. It can be observed that with the same amount of total bitrates, SLS with adequate core bitrate ($\geq$ 96kbps) can achieve relatively more efficient perceptual improvements compared to the performance of SLS at a core bitrate of 64kbps and in non-core mode. However, the performance is still far from that of AAC-LC at the same bitrates. It is also observed that the performance of SLS with a low core bitrate is even worse than the non-core SLS.

This is due to the fact that when the AAC core bitrate is low, the enhancement of SLS is guided by the core information that is not perceptually optimized. As a result, the poor perceptually guided SLS enhancement scheme performs even worse than the naturally guided enhancement scheme (non-core). Thus, the remaining problem with the current structure is that the perceptual quality of SLS at low core bitrates is still far from optimum for most of the audio sequences.

2.5 Applications

As the primary functionality of SLS audio coding is lossless audio coding, it can be used in applications that require bit-exact reconstruction, such as studio operation, music disc delivery, audio archiving, etc. Due to its scalability feature, the SLS audio coding technology in fact fits into virtually every application that requires audio compression. Several potential application scenarios for SLS audio coding technology are listed below.

- **Studio Operations**

  The SLS audio coding technology is useful for storage of audio at various points in the studio operations such as recording, editing, mixing and pre-mastering as studio procedures are designed to preserve the highest levels of quality. The scalability of SLS also provides a nice solution for situations where the bandwidth is not sufficient to support lossless quality.

- **Archival**

  Archives of sound recordings are very common in studios, record labels, libraries, etc. These archives can be very large and compression can play an important role here. In addition, the scalability of SLS technology enables the

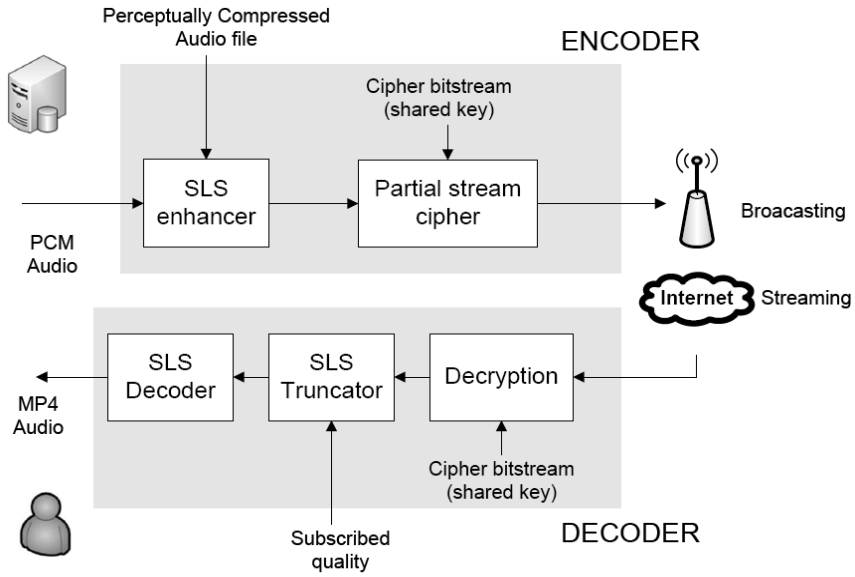


Figure 2.8: SLS coding system for quality on demand broadcasting/streaming.

possibility that low bitrate versions of the archived lossless audio items can be extracted at any time to allow applications such as remote data browsing.

- **Broadcast Contribution/Distribution Chain**

  In a broadcast environment, SLS audio coding technology could be used in all stages comprising archiving, contribution/distribution and emission [67]. In this broadcast chain, one main feature of the SLS technology can be used: In every stage where lower bit rates are required, the bit stream is just truncated, and therefore no re-encoding is required.

- **Streaming**

  The SLS audio coding technology delivers the vital bitrate scalability for streaming applications on channel with variable QoS conditions. Examples for this kind of streaming applications include the Internet audio streaming, multicast streaming applications that feeds several channels of differing capacity, etc.

- **Multi-Quality Online Music Store**

  By applying SLS file managing system, with one archival of the lossless audio file, the music store can offer multiple versions of audio files to the clients at various pricing. Besides, clients only need to keep one file format (the details of this application can be found in Chapter 6).

- **Quality on Demand Broadcasting/Streaming**

  SLS also enables an application that by sending one full quality audio signal from the server, the decoder at clients side can be managed to decode the quality according to the price subscribed (see Figure 2.8).

Chapter 3

On Integer MDCT in Scalable Lossless Coding

3.1 Background

The development of IntMDCT transform has led lossless audio coding into a new era. Most of the conventional audio coding schemes use filterbanks such as MDCT [50, 51] to obtain a blockwise frequency representation of the audio signal. These transforms usually produce floating point values even for integer input samples. For this reason, the MDCT cannot be used realistically for lossless coding as it will generally result in data expansion instead of compression if those floating point spectral values are directly coded to a precision that ensures lossless decoding. This problem can be solved by the introduction of IntMDCT algorithm, which was originally introduced in [16] by providing a lossless integer to integer transform that approximates the normal MDCT transform. It preserves most of attractive properties of MDCT, including the overlapping structure that provides better frequency selectivity than non-overlapping block transforms, time–domain aliasing cancellation and [50] critical sampling.

In [71], a scalable lossy to lossless audio codec structure was introduced which essentially utilized the IntMDCT. Moreover, the possibility of using IntMDCT as an approximation of MDCT for perceptual audio coders is considered and mentioned in [72,73]. But up to now, there is no analysis to justify this implementation. With this assumption, SLS RM employs only IntMDCT filterbanks for both lossy and lossless coding in its structure. This implementation may introduce some potential artifacts. Due to the rounding to integer operations that are present in the IntMDCT algorithm, there exist small additive errors or noise in the output of the IntMDCT compared with that of the floating-point MDCT. This is generally not a problem for lossless coding, since the errors produced by IntMDCT algorithm will eventually be cancelled at the reverse procedure, and the original signal can be restored losslessly at the decoder. However, in lossy operation, this is not the case. Due to the quantization procedure, the rounding errors can no longer be cancelled perfectly. As such, it is necessary to investigate the performance of IntMDCT filterbanks in perceptual coding context. It is the objective of this dissertation to investigate the effects of these noise components that are inherent in the IntMDCT which could potentially cause artifacts when perfect reconstruction operations are not present.

The rest of this Chapter is organized as follows: The potential problems which may be caused by the errors introduced from IntMDCT in lossy coding of SLS are described first. Detailed analysis and simulation results are given in Section 3.3, and Section 3.4 presents the subjective listening test results. Finally, Section 3.5 provides the conclusion.

3.2 Potential Artifacts Caused by IntMDCT in Lossy Coding of SLS

As shown in Figure 2.2, the IntMDCT filterbank is employed as the only filterbank for SLS. In the encoder, it is used as the filterbank for lossless enhancement layer to generate the scalable to lossless bitstream. Instead of having a separate MDCT filterbank, it re-uses the output of the IntMDCT for the AAC core layer for the purpose of complexity reduction. At the decoder, the *integer inverse MDCT* (IntIMDCT) filterbank is employed as the normative synthesis filterbank to generate the PCM audio output. In this section, we will describe some potential problems which may be caused by IntMDCT and IntIMDCT in a lossy coding scenario.

3.2.1 IntIMDCT in SLS Decoder

The purpose of IntIMDCT filterbank in an SLS decoder is to transform the spectrum domain signals into time domain outputs. If IntIMDCT is used instead of IMDCT at lossy bitrate, the output from IntIMDCT can be modelled with one from IMDCT added with the time domain rounding errors r_t.

With reference to the specifications of SLS [15], computation of the rounding errors r_t from the output of IntIMDCT is first adjusted by a gain factor. This serves as a comparison with the IMDCT counterpart. Specifically, the output from IntIMDCT is divided by a factor of $\sqrt{2N}$ (where N is the block length). The rounding errors are then computed as

$$r_t[n] = y[n] - \frac{y_{\text{int}}[n]}{\sqrt{2N}} = y[n] - y'[n], \tag{3.1}$$

where $y[n]$ is the time domain output values from the IMDCT, $y_{\text{int}}[n]$ and $y'[n]$ are

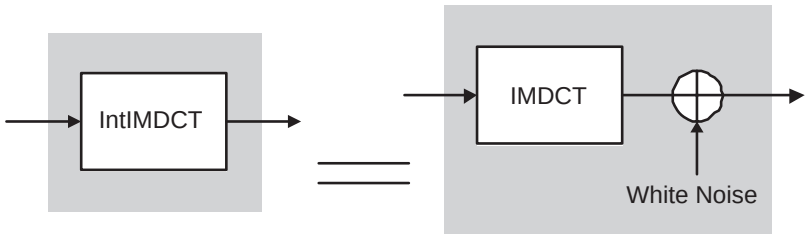


Figure 3.1: Equivalent block diagram of IntIMDCT and IMDCT with additive white noise.

the direct and normalized outputs from IntIMDCT respectively, and n denotes the sample number.

It was previously stated in [16] that the spectral domain error values of IntMDCT are not correlated with the spectral values and remain a constant order of magnitude in the entire spectral range. As expected, the time domain error signals $r_t[n]$ resembles a stationary white noise source. With this observation, it is reasonable to assume that the rounding errors of IntIMDCT can be modelled as an additive white noise being injected into the values of IMDCT, as shown in Figure 3.1.

If this white noise caused by IntIMDCT cannot be masked in the decoded audio signals, it may possibly introduce artifacts which affect the perceptual quality.

3.2.2 IntMDCT in SLS Encoder

For simplicity, the output from IntMDCT is directly used as the input of the AAC core layer in the encoding process. Thus the output from IntMDCT in the SLS encoder can be modelled as the one from MDCT with an addition of rounding errors as depicted in Figure 3.2. Similarly, the spectral domain rounding errors are computed as

$$r[k] = c[k] - c_{\text{int}}[k] \cdot \sqrt{2N} = c[k] - c'[k], \tag{3.2}$$

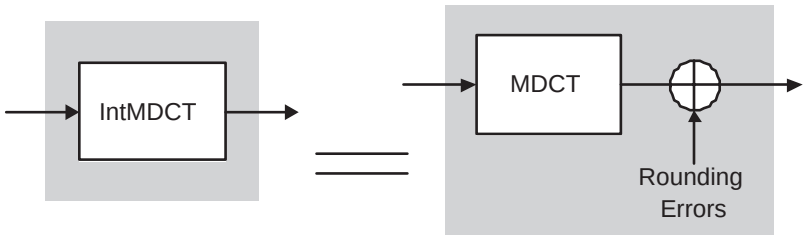
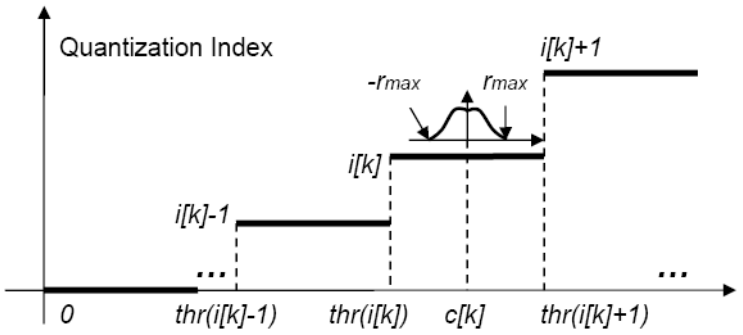


Figure 3.2: Equivalent block diagram of IntMDCT and MDCT with additive rounding errors.

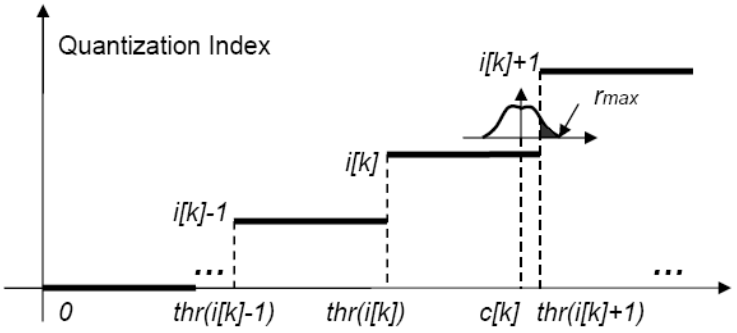
where $c[k]$ is the spectral domain output values from MDCT, $c_{\text{int}}[k]$ and $c'[k]$ are the direct and normalized IntMDCT outputs respectively, and k denotes the spectral coefficient number. Henceforth the normalized value will be used for the calculations.

Due to their small amplitudes, the rounding errors introduced by IntMDCT generally do not affect the results of the quantization process of AAC. However, *mis-quantization* (defined as the instance that AAC quantizer generates different outputs for MDCT input and IntMDCT input) may still occur when the quantization step size is small, or when the IntMDCT coefficients are close to the quantization threshold.

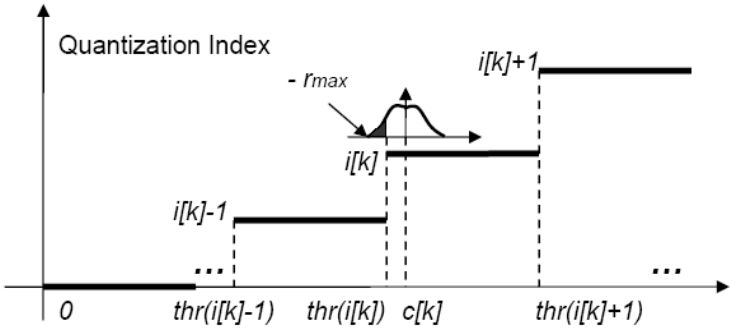
This is illustrated in Figure 3.3. Consider the quantization of a MDCT coefficient $c[k]$. The sub-axes in the graph indicate the distribution of rounding errors $r[k]$ and r_{max} is the maximum absolute value of the errors. It can be shown from simulation results to be presented later that both axes are symmetrical. Assuming that $c[k]$ will be quantized into quantization step $i[k]$, the errors added may cause the IntMDCT output $c'[k]$ to be shifted either to the left or right of $c[k]$. In Figure 3.3(a), $c[k]$ is far from the threshold of $i[k]$ and $i[k]+1$, which can be represented



(a)



(b)



(c)

Figure 3.3: Example to illustrate mis-quantization: (a) Scenario that mis-quantization will not occur. (b) Scenario that mis-quantization may occur (from $i[k]$ to $i[k]+1$). (c) Scenario that mis-quantization may occur (from $i[k]$ to $i[k]-1$).

as

$$\begin{cases} c[k] - r_{max} \geq thr(i[k]) \\ c[k] + r_{max} < thr(i[k]+1) \end{cases}. \tag{3.3}$$

Obviously the IntMDCT coefficient will be quantized into the same quantization step $i[k]$ as MDCT coefficient. Whereas in Figure 3.3(b), when $c[k]$ is approaching the threshold of $i[k]+1$, i.e.,

$$c[k] + r_{max} \geq thr(i[k]+1), \tag{3.4}$$

it is possible that IntMDCT coefficient $c'[k]$ will be quantized into $i[k]+1$ instead of the MDCT quantization output $i[k]$. Similarly for Figure 3.3(c), as $c[k]$ is approaching the threshold of quantization step $i[k]$, i.e.,

$$c[k] - r_{max} < thr(i[k]), \tag{3.5}$$

it is also likely that $c'[k]$ will be quantized into $i[k]-1$ instead of $i[k]$.

The instances depicted in Figures 3.3(b) and 3.3(c) are two types of mis-quantization. For general cases, as the amplitude of rounding errors are small, the differences between the quantization outputs for MDCT and IntMDCT inputs will only be one quantization step. Nevertheless such a difference, if any, will lead to errors in the reconstructed spectrum which may affect the perceptual quality of the decoded audio. As such, further exploration is necessary to justify the use of IntMDCT in lossy coding.

3.3 Analysis on Errors of IntMDCT in Lossy Coding of SLS

In the previous section, we have pointed out the potential artifacts which may be caused by IntMDCT and IntIMDCT filterbanks in lossy coding. The actual influences brought by these two filterbanks will be systematically analyzed in this section.

3.3.1 IntIMDCT in SLS Decoder

In order to evaluate the effect of the white noise introduced by IntIMDCT, the energy is first mapped to *sound pressure level* (SPL) in dB (under a reasonable playback sound level assumption) and then compared with the *absolute hearing thresholds* (AHT). If the noise energy is far below the AHT, the introduced noise will not be perceptible. Otherwise, if the noise energy happens to be higher than the AHT during some spectral range, or they are at similar level, it is necessary to perform further analysis such as consideration of the masking effects. In the following part of this subsection, we will describe the methods to map the noise energy into SPL. These methods will be used as the basis of the simulation described in the later part of this section.

3.3.1.1 Common Computation Method

The time domain noise energy can be transferred into SPL with reference to the method of determining the maximum allowable quantization noise energy in MPEG-1 psychoacoustic model I [11, 74]. It should be noted that the psychoacoustic model in MPEG-4 AAC [14] does not include the absolute SPL computation step.

Firstly, the incoming signal $r_t[n]$ are normalized according to the FFT length N and the number of bits per sample b using the relation

$$x[n] = \frac{r_t[n]}{N(2^{b-1})}. \tag{3.6}$$

The normalized input $x[n]$ are then segmented using a Hanning Window $w[n]$, which is defined as

$$w[n] = \frac{1}{2}\Big[1 - \cos\Big(\frac{2\pi n}{N}\Big)\Big]. \tag{3.7}$$

A *power spectral density* (PSD) estimate $P[k]$ are then obtained using a 2048-point FFT, i.e.,

$$P[k] = 10\log_{10}\left|\sum_{n=0}^{N-1} w[n]x[n]e^{-j(2\pi kn/N)}\right|^2, \tag{3.8}$$

$$0 \leq k < \frac{N}{2}.$$

To guarantee that a 4 kHz signal of 1-bit amplitude will be associated with an SPL near 0 dB where a full-scale sinusoid will be associated with an SPL near 90 dB, the PSD is further normalized by using a power normalization term P_0 which is fixed at 90.032 dB where

$$P'[k] = P[k] + P_0. \tag{3.9}$$

The noise energy for each sfb s is then computed from spectral lines using the sum

$$P_N[s] = 10\log_{10}\sum_{k=l[s]}^{u[s]} 10^{0.1P'[k]} \text{ (dB)} \tag{3.10}$$

where $l[s]$ and $u[s]$ are the lower and upper spectral line boundaries of the sfb s, respectively. Therefore, Eqn.(3.10) combines all of the energy from spectral components in each sfb into a single sum energy.

3.3.1.2 Special Case Method

The above steps show a common computation method to transform a noise energy into SPL. Based on the assumption that the rounding error to be analyzed is a stationary white noise, the SPL computation of the noise can be simplified by using the variance of the noise. The simplified formula is derived by the following steps: Firstly, the PSD of noise can be estimated as

$$P''[k] = 10\log_{10} E\left\{\left|\sum_{n=0}^{N-1}\{w[n]x[n]e^{-j(2\pi kn/N)}\}\right|^2\right\} + P_0, \tag{3.11}$$

where $E\{\cdot\}$ denotes the expectation operation. It can be derived that

$$E\left\{\left|\sum_{n=0}^{N-1}\{w[n]x[n]e^{-j(2\pi kn/N)}\}\right|^2\right\} = \frac{3}{2N4^b}E\left\{|r_t[n]|^2\right\}. \tag{3.12}$$

According to the property of white noise, the noise energy of each n is the variance of the noise,

$$E\left\{|r_t[n]|^2\right\} = \sigma^2. \tag{3.13}$$

By combining Eqns.(3.11) and (3.13), we have

$$P''[k] = 10\log_{10}\left(\frac{3\sigma^2}{2N4^b}\right) + P_0. \tag{3.14}$$

By substituting Eqn.(3.14) into Eqn.(3.10), the noise energy for each sfb s in SPL scale can be estimated by using the variance of the noise signal in the following expression,

$$P'_N[s] = 10\log_{10}\sum_{k=l[s]}^{u[s]} 10^{0.1P''[k]}\ (\text{dB}). \tag{3.15}$$

With the assumption that the noise is white, Eqn.(3.15) is the *special case method* for SPL noise energy computation. Instead of using the rounding error values as the input as in previous common computation method, the proposed method only requires the parameter which is the variance of the noise as the only input.

3.3.1.3 Simulation Results

All the 15 standard tracks [75] for MPEG lossless coding (as listed in TABLE 2.2) are used for testing. With the same MDCT encoded bitstreams, the rounding errors generated by two versions of IntIMDCT which include the one used in MPEG-4 SLS RM 1 (without noise shaping) and RM 5 (with noise shaping) are computed. Both RMs are selected to investigate the effect of the noise and noise shaping algorithm in the filterbank respectively.

Following the previously mentioned two methods, the per frame noise energy (P_N) for each sequence is first calculated using the *common computation method* [Eqn.(3.10)], and compared with the AHT. The noise energy (P'_N) is also computed using the *special case method* for white noise [Eqn.(3.15)].

The summary of the rounding errors and corresponding noise energy for IntIMDCT with and without noise shaping are shown in Tables 3.1 and 3.2, respectively.

The PSD and SPL of the noise for each critical band using both methods are then plotted in Figures 3.4(a) and 3.4(b), respectively. Among the 15 test sequences, avemaria.wav is chosen as an example to be plotted. It can be seen that the PSD of the noise is flat in the whole spectrum, which justifies the white noise assumption of the rounding errors. While the SPL values for the two RMs show different distributions in the low frequency region. This is due to the noise shaping procedure in RM 5.

Table 3.1: Summary of rounding errors and noise energy by IntIMDCT using RM 1.

Items (.wav)	Max absolute value of r_t	Mean absolute value of r_t	Standard deviation of r_t	Max noise energy computed by common computation method (dB)	Mean noise energy computed by common computation method (dB)	Mean noise energy computed by special case method (dB)
avemaria	2.64	0.55	0.68	-15.25	-28.36	-30.12
blackandtan	2.61	0.60	0.75	-14.17	-29.21	-29.53
broadway	2.55	0.61	0.76	-14.32	-28.95	-29.30
cherokee	2.65	0.60	0.78	-14.61	-28.74	-29.33
clarinet	2.86	0.63	0.79	-14.27	-29.16	-29.31
cymbal	2.60	0.59	0.74	-15.40	-28.95	-29.57
dcymbal	2.51	0.61	0.77	-14.53	-28.32	-29.17
etude	2.07	0.53	0.67	-15.21	-29.33	-30.72
flute	2.61	0.60	0.76	-15.06	-28.64	-29.58
fouronsix	2.36	0.61	0.77	-14.60	-28.28	-29.37
haffner	2.53	0.62	0.78	-14.25	-29.10	-29.34
mfv	2.03	0.51	0.64	-15.76	-29.79	-30.11
unfo	2.55	0.60	0.74	-14.98	-28.13	-29.12
waltz	2.44	0.60	0.74	-15.11	-27.98	-29.07
violin	2.92	0.63	0.79	-14.65	-28.07	-29.34
Average	2.53	0.59	0.74	-14.81	-28.73	-29.53

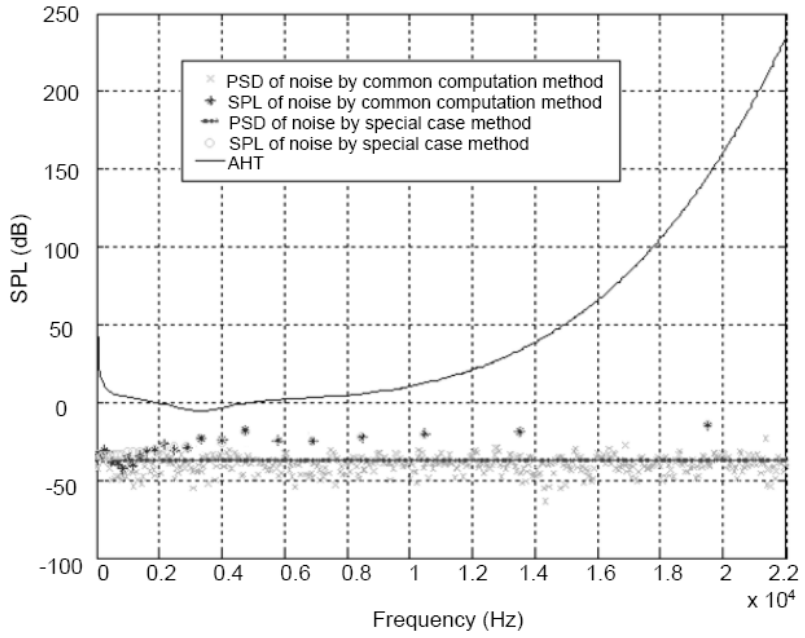
Table 3.2: Summary of rounding errors and noise energy by IntIMDCT using RM 5.

Items (.wav)	Max absolute value of r_t	Mean absolute value of r_t	Standard deviation of r_t	Max noise energy computed by common computation method (dB)	Mean noise energy computed by common computation method (dB)	Mean noise energy computed by special case method (dB)
avemaria	2.89	0.67	0.85	-14.32	-26.13	-27.32
blackandtan	4.37	0.72	0.91	-15.57	-26.76	-27.85
broadway	3.41	0.73	0.91	-15.33	-26.64	-28.19
cherokee	3.27	0.72	0.89	-14.35	-26.56	-27.54
clarinet	3.21	0.74	0.93	-15.37	-26.21	-27.43
cymbal	3.37	0.71	0.90	-15.64	-26.64	-28.11
dcymbal	3.16	0.76	0.96	-14.54	-26.10	-27.36
etude	4.16	0.65	0.92	-15.12	-25.56	-27.45
flute	3.80	0.71	0.90	-14.93	-25.23	-27.18
fouronsix	4.01	0.71	0.89	-15.12	-25.33	-28.10
haffner	4.00	0.74	0.93	-15.11	-25.23	-27.19
mfv	2.79	0.61	0.77	-16.36	-26.32	-28.46
unfo	3.37	0.73	0.91	-15.31	-25.13	-27.96
waltz	3.22	0.73	0.92	-14.76	-25.95	-27.23
violin	3.28	0.79	0.97	-15.24	-25.47	-27.14
Average	3.49	0.71	0.90	-15.14	-25.95	-27.63

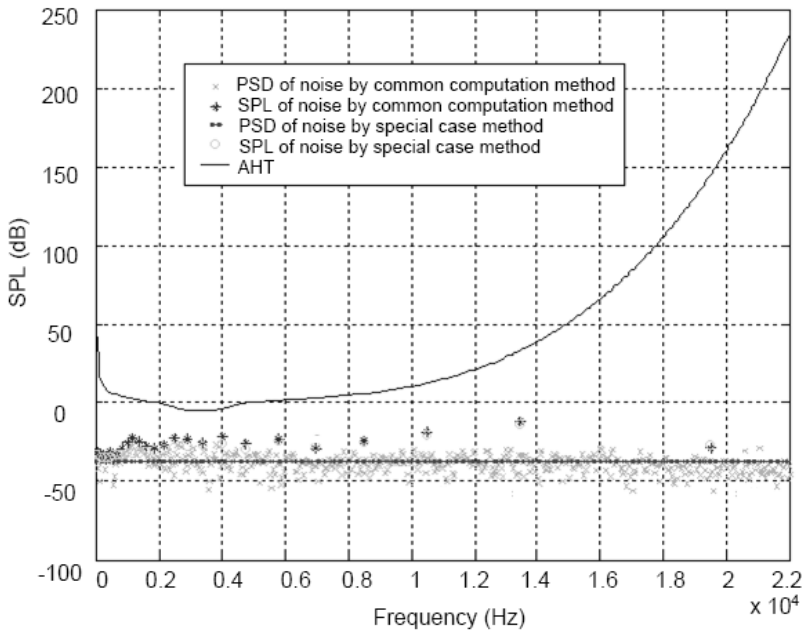
It is observed from the tables that both the amplitude and the variance of rounding errors is small. The minimum value of AHT among the whole spectrum is around -4.98 dB [11]. According to the simulation results, the maximum value of noise energy is in the range of -14 $\sim$ -17 dB. Therefore, the noise energy level of the rounding errors is far below the AHT. The results imply that under standard playback scenario, the noise introduced is not perceptible and hence it will not degrade the perceptual quality of the decoded audio comparatively. It can also be easily understood that the noise energy that is due to the rounding errors is independent of the coding bitrate and the amplitude of transformed coefficients. Also it is observed that the noise energy introduced by IntIMDCT with noise shaping is higher than the one without. This may due to the fact that noise shaping operation raises the signal energy at lower frequency band. In addition, the noise energy computed by *common computation method* and the energy by *special case method* perform a more accurate match for IntIMDCT without noise shaping. Although the results are slightly different for the two versions of IntIMDCT, the final conclusion is not affected. It can be concluded that the IMDCT and IntIMDCT filterbanks (versions include the one used in SLS RM 1 and SLS RM 5) are interchangeable in an SLS decoder at lossy bitrate as the energy of the corresponding errors are below the AHT. Therefore, IntMDCT can be used directly for a scalable audio coder.

3.3.2 IntMDCT in SLS Encoder

The purpose of the analysis in this subsection is twofold. Firstly, it is meaningful to establish a detailed analysis about the probability of IntMDCT introduced errors that will lead to mis-quantization (as mentioned in Section 3.2.2). The objective is to find out the parameters that will affect the probability of mis-quantization. Secondly, we would like to elaborate further the actual influence of mis-quantization.



(a)



(b)

Figure 3.4: Power spectral density and energy plots of the rounding noise from (a) RM 1 (b) RM 5 vs. absolute hearing threshold.

3.3.2.1 Mis-quantization Estimation

There are two cases in mis-quantization to be analyzed:

- Suppose that MDCT coefficient $c[k]$ is originally quantized into step $i[k]$. In [14] the range of quantization step is from 0 to 8191. The probability that $c[k]$ plus the error will be quantized into step $i[k]+1$ is indicated by

$$p_{i[k]\rightarrow i[k]+1} = p_{(c[k]+r[k]\geq thr(i[k]+1))}. \tag{3.16}$$

- Suppose that MDCT coefficient $c[k]$ is originally quantized into step $i[k]$. The probability that $c[k]$ plus the rounding error will be quantized into step $i[k]-1$ (since the error can be negative) is indicated by

$$p_{i[k]\rightarrow i[k]-1} = p_{(c[k]+r[k]<thr(i[k]))}. \tag{3.17}$$

Several issues which include the distribution characteristics of the MDCT coefficients and the rounding errors introduced by IntMDCT should be addressed first. Finally the model to calculate the probability is established.

a. Probability Distributions of the MDCT Coefficients

It has been previously reported in [76] that the Laplacian distribution provides a good approximation to IntMDCT coefficients in SLS. As it is known that the MDCT and IntMDCT coefficients only differ from each other by rounding errors, Laplacian pdf which is defined as

$$f(x) = \frac{1}{\sqrt{2\sigma^2}} e^{-\sqrt{2/\sigma^2}|x|} \tag{3.18}$$

is used to model the MDCT coefficients where σ is the standard deviation.

b. Probability Distributions of Rounding Errors

The IntMDCT coefficients are obtained by replacing the MDCT filterbank in the AAC codec with the IntMDCT filterbank. The effect of noise shaping is also considered. The example matching plots of the rounding errors generated by two versions of IntMDCT and different pdfs are shown in Figures 3.5(a) and 3.5(b), respectively. The rounding errors are computed over 2 frames of an audio sequence (48 kHz/16 bit, stereo). It is compared with the *generalized Gaussian* (GG) pdf [77] which is defined as

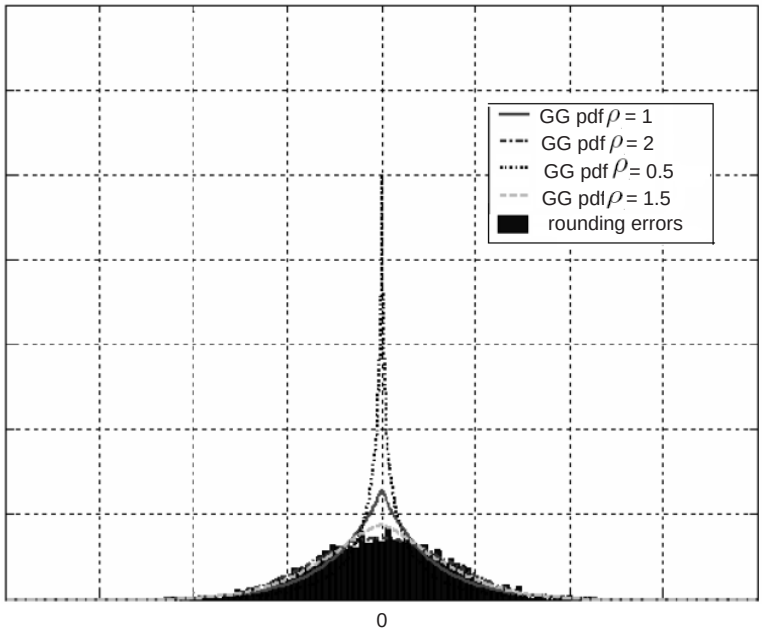
$$GG(m, \sigma, \rho) = K \exp\{-|\mu(x - m)|^{\rho}\}, \tag{3.19}$$

where $\mu = \left[\frac{\Gamma(3/\rho)}{\sigma^2\Gamma(1/\rho)}\right]^{\frac{1}{2}}$, $K = \frac{\rho\mu}{2\Gamma(1/\rho)}$, and the gamma function is defined as

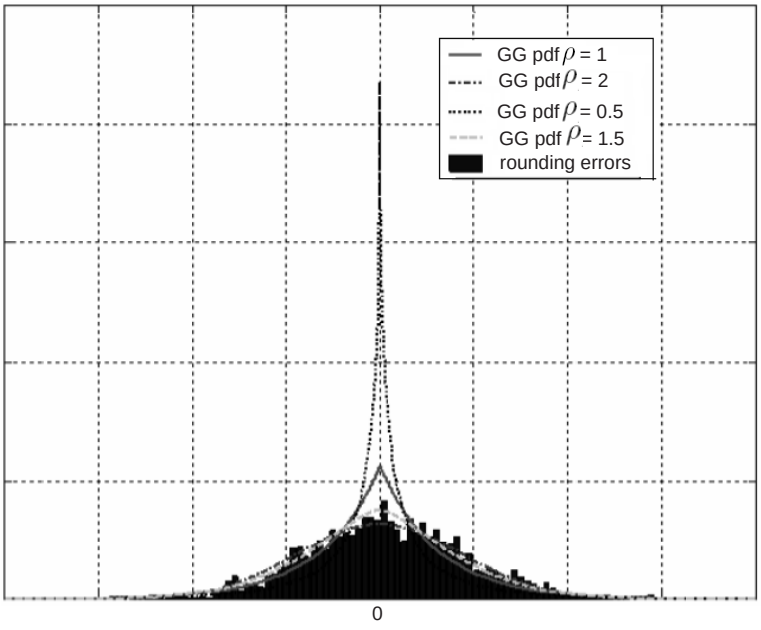
$$\Gamma(x) \triangleq \int_0^{\infty} t^{x-1} \exp(-t)dt, \tag{3.20}$$

where m and σ are the mean and the standard deviation of the random variable x respectively. GG pdf with parameter $\rho = 1$ is equivalent to the Laplacian pdf. It can be observed that the GG pdf with $\rho = 1.5$ provides a good approximation for the distribution of the rounding errors for both versions of IntMDCT.

To statistically identify the best GG model for the distribution of errors which is possibly varied for different audio signals, we further conduct the *Kolomogorov-Smirnov* (KS) goodness-of-fit test [78] to evaluate the deviation of the GG model from the error distributions of a number of audio sequences. In this test, a total of 15 stereo audio sequences same as those employed in Section 3.3.1 are used; the sampling rate for those sequences is 48 kHz; the window length of the IntMDCT

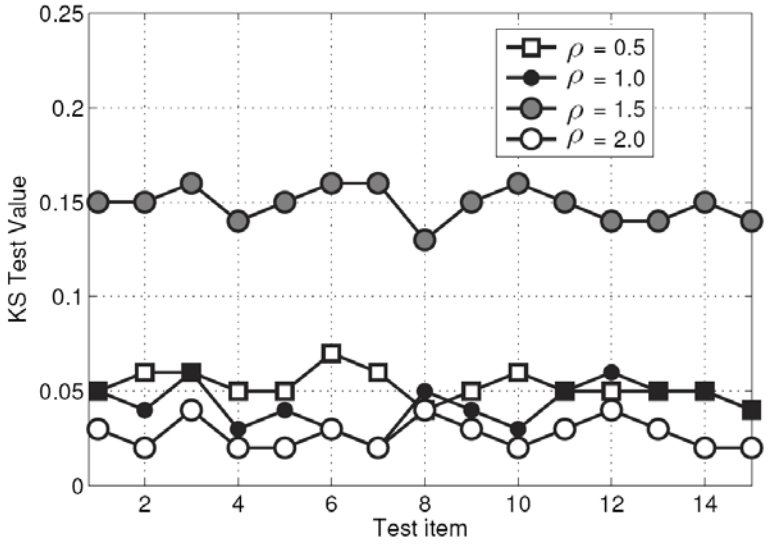


(a)

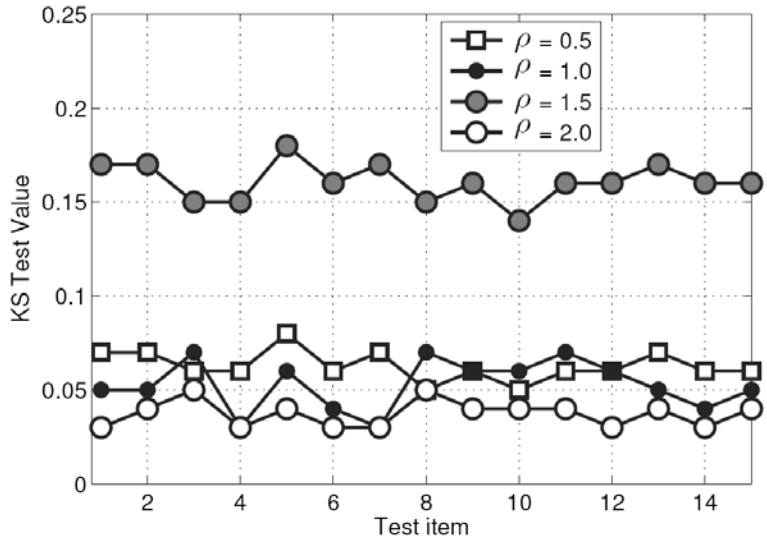


(b)

Figure 3.5: Matching of the histogram of rounding errors in (a) RM 1 (b) RM 5 and GG pdf.



(a)



(b)

Figure 3.6: KS goodness-of-fit test result for modelling IntMDCT rounding errors from (a) RM 1 (b) RM 5 using GG pdf with different value of ρ.

transform is 2048. For each audio sequence, the KS test was performed on the rounding errors. The KS test results are plotted in Figure 3.6, which shows that in most cases the GG pdf with $\rho = 1.5$ provides the most accurate model for the error distribution.

c. Mis-quantization Probability Model

Using the established pdf models for MDCT coefficients and rounding errors, we try to find out the parameters that affect the mis-quantization probability. For clarity, in this subsection, we use variable i equivalent as a particular value of $i[k]$, r as $r[k]$ and x as $c[k]$.

For a particular sfb, the probability $p_{0\to1}$ can be estimated as

$$\begin{aligned} p_{0\to1} = \; & p_{(X+r\geq thr(1))} \\ & \approx \int_{thr(1)-r_{max}}^{thr(1)} p_{(X+r\geq thr(1)|X=x)} p_{(X=x)} dx \\ & \approx \int_{thr(1)-r_{max}}^{thr(1)} g(x) f(x) dx, \end{aligned} \tag{3.21}$$

where $r_{max} = \max(|r|)$, and x denotes the absolute value of MDCT coefficients. $f(x)$ indicates the probability that a random variable X is located at MDCT coefficient position x. As mentioned above, it is modelled by Laplacian pdf as defined in Eqn.(3.18). The function $g(x)$ is the probability that $X + r > thr(1)$ at a certain position $X = x$. It can be computed as

$$g(x) = \int_{thr(1)-x}^{r_{max}} f_r(t) dt, \tag{3.22}$$

where $f_r(t)$ is the distribution function of rounding errors as defined in Eqn.(3.19). As the errors can be modelled by GG pdf with $\rho = 1.5$ and $m = 0$, by substituting these two values in $g(x)$ and by substituting Eqn.(3.18) into Eqn.(3.21), the probability that x will be quantized from step 0 to step 1 is

$$\begin{aligned} p_{0\to 1} \approx\ & 0.476 \int_{thr(1)-r_{max}}^{thr(1)} \left(\frac{1}{\sqrt{2\sigma_x^2}} e^{-\sqrt{2/\sigma_x^2}|x|} \right) \\ & \cdot \left(\int_{thr(1)-x}^{r_{max}} e^{-0.797\left|t\sqrt{\frac{1}{\sigma_x^2}}\right|^{1.5}} \sqrt{\frac{1}{\sigma_r^2}} dt \right) dx. \end{aligned} \tag{3.23}$$

Normalizing the situation from i to $i+1$, the probability that x with original quantized step i will be quantized into step $i+1$ with errors introduced is

$$\begin{aligned} p_{i\to i+1} \approx\ & p_{r(x+r\geq thr(i+1))} \\ =\ & 0.476 \int_{thr(i+1)-r_{max}}^{thr(i+1)} \left(\frac{1}{\sqrt{2\sigma_x^2}} e^{-\sqrt{2/\sigma_x^2}|x|} \right) \\ & \cdot \left(\int_{thr(i+1)-x}^{r_{max}} e^{-0.797\left|t\sqrt{\frac{1}{\sigma_x^2}}\right|^{1.5}} \sqrt{\frac{1}{\sigma_r^2}} dt \right) dx. \end{aligned} \tag{3.24}$$

The commonly used quantization function for [14] is

$$i = \left\lfloor \left(\frac{x}{\Delta}\right)^{\frac{3}{4}} + MAGIC\_NUMBER \right\rfloor \tag{3.25}$$

where $MAGIC\_NUMBER$ is fixed as 0.4054 and $\Delta = 2^{-0.25(scalefactor\text{-}common\_\ scalefactor)}$. We can obtain

$$\begin{aligned} thr(i+1) &\approx (i+1-MAGIC\_NUMBER)^{\frac{4}{3}}\Delta \\ &= (i+0.5946)^{\frac{4}{3}}\Delta. \end{aligned} \tag{3.26}$$

Substituting Eqn.(3.26) into Eqn.(3.24), the probability that the added errors will induce IntMDCT coefficient into step $i+1$ is

$$\begin{aligned} & p_{i\to i+1}(\Delta, \sigma_x, \sigma_r, i) \\ & \approx 0.476 \int_{(i+0.5946)^{\frac{4}{3}}\Delta - r_{max}}^{(i+0.5946)^{\frac{4}{3}}\Delta} \left(\frac{1}{\sqrt{2\sigma_x^2}} e^{-\sqrt{2/\sigma_x^2}|x|} \right) \\ & \cdot \left(\int_{(i+0.5946)^{\frac{4}{3}}\Delta - x}^{r_{max}} e^{-0.797\left|t\sqrt{\frac{1}{\sigma_x^2}}\right|^{1.5}} \sqrt{\frac{1}{\sigma_r^2}} dt \right) dx. \end{aligned} \tag{3.27}$$

Another possibility is that the coefficient which is originally quantized into step i is mis-quantized into step $i-1$. This case can be derived in the same way as $p_{i\to i+1}$.

From the above, the probability of the mis-quantization can be influenced by the following four parameters in the above derived relationships: the quantization step size, the standard deviations of the MDCT coefficients and rounding errors, and the corresponding quantization step index. These are further justified in Section 3.3.2.3.

3.3.2.2 Influence of Mis-quantization

To elaborate further the influence of mis-quantization, we consider two cases. First case, which is most common, refers to the scenario when the amplitude of the IntMDCT coefficients is much larger than that of the rounding errors, i.e.,

$$c'[k] \gg r[k] \tag{3.28}$$

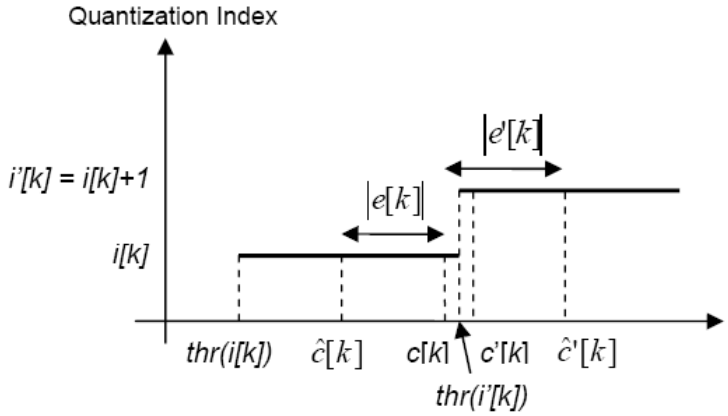


Figure 3.7: Illustration on Mis-quantization introduced under the First case.

and the Second case studies the scenario when the amplitude of the IntMDCT coefficients is equal to or smaller than that of the rounding errors,

$$c'[k] \leq r[k]. \tag{3.29}$$

This may happen when the input audio has a very low energy. In the *First case*, the situation in Figure 3.3(b) is further analyzed in Figure 3.7. When the AAC quantizer generates different outputs for MDCT inputs and IntMDCT inputs, the quantization noise for the IntMDCT filterbank and the MDCT filterbank are

$$e'^2[k] = (c[k] - \hat{c}'[k])^2 \tag{3.30}$$

and

$$e^2[k] = (c[k] - \hat{c}[k])^2 \tag{3.31}$$

respectively. Here $\hat{c}'[k]$ and $\hat{c}[k]$ are the reconstructed values for quantized index $i'[k]$ and $i[k]$. $i[k]$ and $i'[k]$ are the quantization index for MDCT coefficient $c[k]$ and IntMDCT coefficient $c'[k]$, respectively. $i'[k]$ can be different from $i[k]$ by ± 1,

i.e.,

$$i'[k] = i[k] \pm 1, \tag{3.32}$$

where Figure 3.7 only depicts the case that $i'[k] = i[k] + 1$. Since the amplitude of the rounding error $r[k] = c[k] - c'[k]$ is very small compared to that of $c'[k]$, it is clear that the amplitudes of $e[k]$ and $e'[k]$ are very similar. This implies that outputs of IntMDCT and MDCT will have very similar quantization noise. In other words, compared with using MDCT, the usage of IntMDCT in the AAC encoder will not introduce additional errors with significant energy in the reconstructed MDCT spectrum. To further confirm this observation, the quantization noise energy from the encoder with MDCT is compared with another encoder that uses IntMDCT. The test results are shown in next subsection.

In the Second case, the amplitude of the IntMDCT coefficients is equal to or smaller than that of the rounding errors. This implies that pure noise energy of rounding errors is introduced. It can be seen from TABLE 3.3 in next section that the mean absolute value and standard deviation of the errors introduced by IntMDCT is almost the same as those in the time domain. Thus they have almost the same noise energy level. Although the quantized outputs for MDCT coefficients and IntMDCT coefficients can be very different, it is already shown in Section 3.3.1 that the noise energy of rounding errors is far below the AHT. It can be thus concluded that such a difference will not be perceptible to the human auditory system under this case.

3.3.2.3 Simulation Results

The simulation results for the rounding errors introduced by IntMDCT in SLS encoder side and the influence to the quantization are given in this section. The testing sequences are the same as those used in Section 3.3.1. For each sequence, the

Table 3.3: Summary of rounding errors introduced by IntMDCT.

Name of sequences (.wav)	IntMDCT using RM 1			IntMDCT using RM 5		
	Max absolute value of $r[k]$	Mean absolute value of $r[k]$	Standard deviation of $r[k]$	Max absolute value of $r[k]$	Mean absolute value of $r[k]$	Standard deviation of $r[k]$
avemaria	4.71	0.56	0.71	7.71	0.68	0.87
blackandtan	5.96	0.56	0.71	5.96	0.69	0.87
broadway	8.23	0.58	0.76	9.23	0.71	0.92
cherokee	5.29	0.57	0.72	3.92	0.68	0.86
clarinet	17.19	0.58	0.77	16.19	0.70	0.91
cymbal	4.02	0.58	0.73	5.55	0.70	0.89
dcymbal	2.73	0.56	0.69	3.23	0.67	0.85
etude	8.82	0.59	0.80	9.82	0.71	0.94
flute	2.85	0.56	0.70	4.30	0.68	0.86
fouronsix	3.21	0.57	0.71	4.39	0.69	0.87
haffner	8.23	0.59	0.78	8.23	0.71	0.92
mfv	11.13	0.59	0.81	11.59	0.71	0.95
unfo	2.87	0.56	0.70	4.02	0.68	0.86
waltz	2.87	0.57	0.72	3.68	0.67	0.85
violin	4.95	0.56	0.70	5.95	0.68	0.86
Average	6.20	0.57	0.73	6.92	0.69	0.89

maximum, mean and the standard deviation of the rounding errors from IntMDCT for RM 1 and RM 5 are computed, as shown in Table 3.3. It can be observed that the maximum values of rounding errors introduced by IntMDCT are larger than those introduced by IntIMDCT (Tables 3.1 and 3.2). This can be explained by the procedure based on which the rounding errors are computed. Comparing Eqn.(3.2) with Eqn.(3.1), the IntIMDCT coefficients are divided by a gain factor of $\sqrt{2N}$, and IntMDCT coefficients are multiplied by that gain factor in order to match the energy level of the MDCT and IMDCT, respectively. Thus, it is reasonable that some rounding errors introduced by IntMDCT may have larger amplitude than those introduced by IntIMDCT. Nevertheless, as long as the mean absolute value and standard deviation of the errors introduced by IntMDCT is almost the same as those introduced by IntIMDCT, it can be concluded that they have almost the same noise energy level.

The probability functions of mis-quantization are useful for estimating the probability of mis-quantization. Using the AAC bitrates of 96, 128 and 192kbps, the probability related parameters from each test items are collected. With these parameters the theoretical probability of mis-quantization (a sum probability of both cases including $i \rightarrow i+1$ and $i \rightarrow i-1$) is then computed using the two equations and compared with the experimental sum probability. Specifically, the experimental probabilities are computed by dividing the total number of mis-quantized coefficients by the total number of transformed coefficients for all the test sequences. The theoretical values are calculated as the sum probability for all the quantization indexes and all the frames for each test sequence. The results from RM 1 and RM 5 are shown in Figure 3.8. It is observed that for most of the test sequences the probability of mis-quantization increases when the available AAC bitrate increases,

which is caused by the smaller distance between the thresholds of each quantization index. It can be seen that the theoretical probability model has a good match with the experimental data.

The effects of mis-quantization is further evaluated. For each test item, the relative quantization noise difference $D_r[s]$ for each sfb s which is defined as

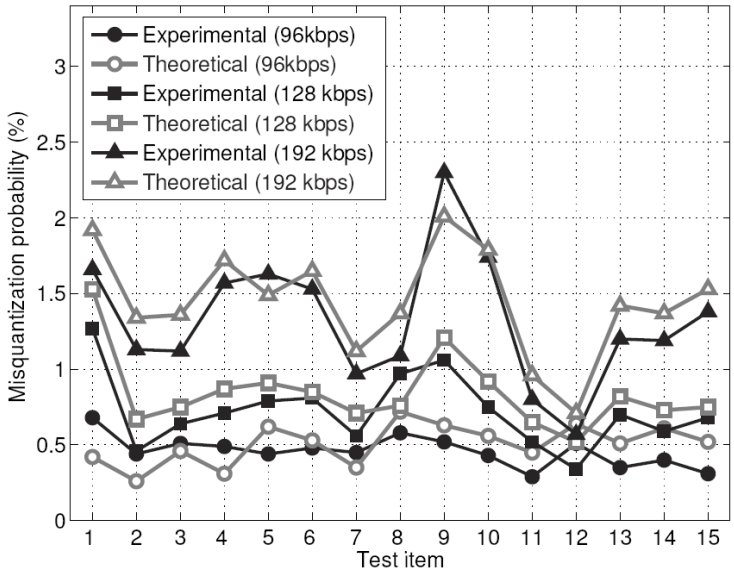
$$D_r[s] = \frac{|E_{\mathrm{MDCT}}[s] - E_{\mathrm{IntMDCT}}[s]|}{E_{\mathrm{MDCT}}[s]} \tag{3.33}$$

is computed, where $E_{\mathrm{MDCT}}[s]$ and $E_{\mathrm{IntMDCT}}[s]$ denote the quantization noise in terms of dB for sfb s with MDCT and IntMDCT input respectively. The testing conditions are: AAC encoder bitrate = 64 kbps, number of sfb per channel per frame = 49. Figures 3.9(a) and 3.9(b) show plots of the relative quantization noise difference for 10 frames of avemaria.wav with two versions of IntMDCT respectively. As expected, it is observed the relative quantization noise difference between the two systems that using MDCT and IntMDCT is at most around 10%. The results are similar for all the testing sequences at normal operating bitrates including 96, 128 and 256kbps. These results verify the theoretical analysis presented in the previous subsection.

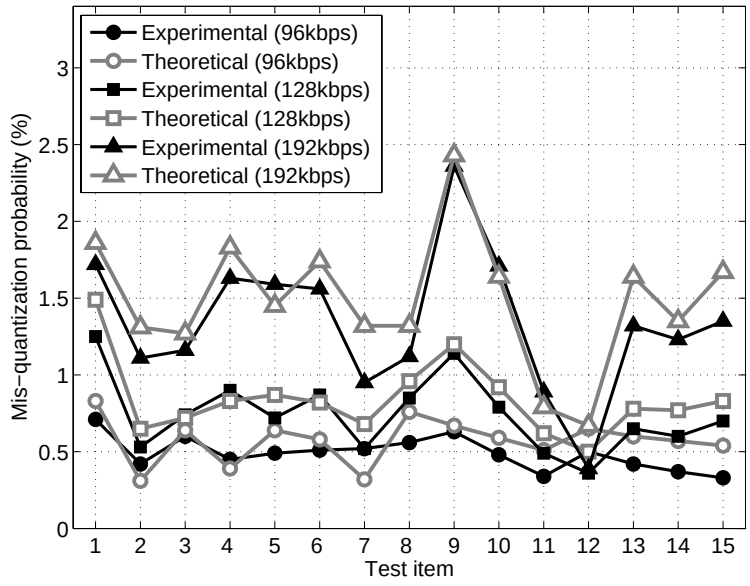
Based on these results, we can conclude that the rounding errors introduced by IntMDCT will not degrade the perceptual quality of the decoded audio as the energy of the rounding errors are below the AHT.

3.4 Subjective Listening Test Results

The analysis results from the previous Section is further verified by subjective listening test. The coding performance of normal MPEG-4 AAC structure with MDCT and IMDCT filterbanks is compared with that of the MPEG-4 AAC with

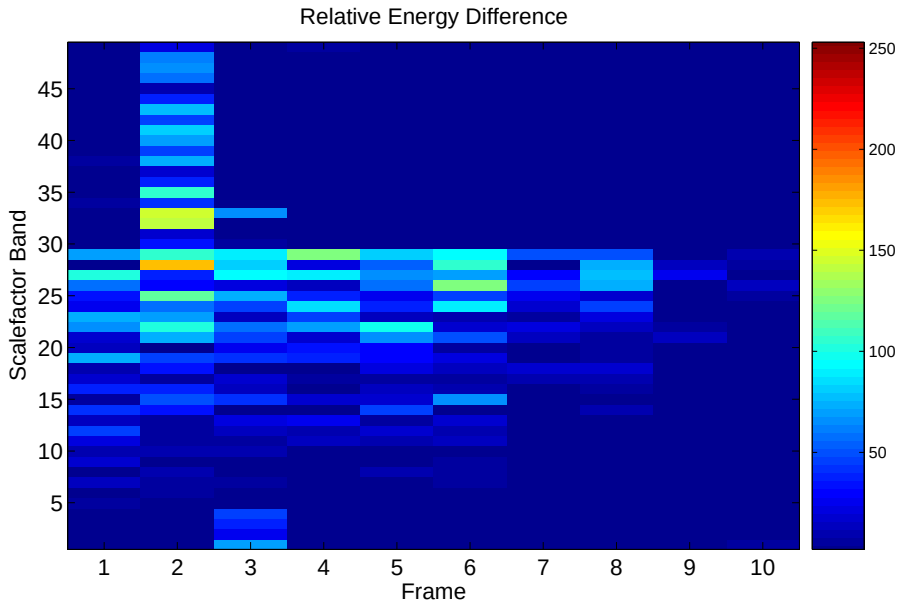


(a)

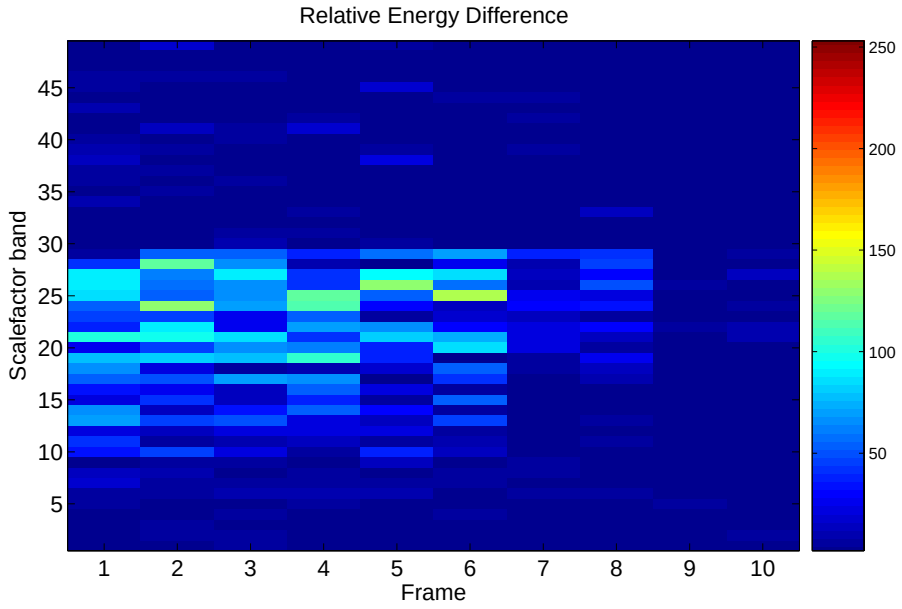


(b)

Figure 3.8: Experimental and theoretical probability of mis-quantization in 15 test items using (a) RM 1 (b) RM 5. Please refer to TABLE I for item name.

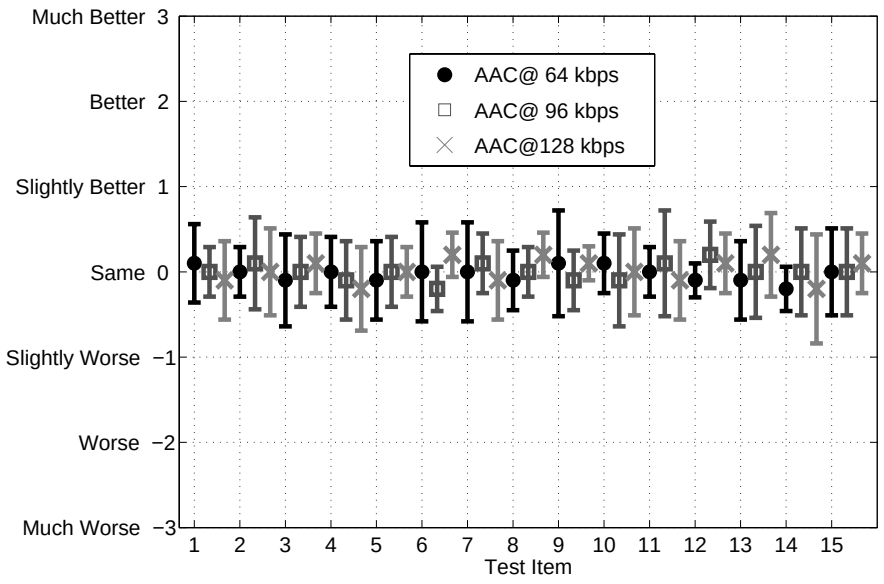


(a)

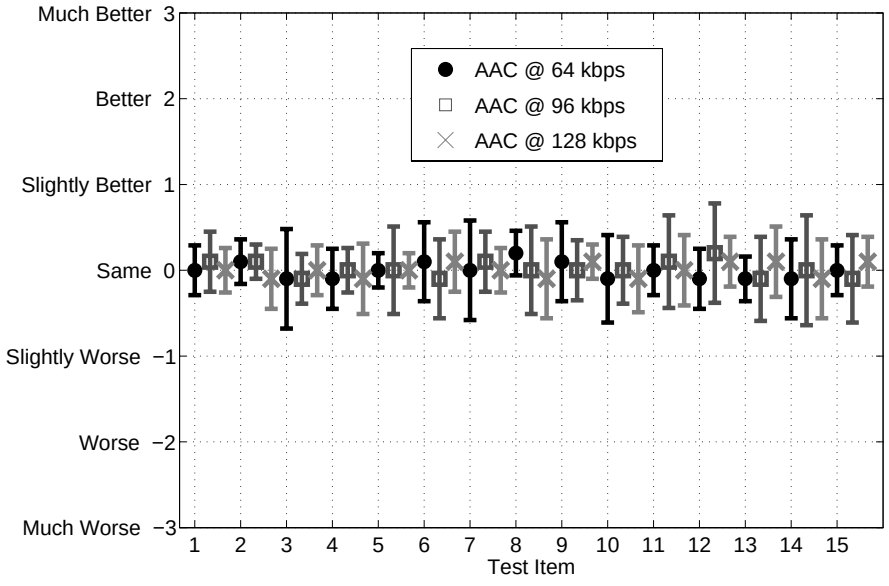


(b)

Figure 3.9: Quantization Noise Energy Comparisons between MDCT and Int-MDCT inputs in (a) RM 1 (b) RM 5.



(a)



(b)

Figure 3.10: Subjective listening test results using ITU-R BS.1284 seven-grade comparison. For each test the first system is the normal AAC stucture with MDCT and the second system is the AAC structure with IntMDCT filterbank in (a) RM 1 (b) RM 5. Score 3 indicates that the first system is much better than the second system, and so on.

IntMDCT and IntIMDCT filterbanks in RM 1 and RM 5. Three groups of tests with the AAC bitrates of 64, 96 and 128 kbps stereo for each RM are performed. The test sequences are the same as those used in Section 3.3.1. The ITU-R BS.1284 [79] test method with seven-grade comparison is used.

It can be observed from Figure 3.10 that the subjective performance of two structures show no apparent differences. This justifies the simple structure of using IntMDCT filterbank only in scalable audio codec.

3.5 Conclusion

Scalable to lossless coding is one of the latest research trends in audio coding area. Owing to its scalable capability, it is a well suited technology for a combination of different audio compression demands. To enable efficient lossless coding, IntMDCT was adopted in MPEG-4 SLS. Notwithstanding the fact that MDCT is generally employed for lossy coding, SLS uses IntMDCT as the only filterbank for both lossy and lossless coding scenarios.

With concerns about the potential problems which may be caused by IntMDCT under perceptual audio coding, this chapter conducts analysis and testings on the influence of the rounding errors introduced by IntMDCT under lossy operation. Based on the results, it is found that the rounding errors of IntMDCT will not degrade the perceptual quality of decoded audio under standard playback circumstances as the energy of the rounding errors are below the AHT. It is therefore concluded that MDCT and IntMDCT filterbanks are interchangeable in lossy coding scenario. This conclusion proves the validity of using only IntMDCT in scalable lossless audio coder.

Chapter 4

Perceptually Enhanced Bit-Plane Coding

4.1 Background

For cases where full scalability or near-full scalability in audio coding are desired, the bitrate occupied by the AAC core in an SLS codec may be zero (i.e., non-core mode of SLS) or very low. Due to the lack of perceptual coding, the spectral shape of the residual signal for bit-plane coding roughly follows the shape of the original signal spectrum, which is far from optimum for bit-plane coding in sequential order. This will directly result in non-optimal perceptual quality of output audio at intermediate bitrates. To improve on this, it was mentioned in [80] that psychoacoustic information such as the *just noticeable distortion* (JND) can be applied in the bit-plane coding process for a perceptually more efficient enhancement coding performance. This approach, with a certain degree of quality improvement, comes with a considerable amount of side information which may cause the degradation of coding efficiency at low bitrates.

As the perceptual quality of fully scalable codec such as non-core SLS is still far inferior to the quality that can be achieved by a perceptual audio coder at the same bitrate, the prioritized bit-plane coding is proposed in this Chapter. The

main purpose is to efficiently improve the quality of fully scalable audio with the least possible extra side information and modification to the standardized codec. Therefore, the proposed method is different from existing approaches in that it involves a combination of two features. First, priorities are assigned according to the energy distribution and thus eliminating any attachment of psychoacoustic side information. Secondly, these priorities are assigned to the bit-planes of several frequency regions or groups of sfbs only. These considerations are important and will be elaborated below.

When the priorities are assigned to the frequency regions instead of a single bit (or coefficient, sfb) without computation or attachment of psychoacoustic information, a system can indeed be very simple in both its structure and computation. In fact, in its simplest version, only as little as 1 bit per frame of side information is needed. Yet, this priority assignment scheme based on pure energy distribution is shown to be perceptually efficient. The test results indicate that even with the simplest version of the proposed system, a considerable amount of perceptual improvement can be achieved for the non-core mode of SLS or SLS with low core bitrate in a wide range of lossy bitrates. It should be noted that this low-complexity enhancement algorithm is not to replace the perceptual coder; rather it is to improve the performance of fully or near-fully scalable coders where the bitrate for the perceptual core is limited.

The rest of this chapter is organized as follows. Section 2 gives the basic idea that inspires the algorithm. The proposed frequency region based prioritized bit-plane coding is formulated and solved with the simplified solution in Section 3. These are followed by extensive testing results from the implementation of the enhanced SLS structure with the proposed system.

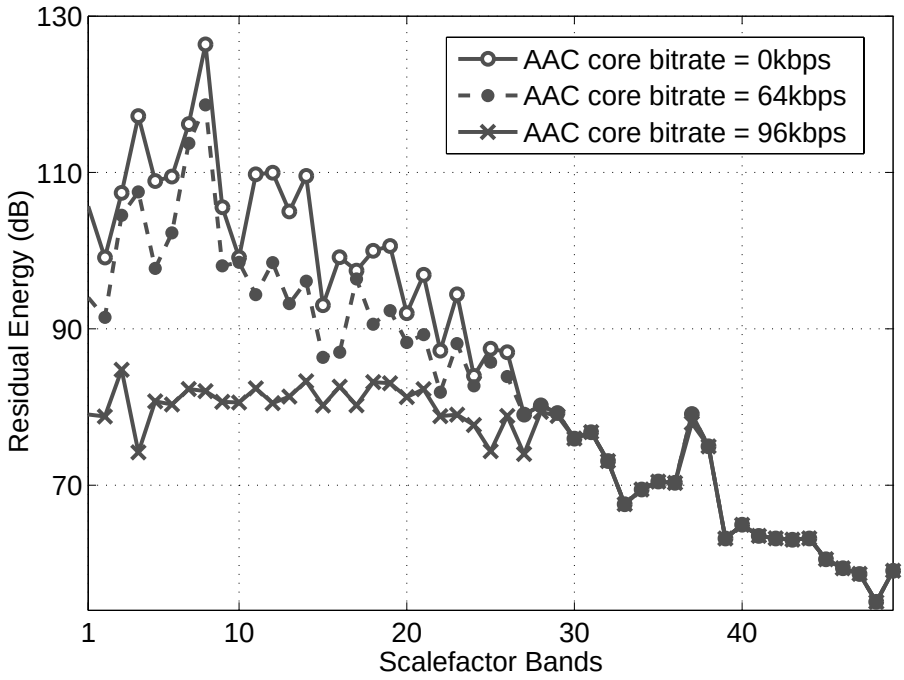


Figure 4.1: Residual energy spectrum of SLS for AAC core bitrates at 0, 64 and 96kbps.

4.2 Basic Idea

Figure 4.1 shows an example distribution of the residual energy between the original signal and AAC quantized signal in the frequency domain. The data are created from the 600th frame of avemaria.wav with a total of 49 sfbs per frame. The plot shows the residual energy at the AAC core bitrates of 0, 64 and 96kbps, respectively. In particular, the residual energy with 0 core bitrate is actually the energy of the original signal itself. Compared with the case at 96kbps, the residual energy at the other two core bitrates are apparently more concentrated in the low frequency region of the spectrum. Figure 4.2 further plots the signal energy against the *noise to mask ratio* (NMR) of the frames numbered 100, 200, 300, 400 and 500 of the avemaria.wav with a bit-plane coding bitrate of 64kbps in the

non-core mode. The noise $N[s]$, $s = \{0, 1, ..., 48\}$ for NMR is computed as

$$N[s] = \sum_{k=O[s]}^{O[s+1]-1} \left(c[k] - b[k]\right)^2, \quad \forall\ s \tag{4.1}$$

where $O[s]$ denotes the offset starting coefficient of sfb s, and $b[k]$ is the coefficient reconstructed by CBAC bit-plane coding. In addition, in SLS RM encoder the psychoacoustic mask $M[s]$ in NMR is computed as

$$M[s] = \begin{cases} \frac{E[s]}{\mathrm{SMR}[s]}, & E[s] > 70\mathrm{dB} \text{ and } \mathrm{SMR}[s] > 1 \\ E[s], & E[s] > 70\mathrm{dB} \text{ and } \mathrm{SMR}[s] \leq 1 \\ E[s] \times 1.1, & E[s] \leq 70\mathrm{dB} \end{cases} \tag{4.2}$$

where $\mathrm{SMR}[s]$ denotes the *signal to mask ratio* computed from the psychoacoustic model of the AAC for sfb s and the signal energy is computed as

$$E[s] = \sum_{k=O[s]}^{O[s+1]-1} c^2[k]. \tag{4.3}$$

Note that Eqn.(4.2) is the default mask implementation in the MPEG-4 AAC. It is observed that by using only sequential bit-plane coding, higher value NMR dominates when the signal energy is above a certain threshold. Together, Figures 4.1 and 4.2 clearly explain why sequential bit-plane coding works inefficiently in low core bitrate scenarios of SLS. In order to achieve the optimized rate-distortion, higher priorities should be assigned to the bit-planes of the frequency regions with a higher level residual energy. The prioritized bit-plane coding algorithm proposed in the current work is based on the above observations.

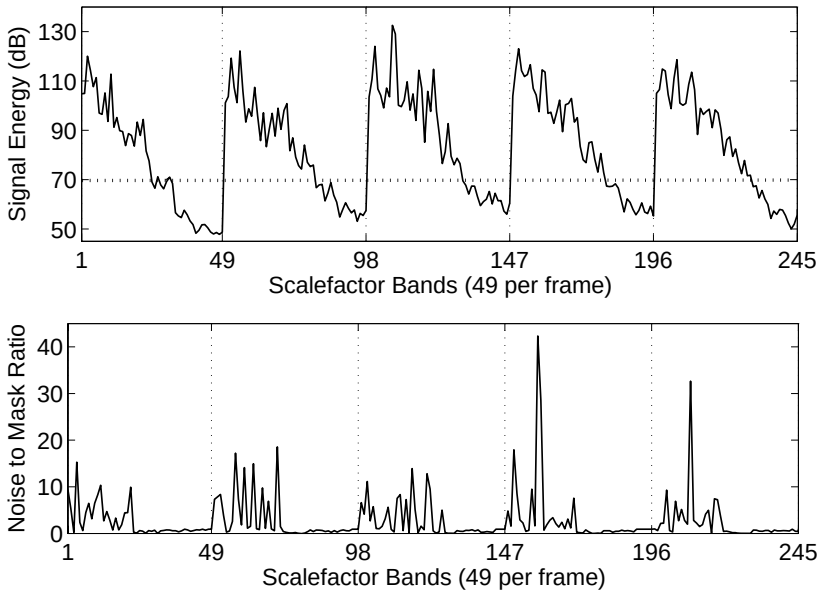


Figure 4.2: Signal energy versus the noise to mask ratio for 5 frames of ave-maria.wav by using SLS non-core coding.

4.3 Frequency Region based Prioritized Bit-plane Coding

The idea of frequency region based prioritized bit-plane coding is, as the name suggests, to divide the entire frequency spectrum into several regions and assigning these regions with priorities according to their respective energy levels. In this section, the basic algorithm definitions are given first. It is followed by the formulation of the optimization problem. Finally, a series of simplification procedures and solutions are discussed.

4.3.1 Basic Algorithm

Figure 4.3 depicts the spectrum of a particular frame i, $i = \{0, 1, ...I-1\}$ where I is the total number of frames. The entire spectrum of frame i with a total of S sfbs is divided into N_r^i regions where $N_r^i \in \mathbb{N}$ and $2 \leq N_r^i \leq S$. Each region is

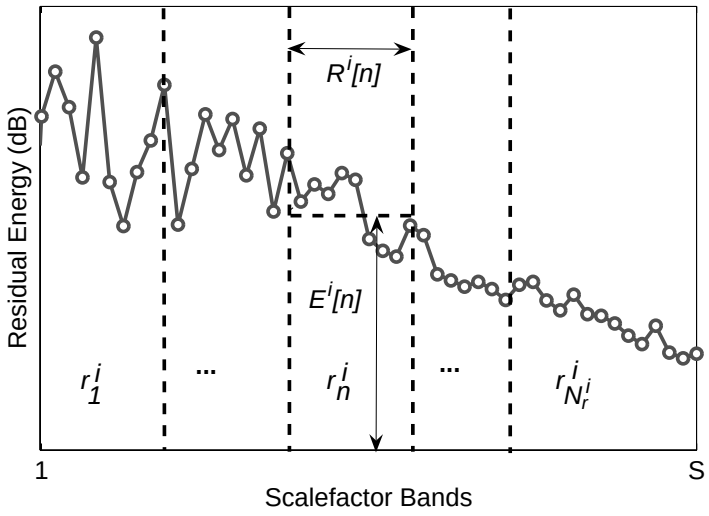


Figure 4.3: Division of regions in one frame.

denoted by r_n^i, $n = \{1, ..., N_r^i\}$. And r_n^i includes a set of sfbs, $R^i[n]$, with

$$R^i[n] = \big\{\Theta(r_n^i), \Theta(r_n^i) + 1, ..., \Theta(r_{n+1}^i) - 1\big\}. \tag{4.4}$$

Here $\Theta(r_n^i)$ is the starting sfb of region r_n^i. The set of starting sfbs for each region, Γ^i, is thus defined as

$$\Gamma^i = \big\{\Theta(r_2^i), \Theta(r_3^i), ..., \Theta(r_{N_r^i}^i)\big\}. \tag{4.5}$$

The corresponding coding priority of r_n^i is denoted by $P^i[n]$, $P^i[n] \in \mathbb{N}$ and $P^i[n] \geq 1$. The lowest priority is represented by $P^i[n] = 1$. We further define $E^i[n]$ as the average energy of r_n^i and it is computed as

$$E^i[n] = \frac{1}{\Theta(r_{n+1}^i) - \Theta(r_n^i)} \sum_{s=\Theta(r_n^i)}^{\Theta(r_{n+1}^i)-1} \Big(10 \log E[s]\Big) \tag{4.6}$$

where $E[s]$ is the residual energy of sfb s. Let $\Delta E^i(n,m)$ be the energy difference ratio between r_n^i and r_m^i,

$$\Delta E^i(n,m) = \frac{E^i[n] - E^i[m]}{E^i[m]}, \ \forall \ m > n \tag{4.7}$$

$$m = n+1, n+2, ..., N_r^i.$$

The priorities are assigned according to the average energy as

$$\begin{cases} P^i[n] > P^i[m], & \text{if } \Delta E^i(n,m) > \delta_1^i \\ P^i[n] = P^i[m], & \text{if } \Delta E^i(n,m) \leq \delta_1^i \end{cases} \tag{4.8}$$

where $\delta_1^i \in \mathbb{R}$ is the minimum energy difference threshold for frame i. It is assumed that the priority of the lower frequency region is always at least equal to that of the higher frequency region.

With the above definitions and rules, the regions of frame i can be sorted as a list with descending orders of priorities. The coding order of the bit-planes for each region can be thus determined accordingly. Note that the priorities of coding are only assigned to the top J ($J \in \mathbb{N}$ and $J \geq 2$) bit-planes while the coding order of the subsequent bit-planes remains the same. For SLS, the top J bit-planes are those non-lazy bit-planes. All the lazy bit-planes of each region follow the same coding orders (as in Figure 2.6) after all the non-lazy bit-planes are coded. The reason behind this is two-fold. Firstly, having more layers of prioritization involves higher complexity along with unwanted side information that can be limited with this method. Secondly, it is redundant to assign priorities to lazy bit-planes because for most of audio sequences, near transparent quality can already be achieved with a small percentage of lazy coding. This is evident in Figure 4.4 which shows the percentage of frames that enter lazy mode coding for the 15 standard test sequences

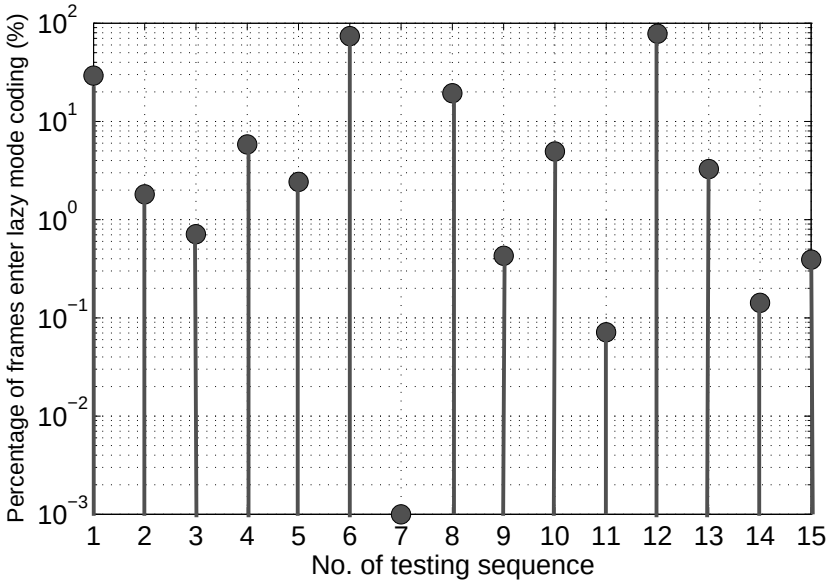


Figure 4.4: Percentage of frames entering lazy-mode coding at non-core bitrate of 384kbps (See TABLE 2.2 for the full list of test sequences).

at 384kbps (non-core). It is observed that 13 out of 15 items have less than 30% of frames in lazy mode coding at 384kbps. While at this coding bitrate, quality is almost transparent (refer to Figure 2.7).

The specific coding order of the bit-planes for each region is dependent on the level of $\Delta E^i(n,m)$. Suppose that the number of non-lazy bit-planes in SLS is fixed at J for a particular coding configuration. The MSB bit-plane of each sfb s is indicated by b_J^s, followed by subsequent bit-planes $b_{J-1}^s, ..., b_j^s, ...b_1^s$, with $j \in \mathbb{N}$ and $1 \leq j \leq J$. Consider a set of energy difference threshold values, $\boldsymbol{\delta^i} = \{\delta_1^i, \delta_2^i, ...\delta_z^i, ...\delta_{Z^i}^i\}$, $z \in \mathbb{N}$ and Z^i with $1 \leq Z^i < J$ is the level of priorities. In particular, δ_1^i is the minimum threshold value which is used in Eqn.(4.8). In addition,

$$\delta_1^i < ... < \delta_z^i < ... < \delta_{Z^i}^i \tag{4.9}$$

Let s_n and s_m denote the sfbs in r_n^i and r_m^i respectively, where

$$\Theta(r_n^i) \leq s_n \leq \Theta(r_{n+1}^i) - 1 \tag{4.10}$$

$$\Theta(r_m^i) \leq s_m \leq \Theta(r_{m+1}^i) - 1. \tag{4.11}$$

If $P^i[n] = P^i[m]$, the priority level difference $\tau^i(n,m)$ with $\tau^i(n,m) \in \mathbb{Z}$ is defined as 0. The corresponding coding sequence follows Rule 1 which is actually the sequential order coding as depicted in Figure 2.6.

> **Rule 1**: *the bit-plane $b_j^{s_m}$ will only be coded when the coding of bit-plane $b_j^{s_n}$ in r_n^i is completed. The bit-planes from $b_j^{s_n}$ in region r_n^i will only be coded when coding of bit-plane $b_{j+1}^{s_m}$ is completed.*

Otherwise, if

$$P^i[n] > P^i[m] \text{ and } \delta_z^i < \Delta E^i(n,m) \leq \delta_{z+1}^i, \tag{4.12}$$

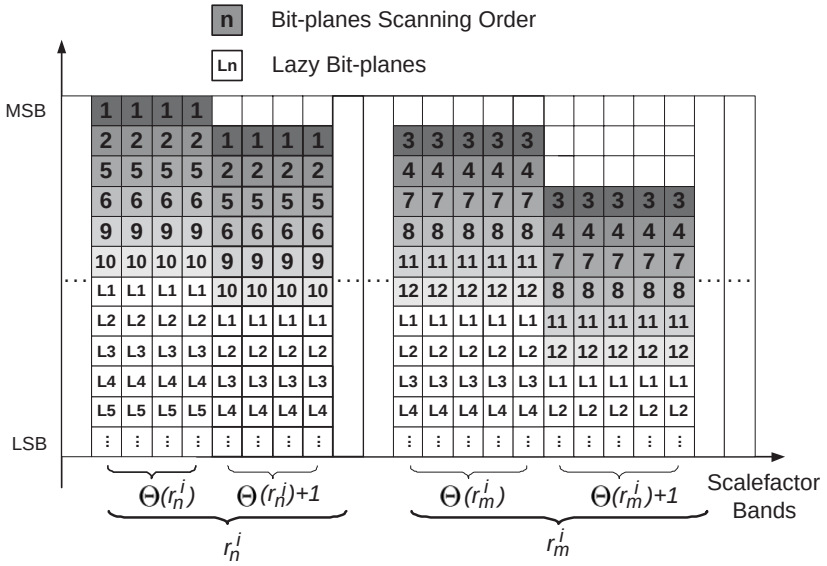
it is defined that $\tau^i(n,m) = z+1$ and the coding will proceed as follows:

> **Rule 2**: *the bit-planes from $b_j^{s_m}$ to $b_{(j-z)}^{s_m}$ in region r_m^i will only be coded when the coding of bit-planes from $b_j^{s_n}$ to $b_{(j-z)}^{s_n}$ in r_n^i is completed. The bit-planes from $b_j^{s_n}$ to $b_{(j-z)}^{s_n}$ in region r_n^i will only be coded when the coding of bit-planes from $b_{j+z+1}^{s_m}$ to $b_{(j+1)}^{s_m}$ in r_m^i is completed.*

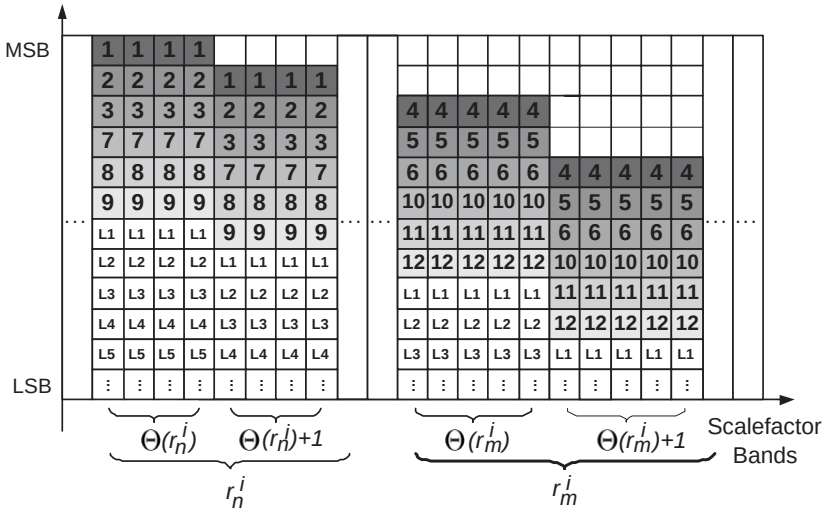
Rule 2 is demonstrated by an example as shown in Figure 4.5, with $J = 6$ and $Z^i = 3$.

4.3.2 Parameter Optimization

With the basic algorithm described in the previous subsection, the relevant parameters should now be chosen to optimize the system's performance. For a particular



(a)



(b)

Figure 4.5: Bit-plane coding order for regions r_n^i and r_m^i with (a) $\tau^i(n,m) = 2$ (b) $\tau^i(n,m) = 3$.

frame i, the following multiple objective optimization problem can be formulated in this way:

$$\arg \min_{N_r^i, \Gamma^i, \boldsymbol{\delta^i}} \left\{ D\big(N_r^i, \Gamma^i, \boldsymbol{\delta^i}\big) \right\} \tag{4.13}$$

where $D(\bullet)$ is the distortion function. For simplification, a few assumptions are made here. First, the available bitrate is not considered as an influencing parameter. Second, the number of regions and priority thresholds are fixed for all the frames with a particular profile of the proposed system and are denoted by N_r and $\boldsymbol{\delta}$= $\{\delta_1, \delta_2, ...\delta_z, ...\delta_Z\}$ respectively.

The principal parameter of the proposed system is the total number of regions, N_r. Let B_s^i denote the side information bits for frame i, and it can be computed as

$$B_s^i = (N_r - 1) \cdot \lceil \log_2 S \rceil + N_r! + \lceil \log_2 Z \rceil, \tag{4.14}$$

which includes information on the starting sfb for each region, the priority list and the coding sequence but ignores the number of regions and values of priority thresholds. Suppose that by using prioritized bit-plane coding, the distortion for frame i is improved by Δd^i compared to sequential coding. This improvement can be achieved by using pure sequential coding if extra ΔB^i bits are assigned to the frame. With fixed values of $\boldsymbol{\delta}$ and Γ^i, N_r is optimized by

$$\arg \max_{N_r} \sum_{i=0}^{I-1} (\Delta B^i - B_s^i). \tag{4.15}$$

Next, with the optimized value of N_r and a fixed value of $\boldsymbol{\delta}$, Γ^i is in turn optimized

by

$$\arg\max_{\Gamma^i}(\overline{\Delta E^i}) \tag{4.16}$$

where the mean energy difference ratio $\overline{\Delta E^i}$ is computed as

$$\overline{\Delta E^i} = \frac{1}{N_r^i - 1}\sum_{p=1}^{N_r^i-1}\left(E^i[n^{p+1}] - E^i[n^p]\right) \tag{4.17}$$

with n^p denoting the region r_n^i and $P^i[n] = p$, $p \in \mathbb{N}$.

The third parameter $\boldsymbol{\delta}$ includes two sub-parameters - the number of threshold values Z and the values of the thresholds. Note that the optimized $\boldsymbol{\delta}$ does not necessarily include all the threshold values, as some δ_z are actually redundant. In this scenario, it is denoted as $\tilde{\boldsymbol{\delta}}$ and may include only several threshold values, e.g. $\tilde{\boldsymbol{\delta}}$= $\{\delta_2, \delta_3\}$ and $\tilde{Z} = 2$. It is assumed that all δ_z with

$$\begin{aligned} 0 < \frac{J-(z+1)}{z+1} < \frac{1}{2} \\ (\frac{2J}{3} - 1) < z < J - 1 \end{aligned} \tag{4.18}$$

are redundant as their values are very similar to δ_Z and these can be easily merged with δ_Z. Let $d(b)$ denote the distortion improvement when bit-plane b is coded. The values of δ_z can then be determined as $\delta_z = \overline{\Delta E^i}$ when

$$\begin{aligned} &\sum_{p=2}^{N_r}\sum_{s_p=\Theta(n^p)}^{\Theta(n^p+1)-1} d(b_J^{s_p} + b_{J-1}^{s_p} + ... + b_{J-z}^{s_p}) = \\ &\sum_{p=2}^{N_r}\sum_{s_p=\Theta(n^p)}^{\Theta(n^p+1)-1} d(b_J^{s_p} + b_{J-1}^{s_p} + ... + b_{J-(z-1)}^{s_p} + b_J^{s_{p-1}}) \end{aligned} \tag{4.19}$$

is satisfied. By combining Eqn.(4.19) and Eqn.(4.17), the optimized values of δ_z can be obtained.

The above optimization calculations including Eqns.(4.15) and (4.19) are not directly computable; however, their conditions do provide at least the qualitative insights for the selection of parameters. The near optimal parameters can thus be chosen based on statistical results obtained from a large number of test sequences.

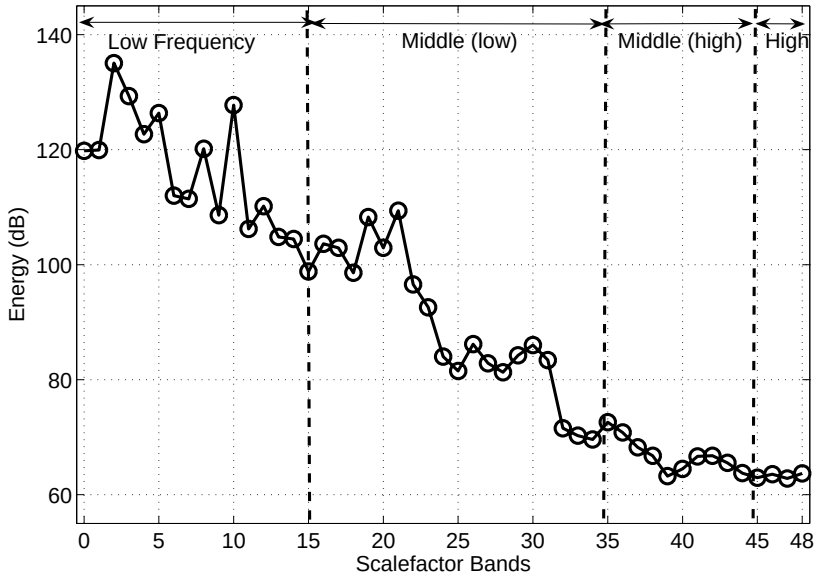
4.3.3 The Simplified SLS Implementation

The basic algorithm and parameter optimization have been described in the last two subsections. A further generalized modelling of the audio spectrum is proposed and discussed in this subsection. Based on this, a simplified structure of prioritized bit-plane coding is implemented in SLS, with the goal of optimizing the perceptual performance with the least added complexity, side information and syntax change.

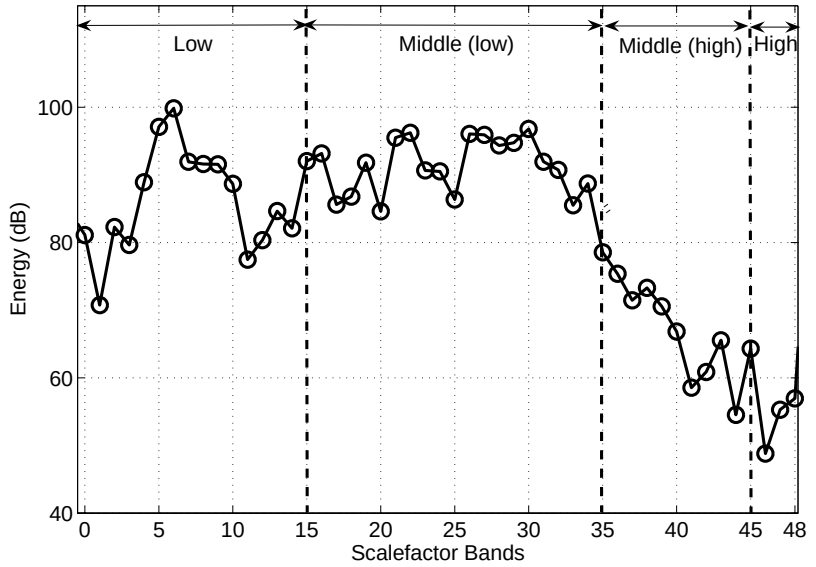
4.3.3.1 Modelling of the Audio Spectrum

It can be understood that if the value of N_r is big, a large amount of side information will be needed to process the prioritized coding scheme. On the other hand, if N_r is small, each region will be very broad and the efficacy of the prioritized coding scheme will be low. By using Eqn.(4.15), experiments are conducted for 15 standard test sequences. Statistical results show that the optimized value of N_r varies from 2 to 4 for most of the audio sequences. The audio spectrum for each frame is divided into four regions which are denoted by low frequency (r_{LF}), middle-low frequency (r_{MLF}), middle-high frequency (r_{MHF}) and high frequency (r_{HF}) regions with fixed region boundaries $\Gamma = \big\{\Theta(r_{\mathrm{MLF}}), \Theta(r_{\mathrm{MHF}}), \Theta(r_{\mathrm{HF}}))\big\}$. It is observed that most of the spectra can be categorized into two basic models based on the energy distribution in these regions. The typical plots of these two models are shown in Figure 4.6.

The main criterion that distinguishes these two models is $\Delta E(r_{\mathrm{LF}}, r_{\mathrm{MLF}})$, which

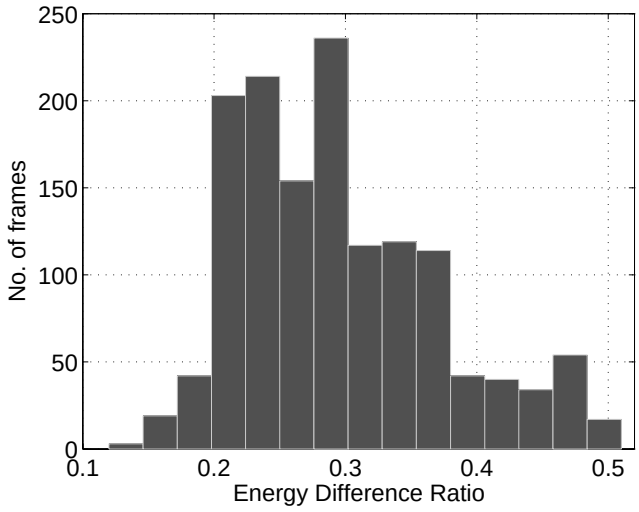


(a)

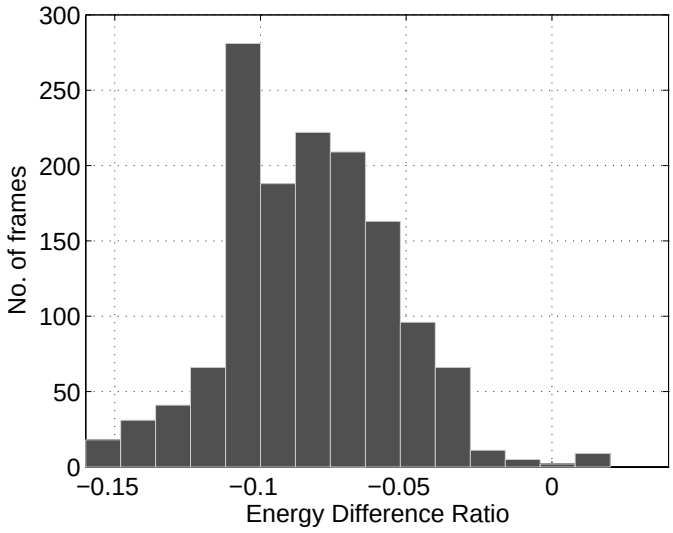


(b)

Figure 4.6: Typical spectrum plots for (a) Model I and (b) Model II.



(a)



(b)

Figure 4.7: Histogram of $\Delta E(r_{\mathrm{LF}}, r_{\mathrm{MLF}})$ for (a) avemaria.wav and (b) dcymbals.wav.

is the energy difference ratio between regions r_{LF} and r_{MLF}. Specifically, one frame is identified as

$$\begin{cases} \text{Model I}, & \text{if } \Delta E(r_{\mathrm{LF}}, r_{\mathrm{MLF}}) \geq \delta_{\mathrm{LM}} \\ \text{Model II}, & \text{if } \Delta E(r_{\mathrm{LF}}, r_{\mathrm{MLF}}) < \delta_{\mathrm{LM}} \end{cases} \tag{4.20}$$

where δ_{LM} is a constant threshold value (refer to Eqn.(4.9)). The coding priorities for Model I are assigned as

$$P_{\mathrm{I}}(r_{\mathrm{LF}}) > P_{\mathrm{I}}(r_{\mathrm{MLF}} + r_{\mathrm{MHF}}) > P_{\mathrm{I}}(r_{\mathrm{HF}}) \tag{4.21}$$

where r_{MLF} and r_{MHF} are merged as one region in the coding process. r_{LF}, $(r_{\mathrm{MLF}} + r_{\mathrm{MHF}})$ and r_{HF} are equivalent to $r_{\mathrm{LF}'}$, $r_{\mathrm{MF}'}$ and $r_{\mathrm{HF}'}$ for Model I. The corresponding priority level differences are denoted by $\tau_{\mathrm{I}}(\mathrm{LF}', \mathrm{MF}')$ and $\tau_{\mathrm{I}}(\mathrm{MF}', \mathrm{HF}')$. For Model II, the coding priorities are assigned as

$$P_{\mathrm{II}}(r_{\mathrm{LF}} + r_{\mathrm{MLF}}) > P_{\mathrm{II}}(r_{\mathrm{MHF}}) > P_{\mathrm{II}}(r_{\mathrm{HF}}) \tag{4.22}$$

with the merging of regions r_{LF} and r_{MLF}. Similarly, $(r_{\mathrm{LF}} + r_{\mathrm{MLF}})$, r_{MHF} and r_{HF} are equivalent to $r_{\mathrm{LF}'}$, $r_{\mathrm{MF}'}$ and $r_{\mathrm{HF}'}$ for Model II and their priority level differences are denoted by $\tau_{\mathrm{II}}(\mathrm{LF}', \mathrm{MF}')$ and $\tau_{\mathrm{II}}(\mathrm{MF}', \mathrm{HF}')$.

The histograms of $\Delta E(r_{\mathrm{LF}}, r_{\mathrm{MLF}})$ of two example test sequences, avemaria.wav and dcymbals.wav, are shown in Figure 4.7. It can be observed that for the first test sequence, most of the energy ratios are positive, which implies that most of the frames fall into Model I if $\delta_{\mathrm{LM}} = 0$. It is the opposite for dcymbals.wav.

4.3.3.2 Integrated Structure

The integrated structure of SLS with the prioritized bit-plane coding based on the above-mentioned models is depicted in Figure 4.8. In this structure, a conditional

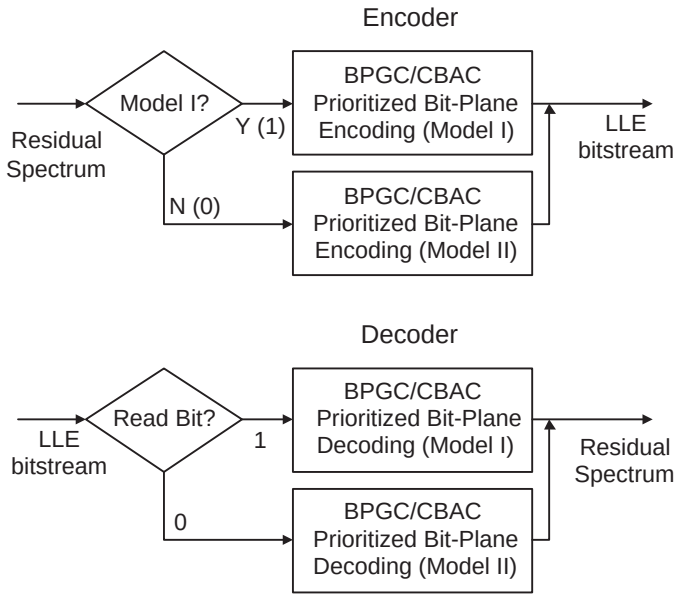


Figure 4.8: The integrated structure of SLS with frequency region based prioritized bit-plane coding.

switch (Eqn.(4.20)) and two alternative prioritized bit-plane coding sequences are adopted to replace the original sequential bit-plane coding block of SLS.

As it is assumed that all the parameters are fixed in this profile, the side information for each frame is uniform at 1 bit. There is also a reserved bit in SLS bit-plane coding which can be used as the switch function. This means that there is no additional side information to the bitstream at all. In addition, the complexity of this profile is maintained as well, since the only added computation is the energy difference calculation for the switching criteria.

4.3.3.3 Parameter Setting

It is observed from Figure 4.2 that for sfbs with energy equal to or lower than 70dB, the corresponding NMRs are relative low. This can be explained by the

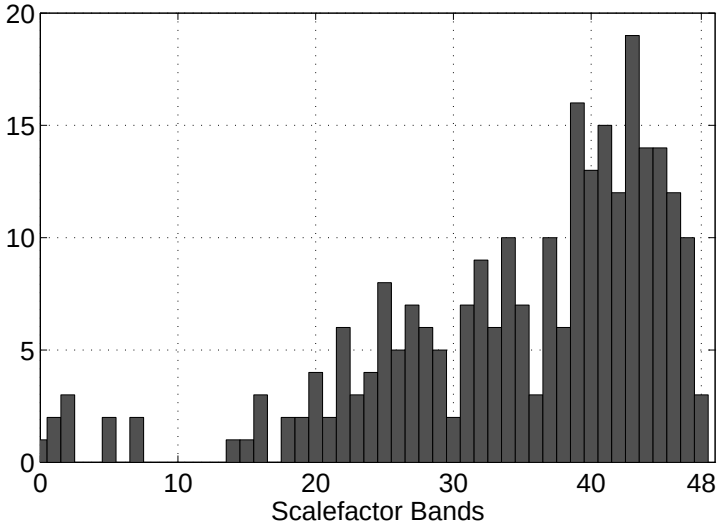


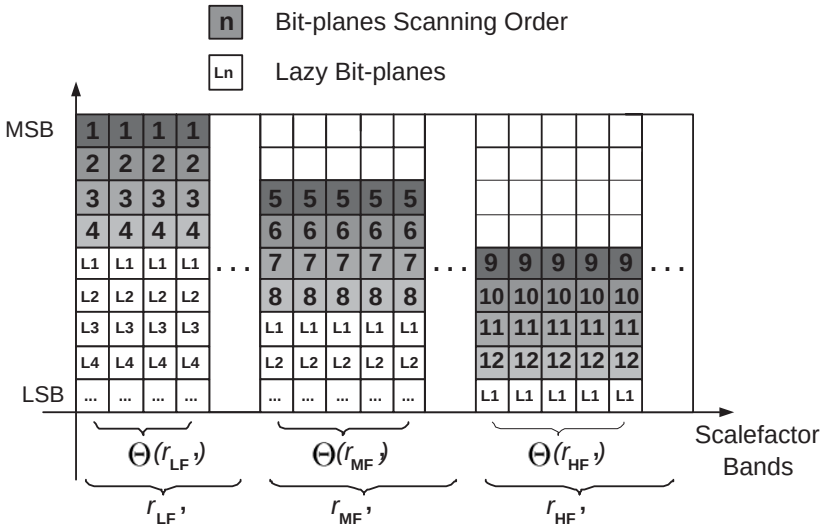
Figure 4.9: Histogram of the start sfb of the low energy region (the energy level is equal to or less than 70dB).

mask computation in Eqn.(4.2). Therefore, for each frame i,

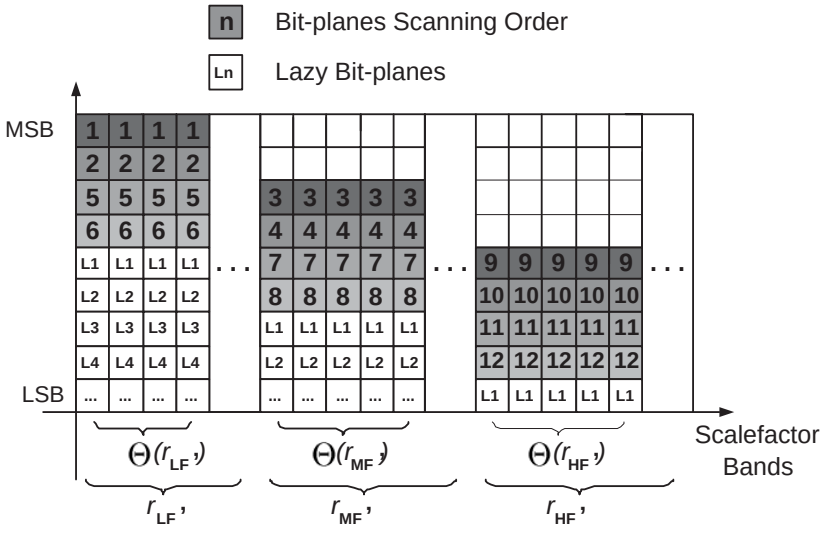
$$\Theta(r^i_{\mathrm{HF}}) = \{s' | \forall s : (s < s' \Longrightarrow E[s] > 70\mathrm{dB})\}. \tag{4.23}$$

A statistical study is performed on the region in which the energy level is equal to or less than 70dB. The histogram result is shown in Figure 4.9 with the horizontal axis denoting the start sfb of the low energy region. The data is extracted from 150 frames out of the 15 standard test sequences. It is noted that for most sequences, the energy level declines to 70dB from $s = 45$. In addition, the last 4 sfbs correspond to the highest frequency region and the noise are relatively less evident to human perception. Therefore, the last 4 sfbs are always grouped as one and assigned with the lowest priority. As such, these low-energy sfbs should always be assigned with low coding priorities.

The other parameters including energy difference ratio threshold δ_{LM}, the region boundaries $\Theta(r_{\mathrm{MLF}})$, $\Theta(r_{\mathrm{MHF}})$ and coding priority differences $\tau^i(\mathrm{LF}', \mathrm{MF}')$,

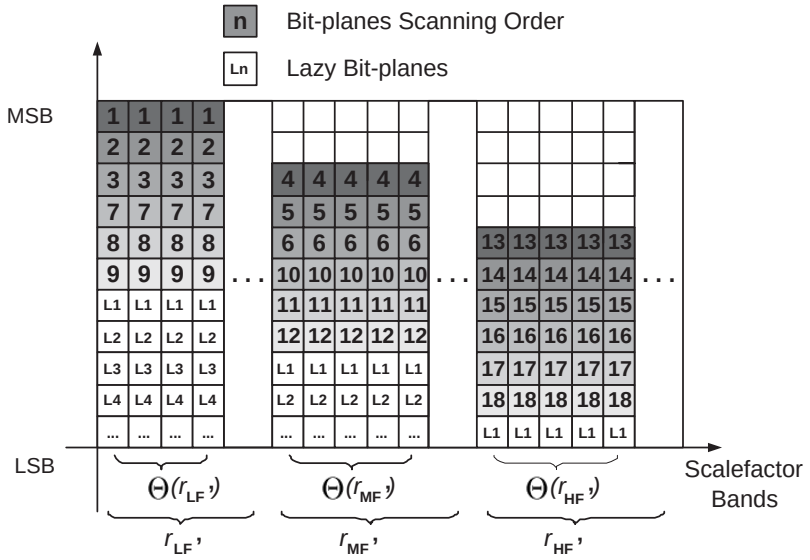


(a)

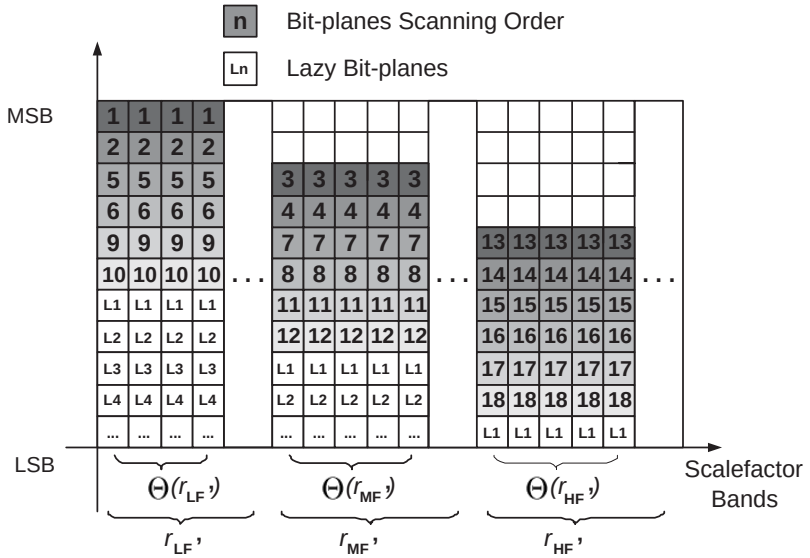


(b)

Figure 4.10: Bit-plane coding order for BPGC coding with (a) Model I (b) Model II.



(a)



(b)

Figure 4.11: Bit-plane coding order for CBAC coding with (a) Model I (b) Model II.

Table 4.1: Parameters setting for BPGC/CBAC coding

CBAC ($J = 6$)			
Parameter	Value	Parameter	Value
δ_{LM}	0	$\tau_{\mathrm{I}}(\mathrm{LF}', \mathrm{MF}')$	3
$\Theta(r_{\mathrm{MLF}})$	15	$\tau_{\mathrm{II}}(\mathrm{LF}', \mathrm{MF}')$	2
$\Theta(r_{\mathrm{MHF}})$	35	$\tau_{\mathrm{I}}(\mathrm{MF}', \mathrm{HF}')$	6
$\Theta(r_{\mathrm{HF}})$	45	$\tau_{\mathrm{II}}(\mathrm{MF}', \mathrm{HF}')$	6
BPGC ($J = 4$)			
Parameter	Value	Parameter	Value
δ_{LM}	0	$\tau_{\mathrm{I}}(\mathrm{LF}', \mathrm{MF}')$	4
$\Theta(r_{\mathrm{MLF}})$	15	$\tau_{\mathrm{II}}(\mathrm{LF}', \mathrm{MF}')$	2
$\Theta(r_{\mathrm{MHF}})$	35	$\tau_{\mathrm{I}}(\mathrm{MF}', \mathrm{HF}')$	4
$\Theta(r_{\mathrm{HF}})$	45	$\tau_{\mathrm{II}}(\mathrm{MF}', \mathrm{HF}')$	4

$\tau^i(\mathrm{MF}', \mathrm{HF}')$ can be derived according to Eqns.(4.16) and (4.19). The optimized parameters do vary for different items. However, to keep the side information and complexity to the minimum, the parameters are fixed for a particular profile. The parameters for one typical profile are summarized in TABLE 4.1 and the corresponding coding sequences are depicted in Figures 4.10 and 4.11 for BPGC and CBAC coding, respectively.

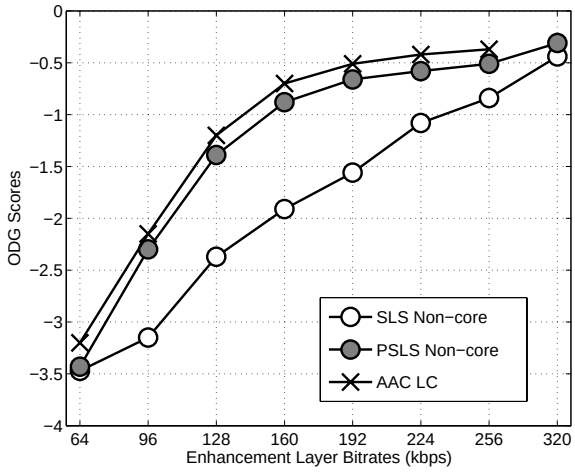
4.4 Experimental results

By replacing the sequential BPGC/CBAC bit-plane coding block in SLS with the simplified implementation of prioritized bit-plane coding as depicted in Figure 4.8, the combined structure is denoted as PSLS (for prioritized SLS). In the PEAQ test, the performance of the PSLS coder is evaluated against 2 main sets of test data where one set uses CBAC and the other one uses BPGC coding. The full list of items can be found in TABLE 2.2, with the parameters shown in TABLE 4.1. For each test set, there are 3 subsets with core bitrates at 0, 32 and 64kbps in total for

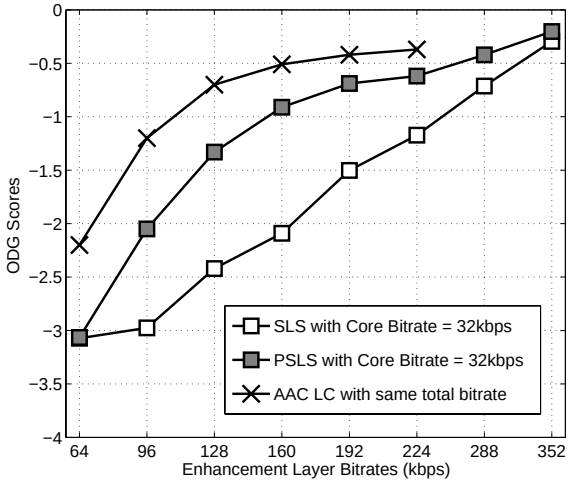
stereo channels. The enhancement layer bitrate for each subset starts from 64kbps up to the bitrate where the corresponding ODG value falls in the range of (-0.5 $\sim$ 0). The performance of PSLS is compared with that of the original SLS and state-of-the-art perceptual audio codec AAC. The AAC codec selected for comparison is the MPEG-4 AAC *verification model* (VM) *low complexity* (LC) profile. The AAC core in SLS and PSLS is MPEG-4 VM LC profile too. The performance of PSLS non-core which is the main enhancement target of the proposed method is further evaluated using the subjective test with comparison of the AAC-LC and SLS non-core at common lossy bitrates of 64, 96, 128, 192 and 256kbps (in total for stereo channels) for all the test items. The perceptual quality of lossy audio reconstruction when SLS and PSLS are working at zero and low core bitrates is evaluated based on both objective and subjective tests. Specifically, the objective test is conducted by using OPERA voice/audio quality analyzer [65]) which applies the ITU-R BS.1387 PEAQ test method and the subjective test is conducted by using ITU-R BS.1116 [81].

The summarized results are plotted in Figures 4.12, 4.13 and 4.14, with numerical ODG results of SLS non-core with enhancement bitrates of 128, 192 and 256kbps (in total for stereo channels) shown in TABLE 4.2. For the PEAQ test in Figures 4.12 and 4.13, the grading scale of ODG ranges from -4 ("very annoying") to 0 ("imperceptible difference"). For the BS. 1116 test in Figure 4.14, the grading scale ranges from 1 ("Very Annoying") to 5 ("Imperceptible"). Here are a few key observations:

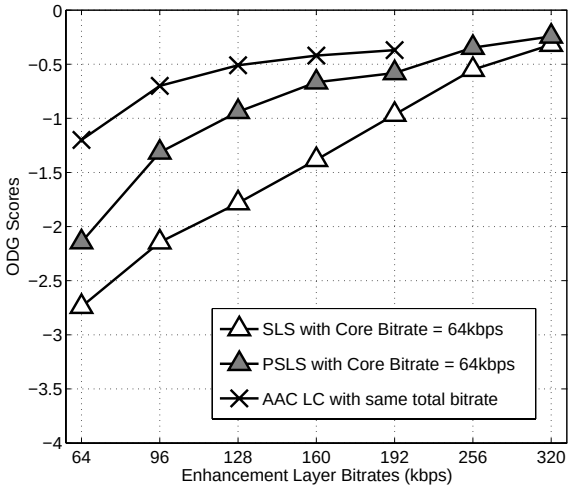
- Compared with the perceptual quality of SLS in terms of ODG scores, PSLS displays obvious improvements for both of the CBAC and BPGC coding test data sets. The scalable quality curves with increasing enhancement layer bitrates for original SLS are rather linear when core bitrates are low. While



(a)

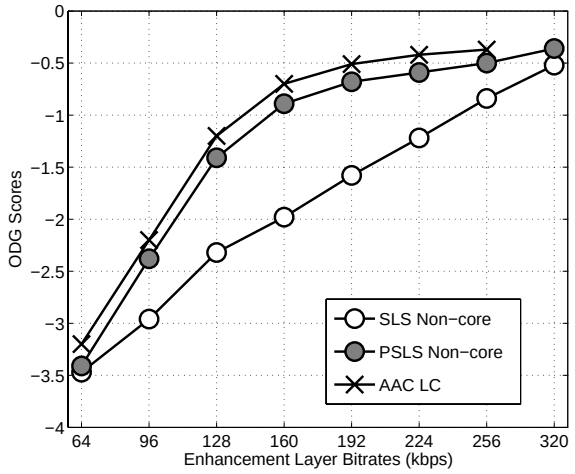


(b)

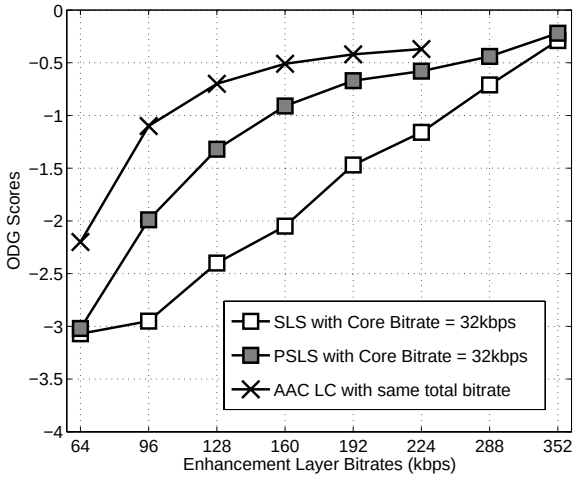


(c)

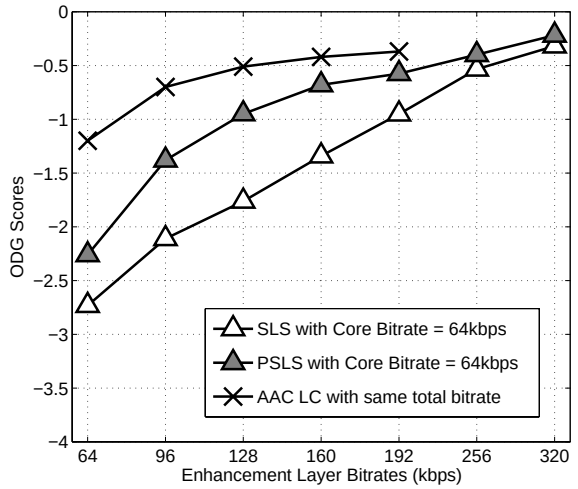
Figure 4.12: Objective results for CBAC coding with core bitrate at (a) 0kbps (b) 32kbps (c) 64kbps. It should be noted that the total bitrate is equal to the sum of the core bitrate and the enhancement bitrate.



(a)



(b)



(c)

Figure 4.13: Objective results for BPGC coding with core bitrate at (a) 0kbps (b) 32kbps (c) 64kbps. It should be noted that the total bitrate is equal to the sum of the core bitrate and the enhancement bitrate.

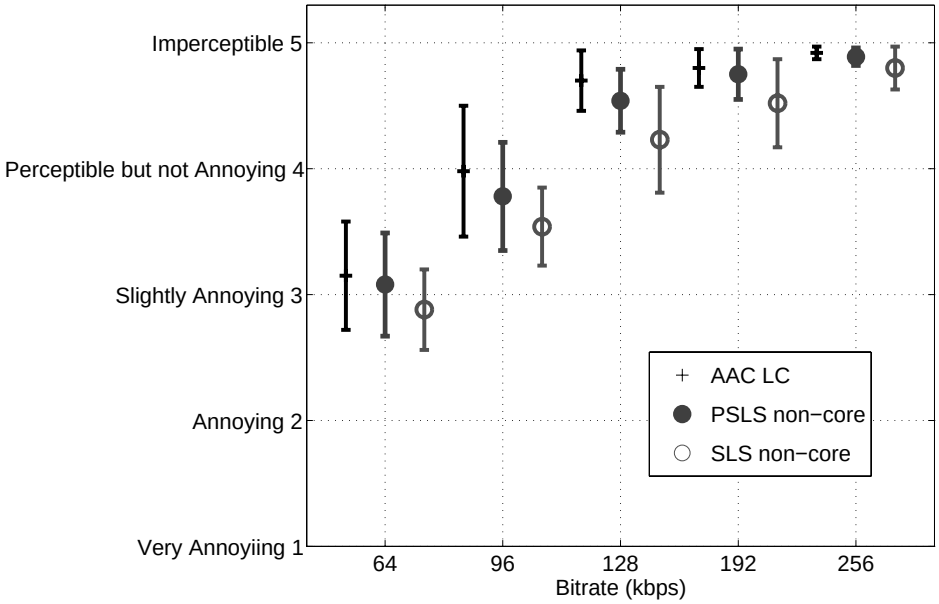


Figure 4.14: Subjective test results of PSLS non-core with comparison of SLS non-core and AAC LC at variable lossy bitrates (1: Very Annoying 2: Annoying 3: Slightly Annoying 4: Perceptible but not Annoying 5: Imperceptible).

for the PSLS, a non-linear, or a more "perceptually representative" scalability is achieved.

- In both the objective and subjective tests, it is observed that PSLS has significantly shorten the quality gap between perceptual audio codec and fully scalable audio codec, though AAC LC still outperforms PSLS at all test bitrates. However, comparing with AAC, SLS and PSLS have two important features: the fine granular scalability and the lossless coding capability.

- The more effective improvement range of PSLS begins when the enhancement layer bitrates are higher than 64kbps for the first two subsets. In addition, the improvement range begins when the enhancement layer bitrate is equal to 64kbps for the third subset. This can be explained by the fact that since only 4 frequency regions are considered, the priorities assigned by PSLS is

rather coarse. As a result, in very low bit-plane coding bitrates, there will be no obvious improvements. The improvements are more significant in the middle-range bitrates between 96 to 192kbps.

- The improvements are more significant for the non-core subsets, as the proposed prioritized bit-plane coding works more efficiently when the signal spectrum is more unbalanced.

- From TABLE 4.2 it is observed that PSLS produces perceptual quality results which are more stable and balanced. The improvements are relatively small for one test item "cymbal.wav", as this item contains a rather extended state of silence and its spectrum is more balanced compared to those of the rest of the test data.

The quality evaluation conducted shows that the frequency region based prioritized bit-plane coding does improve the perceptual quality of SLS in a wide range of bitrates. In addition, as mentioned in last section, this comes with no unwanted extra side information bits, as the only additional bit per frame can be implemented using the reserved 1-bit of SLS. It is expected that greater improvements can be achieved in a wider bitrate range when a spectrum is divided into more frequency regions. However, this entails more complex system and more side information. Since SLS is already standardized, the proposed system is in fact a simple yet efficient modification to it. It does not require a complete overhaul and fits this purpose well.

4.5 Conclusion

Comparing with the well known AAC-LC, SLS has two novel features: the fine granular scalability and the lossless coding capability. However, the lossy quality

Table 4.2: ODG performance of enhanced SLS Non-core comparing with that of original SLS Non-core (Improvement = PSLS-SLS).

Items index	ODG Scores								
	128kbps			192kbps			256kbps		
	SLS	PSLS	Improvement	SLS	PSLS	Improvement	SLS	PSLS	Improvement
1	-2.62	-1.33	1.29	-1.68	-0.66	1.02	-0.73	-0.33	0.40
2	-1.56	-1.13	0.43	-0.84	-0.50	0.34	-0.41	-0.41	0.01
3	-3.39	-1.54	1.86	-2.80	-0.70	2.09	-1.62	-0.49	1.13
4	-1.67	-1.22	0.44	-0.92	-0.54	0.38	-0.43	-0.44	-0.01
5	-1.61	-1.28	0.33	-0.90	-0.55	0.35	-0.46	-0.44	0.02
6	-3.14	-2.44	0.70	-2.55	-1.74	0.81	-1.74	-1.20	0.53
7	-2.33	-1.26	1.07	-1.70	-0.47	1.23	-0.75	-0.39	0.36
8	-2.92	-1.37	1.55	-2.00	-0.62	1.38	-0.91	-0.46	0.44
9	-3.30	-1.31	1.99	-2.58	-0.58	2.00	-1.48	-0.42	1.06
10	-1.85	-1.41	0.44	-0.91	-0.61	0.30	-0.44	-0.48	-0.04
11	-2.20	-1.36	0.84	-1.26	-0.53	0.73	-0.56	-0.43	0.14
12	-2.92	-1.09	1.83	-1.96	-0.83	1.13	-1.35	-0.76	0.59
13	-1.73	-1.40	0.33	-0.82	-0.59	0.22	-0.44	-0.45	-0.01
14	-2.65	-1.39	1.27	-1.69	-0.58	1.10	-0.85	-0.47	0.38
15	-1.63	-1.26	0.37	-0.83	-0.51	0.31	-0.44	-0.42	0.02
Average	-2.37	-1.39	0.98	-1.56	-0.67	0.89	-0.84	-0.51	0.33

performance of the current SLS structure at low or non-core bitrates are shown to be inefficient as compared to the performance at high core bitrates. Inspired by observations on the energy distribution of the residual spectrum, a frequency-region based prioritized bit-plane coding has been proposed along with an analysis on parameter optimization. Based on the statistical modelling of the spectrum, a much more simplified implementation is designed for SLS. The results of simulations show that with zero extra bit involved and merely trivial added complexity, SLS with low core bitrates is significantly improved by the proposed system with variable intermediate bitrate combinations. This is especially important for the non-core mode of SLS as a perceptually more efficient fully-scalable coding is achieved. The generalized frequency-region based prioritized bit-plane coding can also be applied in other bit-plane coding scenarios besides SLS.

Chapter 5

Efficient Stereo Bitrate Allocation for SLS

5.1 Background

Stereo audio involves the recording and reproduction of audio using two or more independent channels. It is able to create a pleasant and natural impression of sound heard from various directions as in natural hearing which is often contrasted with monophonic audio.

Joint stereo coding, which takes advantage of the fact that both channels of a stereo channel pair are highly correlated, has become commonly used as an efficient technique to enhance the quality of compressed digital audio. The stereophonic irrelevancies and redundancies are exploited in joint stereo coding to reduce the total bitrate. One of the most popular joint stereo encoding techniques is the *mid/side* (M/S) stereo coding, which is widely used in many audio codecs such as MPEG-1 *Layer 3* (MP3) [82] and MPEG-4 AAC [14]. M/S coding transforms the *left* (L) and *right* (R) channels into a *mid* (M) channel and a *side* (S) channel. As shown in Figure 5.1, the M channel is the sum of the L and R channels and the S channel is the difference of the L and R channels. Re-arranging the data into M and S channels usually results in a situation where the data in the M channel

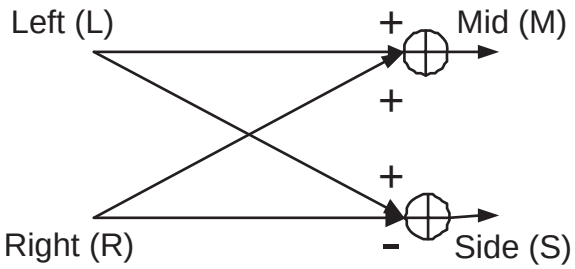


Figure 5.1: Mid/Side stereo coding.

have much larger amplitudes than those in the S channel if the L and R channels are highly correlated. In this case, the S channel can then be accurately encoded using fewer bits and more resources can then be employed efficiently on the M channel. When the M and S channels are subsequently reformatted back to L and R channels, the net result will be a more accurate representation of the original L and R input channels. In addition, M/S coding retains the audio perfectly and does not introduce artifacts by itself.

M/S stereo coding is implemented in various ways with different codecs. The coder in [83] suggests that for each frequency partition, M/S should only be switched on if the corresponding masking thresholds of L and R channels calculated by the psychoacoustic model vary by less than 2dB. The masking thresholds of M and S channels are then calculated based on the basic thresholds of M/S which can be obtained using the same model of L/R, together with a factor called *masking level difference* (MLD). MLD calculates a second level of detectablity of noise across frequency in the M/S channels, and it can be computed by multiplying the spread signal energy by a MLD factor (shown graphically in [83]). Specifically, the actual threshold for the M and S channels, THR_M and THR_S, are calculated respectively as

$$\mathrm{THR}_M = \max\left(\mathrm{THR}_M^B, \min(\mathrm{THR}_S^B, \mathrm{MLD}_S)\right),$$

$$\mathrm{THR}_S = \max\left(\mathrm{THR}_S^B, \min(\mathrm{THR}_M^B, \mathrm{MLD}_M)\right) \tag{5.1}$$

where THR_M^B and THR_S^B denote the basic thresholds for the M and S channels, respectively. The *perceptual entropy* (PE) [84] which reflects the minimum number of bits sufficient to code the signal below threshold can thus be calculated according to the corresponding signal to mask ratio. The bits are then assigned based on the PE of the M and S channels.

The concept of "*allocation entropy* (AE)", which reflects the number of bits for best quality instead of transparent quality as in PE, is developed in [85] to further enhance the performance of M/S coding in a perceptual coder. In this implementation the masking thresholds of M/S simply depend on the thresholds of the L/R channels. Specifically, AE for each channel is defined as

$$\mathrm{AE} = \sum_{j=0}^{M_{\mathrm{SFB}}} (W[j] \cdot \log_{10} \mathrm{SMR}[j]) \tag{5.2}$$

where j and $W[j]$ denote the sfb and the number of spectral lines in the band, respectively. $\mathrm{SMR}[j]$ is computed as

$$\mathrm{SMR}[j] = \begin{cases} \frac{E[j]}{T[j]B[j]}, & E[j] > T[j] \\ 0, & E[j] \leq T[j] \end{cases} \tag{5.3}$$

where $E[j]$ and $T[j]$ denote the spectrum energy and the masking threshold of the band j, respectively. $B[j]$ is the effective bandwidth proposed in [86]. The AE method can provide better performance than LAME player [87] and it has low complexity.

In SLS RM codec, M/S coding is applied. The bitrate for the two channels is equally allocated, regardless of the possible level difference between the M and

S channels. This is especially inefficient for the non-core mode of SLS, where the perceptual core is absent. The perceptual allocation algorithms in [84] and [85] are not directly applicable as there is no perceptual quantization process in non-core SLS. This chapter thus proposes new bit allocation strategies with negligible complexity for both the encoder and the truncator of SLS. Based on the prioritized nature of bit-plane coding, the available bitrate is assigned according to the energy ratio of the two channels. Although without using any psychoacoustic information, the proposed method is proven to be efficient in improving the perceptual quality for the non-core mode of SLS, or the fully scalable audio, without introducing any extra side information.

The rest of this Chapter is organized as follows. Section 2 describes issue of SLS on stereo bit allocation. The proposed enhancement methods for both the encoder and truncator are elaborated in Section 3. These are followed by presentation of extensive test results from the implementation of the proposed structure.

5.2 Bit Allocation Issue in SLS

In the RM encoder, the available bitrate is evenly distributed to the L and R channels or to the M and S channels. This bit allocation strategy is not that inefficient with the presence of the perceptual core as the level difference of the M/S channels can be adjusted by the different scalefactors used for M and S in the quantization stage of perceptual coding. However, for non-core SLS where the bit-plane coding is directly applied on the transformed coefficients, the coding quality depends much more on the available bitrate assigned to different channels. This even bit allocation scheme thus results in an extremely inefficient coding when the data in the M channel has a much higher amplitude level than those in the S

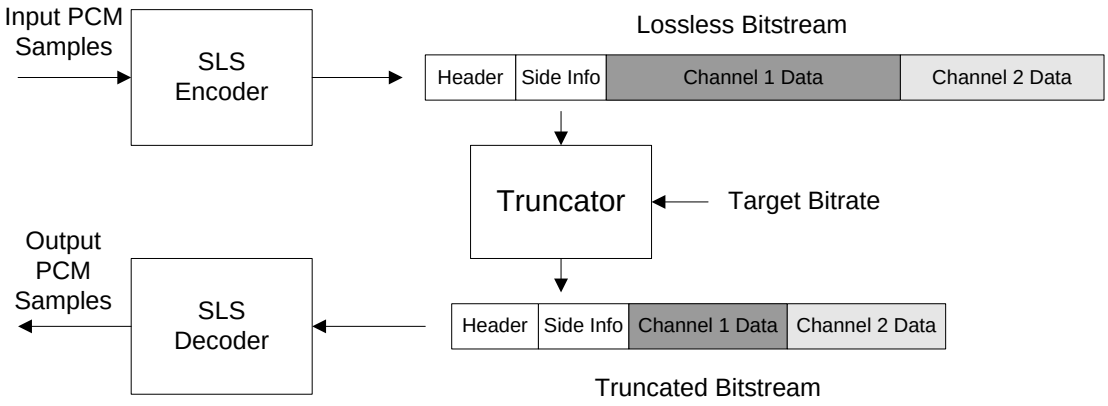


Figure 5.2: The SLS truncator.

channel.

The SLS bitstream can be directly decoded or truncated by a truncator as shown in Figure 5.2. The input bitstream is truncated for each frame based on the target bitrate. It should be noted that if the input bitstream is lossless, the length of two channels may be different. Yet in the RM truncator the available bits are also evenly distributed to the two channels. This is another inefficient bit allocation process in the SLS RM structure.

5.3 Efficient Stereo Bitrate Allocation

As described in Section 5.2, the stereo bit allocation algorithms in SLS RM encoder and truncator are still far from being optimized. An enhanced bit allocation algorithm is developed in this section for both the encoder and the truncator. It should be noted that though the proposed methods are mainly targeted at enhancing the non-core mode of SLS, they work in the normal mode (with core) as well.

5.3.1 Enhanced Encoder

It was mentioned in Section 5.1 that highly correlated L and R channels will result in M channels with relatively larger data values and S channels with smaller data values. In this case more resources should be located to M channel for a more accurate recovery of L/R channels. However, the perceptual allocation algorithms in [84] and [85] are not directly applicable in non-core SLS as only bit-plane coding is performed without any perceptual quantization process. Therefore, bit allocation should be adjusted to optimize the performance of pure bit-plane coding.

Firstly, the masking thresholds of the two channels can be estimated using the algorithm proposed in [85],

$$Thr_M = Thr_S \leq \min(Thr_L, Thr_R), \tag{5.4}$$

where Thr is the masking threshold of the corresponding channel. As the thresholds of the M and S channels are the same, the bits should be allocated in such a way that the distortions in both of the channels are similar and minimized. We would like to simplify the problem by making the assumption that all the bit-planes are directly coded with frequency assignment of 1/2. Therefore, to maintain similar distortions for the two channels, the bits should be allocated proportionally to the bit-plane levels of the two channels. The average bit-plane level, M_{ave}^M and M_{ave}^S for the M and S channel respectively, can be calculated as

$$M_{ave}^M = \frac{\sum_{i=0}^{N-1} M^M[i]}{N}, \quad M_{ave}^S = \frac{\sum_{i=0}^{N-1} M^S[i]}{N}, \tag{5.5}$$

where N is the total number of sfbs in the frame; $M^M[i]$ and $M^S[i]$ denote the maximum bit-plane number for the particular sfb i (M for M channel and S for S

channel, respectively). The maximum bit-plane number is the maximum number of the bit-planes for an sfb. For example, for the 15th sfb in Figure 2.6, the maximum bit-plane number is 8. It should be noted that Eqn. (5.5) is just an approximate and low complexity approach to realize Eqn. (5.4).

Alternatively, the average bit-plane level can be computed with the consideration of the band width. Specifically, M_{ave}^{M} and M_{ave}^{S} for the M and S channel respectively, can be calculated as

$$M_{ave}^{M} = \frac{\sum_{i=0}^{N-1} \left(M^{M}[i] \cdot W[i] \right)}{N}, \quad M_{ave}^{S} = \frac{\sum_{i=0}^{N-1} \left(M^{S}[i] \cdot W[i] \right)}{N}, \tag{5.6}$$

where $W[i]$ is the number of spectral lines in sfb i. If the number of available bits for the frame is B, it should be assigned to M and S channels as

$$B^{M} = B \cdot \frac{M_{ave}^{M}}{M_{ave}^{M} + M_{ave}^{S}}, \quad B^{S} = B \cdot \frac{M_{ave}^{S}}{M_{ave}^{M} + M_{ave}^{S}}, \tag{5.7}$$

where B^{M} and B^{S} are the numbers of bits for the M and S channel, respectively. As the maximum bit-plane number is already calculated for bit-plane coding in the original codec, the complexity for the proposed enhancement is very low.

5.3.2 Enhanced Truncator

Suppose that for a particular frame, the original bitstream lengths for the first and second channels are BS_1 and BS_2, respectively. The target bitstream (after truncation) length is denoted as BS^{T}. In SLS RM truncator, the bits are allocated as

$$BS_1^{T} = BS_2^{T} = \min \left\{ \min(BS_1, BS_2), \frac{BS^{T}}{2} \right\} \tag{5.8}$$

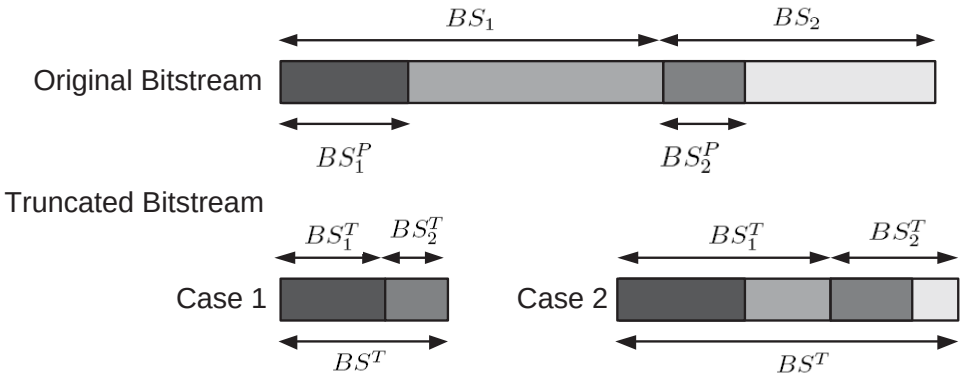


Figure 5.3: The enhanced truncator.

where BS_1^T and BS_2^T are the truncated bitstream length for the two channels, respectively. As illustrated in Figure 5.3, let BS_1^P and BS_2^P denote the perceptual core in the original bitstream. In the enhanced truncator, the following two cases are considered:

(i). $BS^T \leq BS_1^P + BS_2^P$. In order to optimize the basic perceptual quality, the bits are allocated as

$$\begin{aligned} BS_1^T &= BS^T \cdot \frac{BS_1^P}{BS_1^P + BS_2^P} \\ BS_2^T &= BS^T \cdot \frac{BS_2^P}{BS_1^P + BS_2^P} \end{aligned} \tag{5.9}$$

(ii). $BS^T > BS_1^P + BS_2^P$. In this case, firstly the perceptual core bitstreams are located. They are followed by the proportional enhancement bitstreams as

$$\begin{aligned} BS_1^T = BS_1^P + (BS^T - BS_1^P - BS_2^P) \cdot \\ \frac{BS_1 - BS_1^P}{BS_1 - BS_1^P + BS_2 - BS_2^P} \\ BS_2^T = BS_2^P + (BS^T - BS_1^P - BS_2^P) \cdot \\ \frac{BS_2 - BS_2^P}{BS_1 - BS_1^P + BS_2 - BS_2^P} \end{aligned} \tag{5.10}$$

The above two cases are based on the assumption that the target bitrate is less than the original bitrate. If this is not the case, the truncated bitstream will remain unchanged as the original one. For the non-core mode of SLS, $BS_1^P = BS_2^P = 0$.

5.4 Performance

To evaluate the performance of the enhanced encoder and truncator, two independent tests are conducted. In test 1, the bitstream encoded by the non-core mode of RM encoder and the enhanced encoders at various bitrates (64, 96, 128, 192, 256 and 384kbps, stereo) are directly decoded by the standard decoder. In particular, the enhanced encoder by applying Eqn. (5.5) and the one that applying Eqn. (5.6) are named as enhanced (encoder) 1 and enhanced (encoder) 2, respectively.

In test 2, the audio sequences are losslessly encoded using the RM encoder (non-core). The lossless bitstreams are then truncated by the RM truncator and the enhanced truncator at different bitrates. The truncated bitstreams are then decoded by the RM decoder. The perceptual quality of the decoded audio is measured in terms of ODG scores using OPERA voice/quality analyzer [65] which performs ITU-R BS.1387 PEAQ [66] test. Part of the *sound quality assessment material* (SQAM) [88] are used as test sequences, which includes 6 stereo files with a sampling frequency of 44.1 kHz and a resolution of 16 bits/sample. These test sequences are all speech signals, as the advantage of M/S stereo coding is more pronounced for these sequences.

The test results are shown in Figure 5.4 and Table 5.1 for test 1 and Figure 5.5 for test 2. It can be observed that both the enhanced encoder and the enhanced truncator significantly improve the perceptual quality of the first four test sequences at various bitrates. The results of the enhanced encoders 1 and 2 are

Table 5.1: ODG Performance of the original SLS RM codec and that of the SLS RM with the enhanced encoders 1 and 2. Test items: 1. Male speech (German) 2. Female speech (German) 3. Male speech (English) 4. Female speech (English) 5. Male speech (French) 6. Female speech (French).

Bitrate	Codec	1	2	3	4	5	6	Average
64 kbps	SLS non-core	-3.72	-3.80	-3.75	-3.81	-3.66	-3.77	-3.75
	Enhanced 1	-3.06	-3.39	-3.23	-3.33	-3.54	-3.56	-3.35
	Enhanced 2	-3.03	-3.35	-3.11	-3.27	-3.58	-3.65	-3.33
96 kbps	SLS non-core	-3.14	-3.45	-3.03	-3.35	-2.93	-3.15	-3.18
	Enhanced 1	-2.24	-2.59	-1.92	-2.18	-2.48	-2.68	-2.35
	Enhanced 2	-2.22	-2.59	-1.77	-2.16	-2.57	-2.80	-2.35
128 kbps	SLS non-core	-2.49	-2.86	-2.05	-2.56	-2.17	-2.35	-2.41
	Enhanced 1	-1.64	-2.10	-1.59	-2.10	-1.93	-2.17	-1.92
	Enhanced 2	-1.50	-2.02	-1.55	-2.01	-1.97	-2.20	-1.88
192 kbps	SLS non-core	-1.88	-2.31	-1.51	-2.02	-1.85	-2.08	-1.94
	Enhanced 1	-0.93	-1.23	-0.68	-1.27	-1.14	-1.35	-1.10
	Enhanced 2	-0.69	-1.16	-0.69	-1.26	-1.26	-1.57	-1.11
256 kbps	SLS non-core	-1.30	-1.65	-0.71	-1.26	-1.03	-1.15	-1.18
	Enhanced 1	-0.47	-0.77	-0.34	-0.84	-1.06	-1.11	-0.77
	Enhanced 2	-0.34	-0.65	-0.34	-0.84	-1.04	-1.13	-0.72
384 kbps	SLS non-core	-0.64	-0.95	-0.34	-0.82	-0.33	-0.42	-0.58
	Enhanced 1	-0.09	-0.15	-0.17	-0.43	-0.20	-0.14	-0.20
	Enhanced 2	-0.09	-0.15	-0.16	-0.43	-0.14	-0.24	-0.20

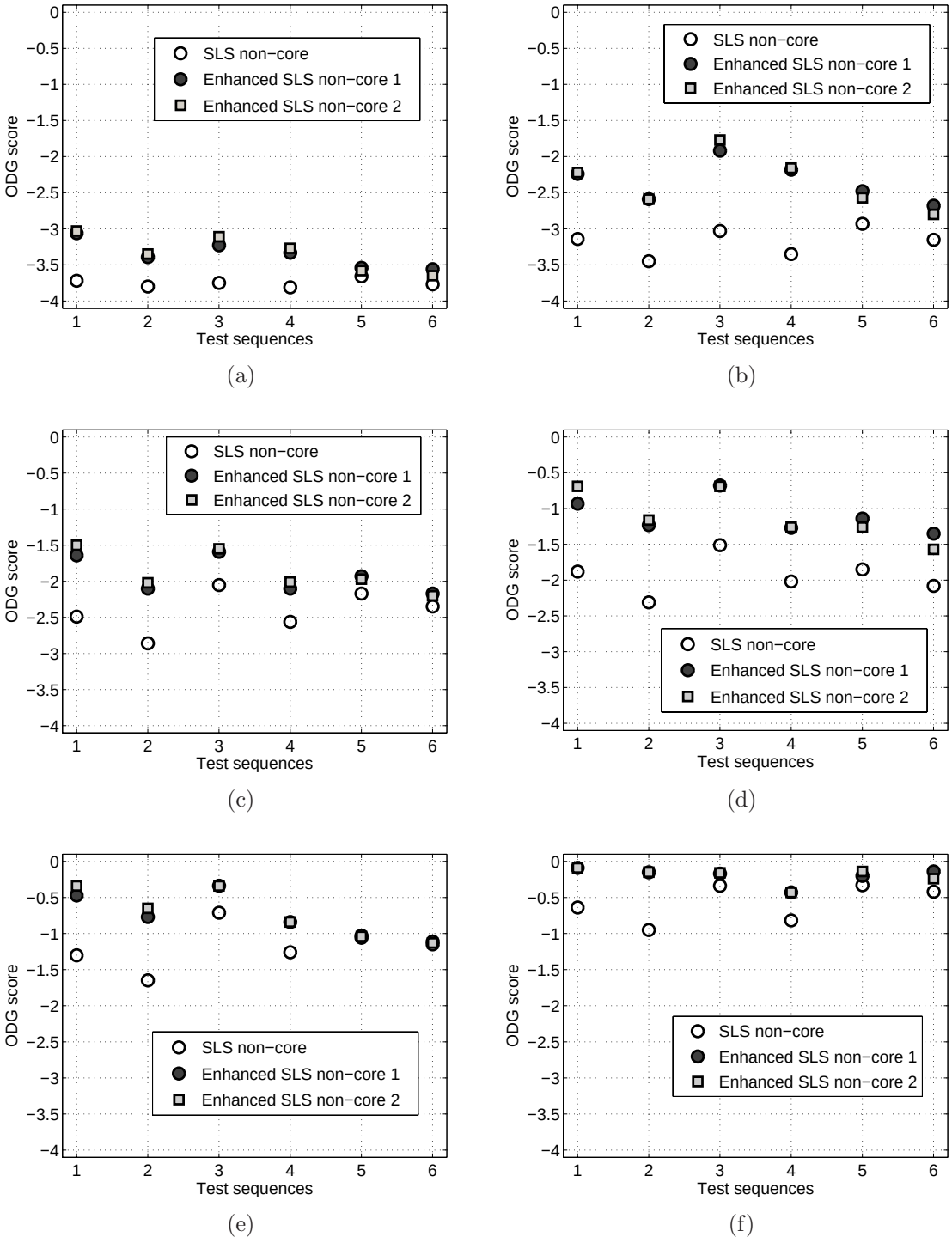


Figure 5.4: Performance of the original SLS RM codec and that of the SLS RM with the enhanced encoders 1 and 2 at non-core bitrate of (a) 64kbps (b) 96kbps (c) 128kbps (d) 192kbps (e) 256kbps (f) 384kbps. Test items: 1. Male speech (German) 2. Female speech (German) 3. Male speech (English) 4. Female speech (English) 5. Male speech (French) 6. Female speech (French).

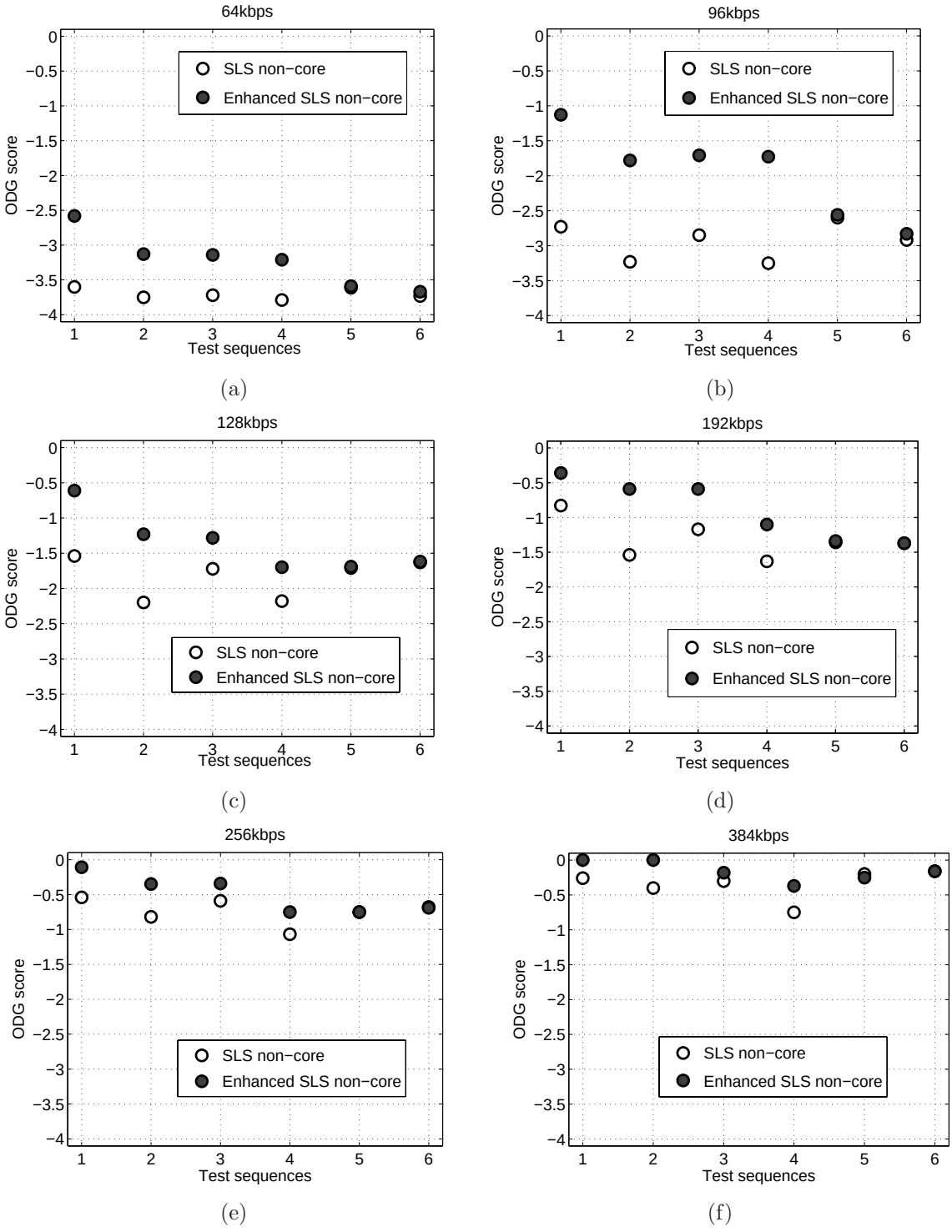


Figure 5.5: Performance of the original SLS RM codec and that of the SLS RM with the enhanced truncator at non-core bitrate of (a) 64kbps (b) 96kbps (c) 128kbps (d) 192kbps (e) 256kbps (f) 384kbps.

very similar. For the last two sequences, the improvements are not very obvious. This is due to the reason that the last two sequences are not as highly correlated as the first four for the L and R channels. Therefore, the advantage of M/S stereo coding is not apparent for the last two sequences.

5.5 Conclusion

In this chapter, efficient stereo bitrate allocation algorithms were proposed to enhance the performance of non-core SLS or the fully scalable audio. In the original SLS codec, the bitrate is evenly distributed to two channels, which is inefficient when M/S coding is used. The enhancing algorithms are specially designed for pure bit-plane coding with low complexity. In the enhanced encoder, the bitrates for the M/S channels are assigned according to the average bit-plane level of each channel. In the enhanced truncator, the bitrates are truncated according to the original length of the lossless bitstreams and the length of perceptually coded bitstreams. Test results show that for sequences with relatively high correlations between the L and R channels, the proposed methods significantly enhanced the M/S stereo coding scheme for fully scalable audio at various intermediate bitrates.

Chapter 6

Smart Enhancer for Scalable Lossless Coding

6.1 Background

Launched in April of 2003, Apple's iTunes music store has successfully proved the viability of online music sales. The music in the store is encoded in MPEG-4 AAC format [14] at 128kbps and priced at US$0.99 per song. As of January 2008, Apple has sold about 4 billion songs. However, an existing limitation for most online music sales is that the songs are only offered at fixed and lossy compressed bitrates. With the proliferation of broadband access and continuous decline of storage prices, there is a increasing number of music lovers who wish to purchase their favorite songs online at the highest resolution which is as good as and beyond the CD quality. On the other hand, some users may prefer to purchase songs that are cheaper but are encoded at a lower bitrate. This is because the perceptual audibility between, for example, a 96kbps and a 128kbps audio, is either transparent or not critical to them, especially when the music is played on mobile devices.

In order to satisfy multiple bitrate requirements from their customers, music

stores may need to archive different versions of the same piece of music at different bitrates on their servers. This is undoubtedly a burden to the server as the complexities of its database management and its storage space may be critically utilized. Alternatively, music stores may prefer to encode songs at the required bitrate only when a purchase order is received, which is both time consuming and computationally intensive. Moreover, there are customers who may wish to upgrade the music that they have purchased to a better quality as they may want to enjoy the music through a hi-fi system. In this case, the only option is to purchase and download the entire audio piece with a larger size, resulting in them having to keep different versions of the same audio piece for different devices. The feasibility of employing traditional fixed bitrate audio coding on an online music store offering multiple quality of a piece of music is therefore non-pragmatic to both the online stores and their customers.

At end of 2007, a music manager system based on SLS coding technology was deployed by Asia's largest online mobile music company *Soundbuzz* [89]. With this music manager system, the server maintained by the online stores is able to deliver music files to its clients at various bitrates and prices with single file archival for each piece of music. The processing of the files is hundreds of times faster than the traditional encoding process. Upgrading the quality of music by customers can also be easily and efficiently achieved by offering a "top-up" to the original version of music without the hassles of keeping multiple copies.

In this way, multi-bitrate music are currently provided by this online store. However, it does not mean that *multi-quality* music sales are already available. This is simply because the quality of music at a fixed bitrate actually varies for different tracks. The multi-quality model is actually more desirable, but it cannot be directly realized due to the lack of a clear link between scalable audio and its

perceptual quality. In this chapter, a smart enhancing algorithm is proposed, which brings an important function to the scalable audio coding that can be applied to satisfy the multi-quality music requirements. With a low quality perceptual audio input and its original (uncompressed) format, the smart enhancing algorithm enables the scalable audio coder to encode the minimum enhancing bitrate needed for this particular input to achieve a *transparent* quality. By applying the perceptual information from the state-of-the-art psychoacoustic model, the encoding based on this algorithm is able to stop at the optimal position where the transparent quality can just be achieved. In this way, the optimal scalable compression can be achieved for different audio tracks. The evaluation results show that compared to the traditional fixed bitrate (256kbps) for high quality lossy audio format, an average bitrate reduction of approximately 20% can be achieved for MPEG-4 standard SLS test items (48kHz/16bits).

The rest of this chapter is organized as follows. Firstly, the SLS music manager system is introduced. It is followed by a detailed description of the concept and proposed structure of smart audio enhancing function. The performance of the proposed system is evaluated at the end of this chapter.

6.2 The SLS Music Manager

By using SLS music manager system, Soundbuzz music store currently offers multiple versions of audio files with one archival of the lossless audio file. As shown in Figure 6.1, the first version is SLS high-quality lossy file at a total bitrate of 256kbps, which contains an AAC core format at 64kbps and a LLE enhancement at 192kbps. This high-quality lossy file can be truncated to a low-quality lossy file with AAC format at 64kbps, which can be used for mobile devices that normally

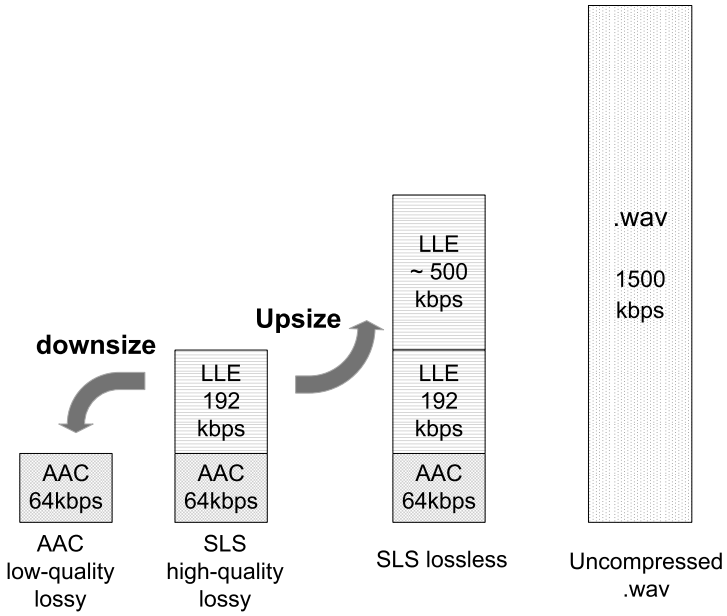


Figure 6.1: Audio formats provided by SLS music manager.

have limited storage size. A further "top-up" or "upsize" track (around 500kbps) for upgradation to lossless can be added on to high-quality lossy format and can be sold separately. The lossless format is also available as a whole and is sold at a higher price compared to SLS high-quality lossy format. Due to the FGS feature of the SLS lossless format, variable selling packages are available. In addition, some flexible options are enabled as well. For example, for a VIP customer, a better quality (higher bitrate) format can be offered at the same price.

Specifically, the functions of SLS music manager is depicted in Figure 6.2 and elaborated as follows. Firstly, for the store server side:

- The music in raw wave format (CD format) is encoded using an SLS encoder to produce the lossless compressed format consisting of the AAC core layer (64kbps) and the LLE layer.

- The archived format is then truncated into 3 tracks which consist of the AAC

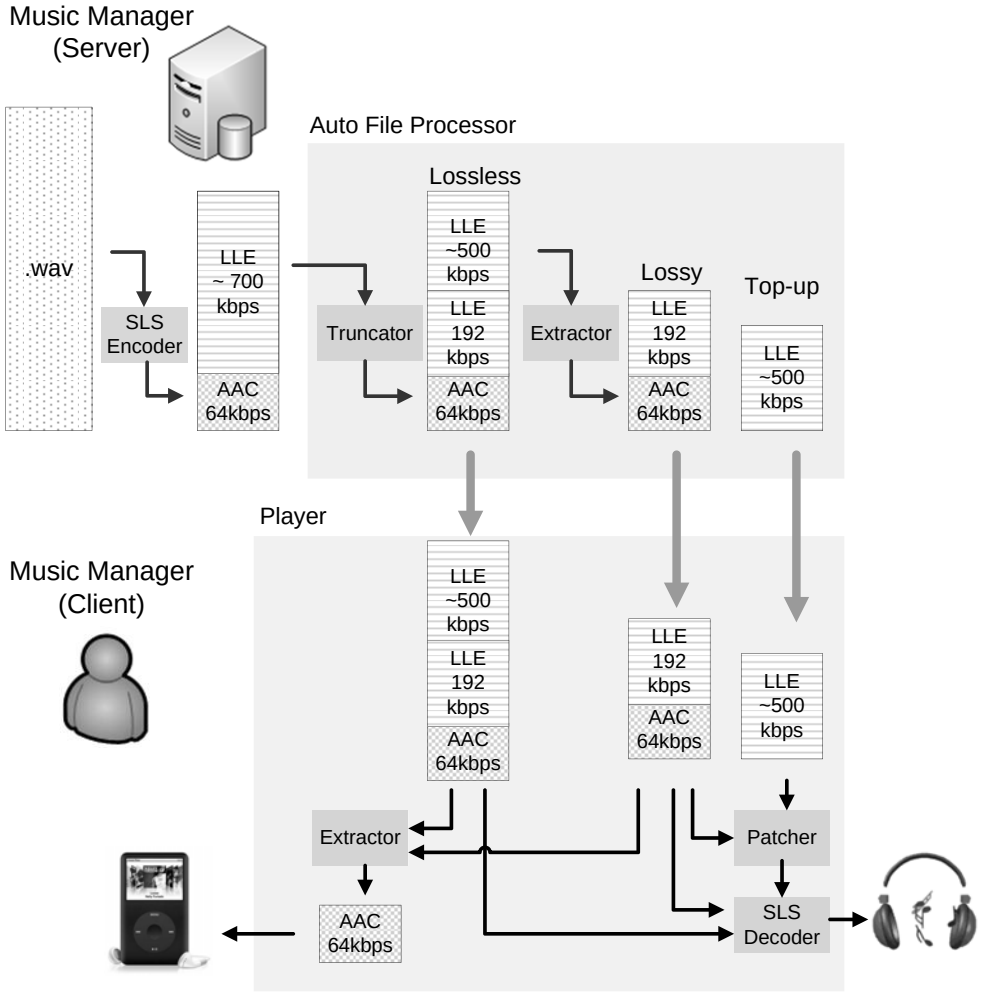


Figure 6.2: Structure of SLS music manager for store servers and clients.

track and two enhancement layer tracks with the first layer at 192kbps. All music pieces are stored in the server in this truncated lossless format only. If the client wants to buy the losslessly compressed audio, this full version will be available for them to download.

- If the client wants high-quality lossy version instead, the 256kbps format is extracted from the lossless format using a extractor.

- If the client has purchased lossy version and further decides to upgrade to

lossless version, top-up track is extracted from lossless format using extractor and sent to the client.

Secondly, for the client side:

- If the client has downloaded the lossy or the lossless version, these versions can be directly decoded by the player.
- If the client has downloaded lossy version and wants to upgrade to lossless, he/she just needs to download the top-up track. This top-up track can be patched together with the lossy version using a patcher to achieve the lossless audio format.
- If the client wishes to transfer the downloaded music to a mobile device which normally has limited storage size and playback quality, the AAC core can be extracted from the downloaded music using the extractor function of the music player.

6.3 Smart Enhancing Concept

6.3.1 Psychoacoustics and Transparent Quality

The field of psychoacoustics ([90–97]) is widely applied in current perceptual audio coders to achieve high compression by exploiting the fact that "undetectable" signal information can be abandoned in the coding process. These undetectable information is identified during signal analysis by incorporating into the coder several psychoacoustic principles, including absolute hearing thresholds, critical band frequency analysis, simultaneous masking, the spread of masking along the basilar membrane, and temporal masking.

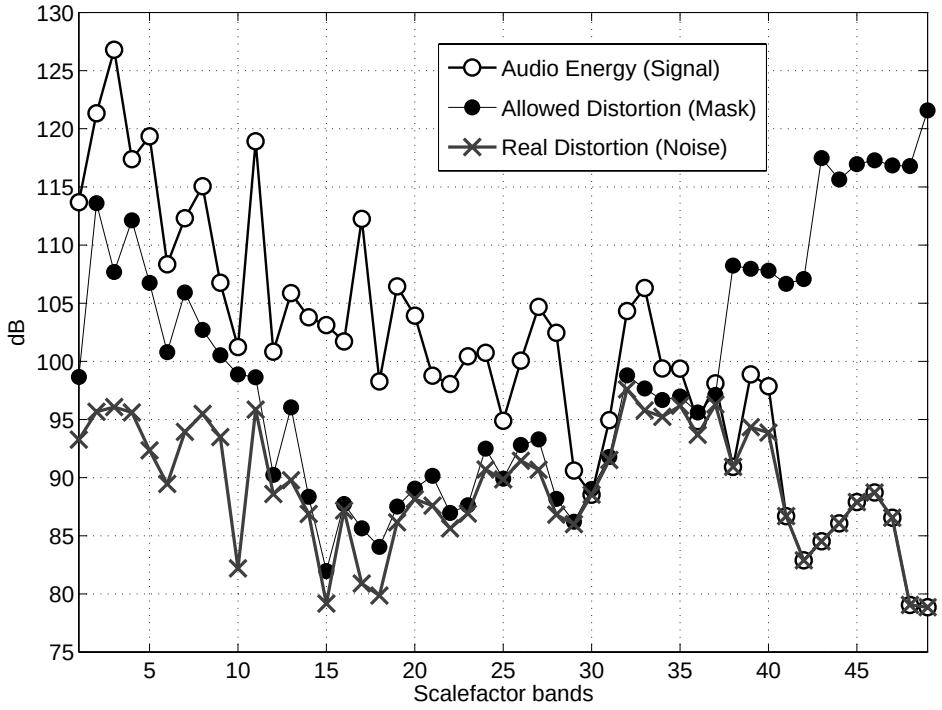


Figure 6.3: Signal, mask and noise plot for one frame in avemaria.wav (48kHz/16bit) coded by AAC at 128kbps.

Combining these psychoacoustic notions with basic properties of signal quantization has led to the theory of perceptual entropy, an estimate of the basic limit of transparent audio signal compression. Measurements in [98] and [99] suggest that a wide variety of CD-quality audio source material can be *transparently* compressed when the quantization distortion is controlled to be lower than the masking threshold.

An example plot of the signal, mask and quantization noise for one frame in excerpt named avemaria.wav is shown in Figure 6.3, where the excerpt is coded by MPEG-4 AAC reference codec (low complexity mode) at 128kbps.

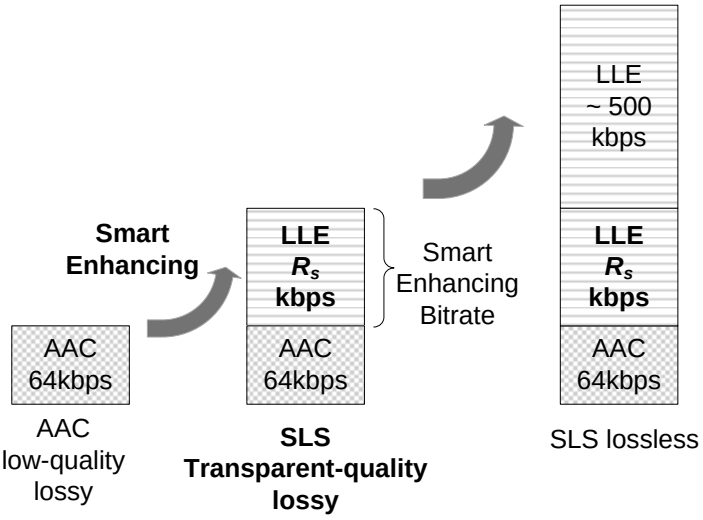


Figure 6.4: The smart enhancing function.

6.3.2 Smart Enhancing using SLS

In the music manager model of the SLS coder, the high-quality lossy enhancing bitrate is fixed at 192 kbps. However, the quality at this bitrate actually varies with different input audio sequences. For some audio sequences, this bitrate is good enough to achieve a transparent quality. However, 192kbps may not be enough for some audio inputs with high dynamic range or energy.

As shown in Figure 6.4, smart enhancing aims to provide an important function that, with a low-quality audio input and its original (uncompressed) format, it enables a scalable encoder to automatically encode the minimum amount of enhancing bits necessary to achieve the transparent quality for this particular input. This transparent quality lossy format can also have a further top-up to lossless format.

In addition, a transparent bitrate estimation function is also desired (Figure 6.5). Given a lossless SLS audio format (without smart enhancing process), this

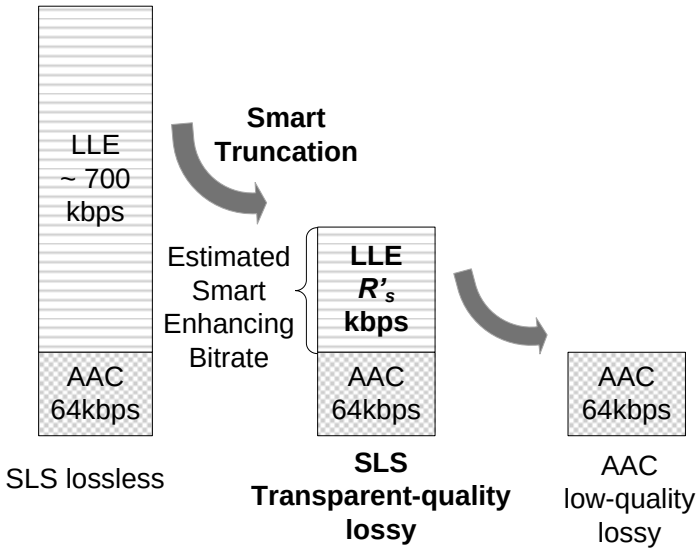


Figure 6.5: The smart truncation function.

estimation function is able to estimate the transparent bitrate for this audio sequence. This bitrate can be used to perform a smart truncation. However, this estimated transparent quality format may not be as accurate as the one obtained in Figure 6.4.

With the above smart enhancing functions, three versions of audio with three qualities which include low quality lossy, transparent quality lossy and lossless quality are thus available.

6.4 Smart Enhancing Structure

The smart enhancing can be achieved by a perceptually controlled bit-plane coding process and the enhancing bitrate can be estimated based on perceptual information. The detailed coding structure and the estimation model are described respectively in this section.

6.4.1 Perceptual Information Extraction

In the smart enhancing process, the perceptual information extracted from a psychoacoustic model is utilized in controlling the bit-plane coding scheme. Here it is assumed that similar to perceptual audio coding, if the distortion created in the scalable coding is below the mask, the transparent quality can be achieved. In the proposed implementation, the masking threshold for each sfb s, $M[s]$, in terms of dB is obtained by

$$M[s] = \begin{cases} C[s] \times \delta[s], & C[s] > 70\text{dB} \\ C[s] \times 1.1, & \text{otherwise} \end{cases} \tag{6.1}$$

where $s(0 \leq s < S)$ and S is the total number of sfb. $C[s]$ is the signal energy (in terms of dB) for sfb s and is computed by

$$C[s] = 10 \log_{10} \left(\sum_{k=O[s]}^{O[s+1]-1} c[k]^2 \right) \tag{6.2}$$

where $O[s]$ is the starting index of spectrum coefficient for sfb s and $c[k]$ is Int-MDCT coefficient. The mask to signal ratio $\delta[s]$ is calculated using psychoacoustic model II from MPEG-4 AAC.

6.4.2 Smart Enhancing Process

The basic idea of smart enhancing is depicted in Figure 6.6 and the flow chart of the enhancing process is shown in Figure 6.7. For each bit-plane, the encoding will stop at the optimal position to check the current distortion. If the distortion is still above the mask, the encoding process continues. Otherwise, the encoding for the current frame stops and proceeds to the next frame. It is also possible that the

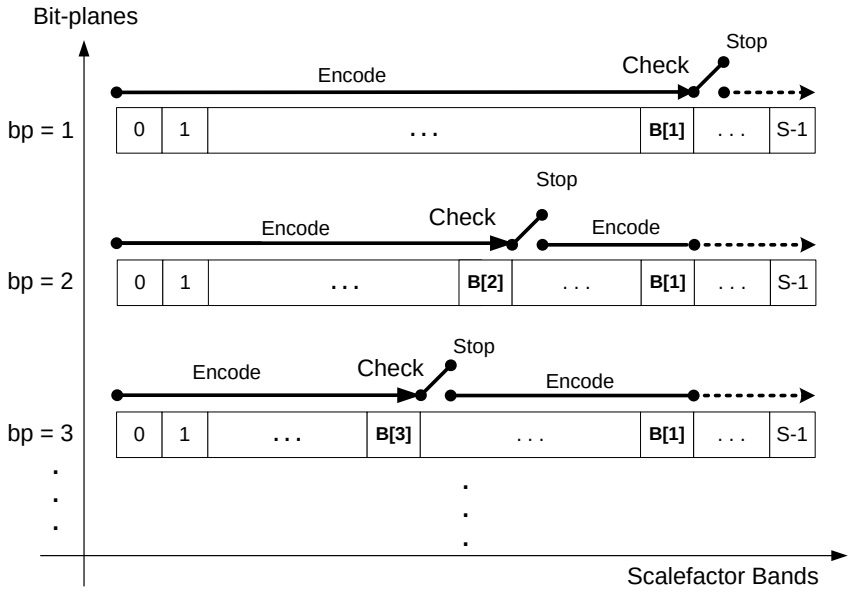


Figure 6.6: The proposed smart enhancing process.

encoding will proceed all the way to lossless, but the transparent-quality bitrate will be recorded in the meta data of the lossless format. Therefore, the proposed enhancing structure is based on two assumptions:

- The standardized SLS decoder structure remains unchanged for compatibility.
- The transparent-quality format can be further enhanced to lossless format (as indicated in Figure 6.4).

For $bp = 1$, the encoding procedure starts from $s = 0$, $s = 1$, ... to $s = B[1]$ by using BPGC/CBAC. $B[1]$ is defined as

$$L[s] \leq 0, \ \forall \ B[1] < s < S \tag{6.3}$$

where $L[s]$ is the lazy bit-plane of sfb s (as defined in Eqn. (2.27)). The sfbs after $B[1]$ are identified as low energy sfbs and will only be encoded after all the

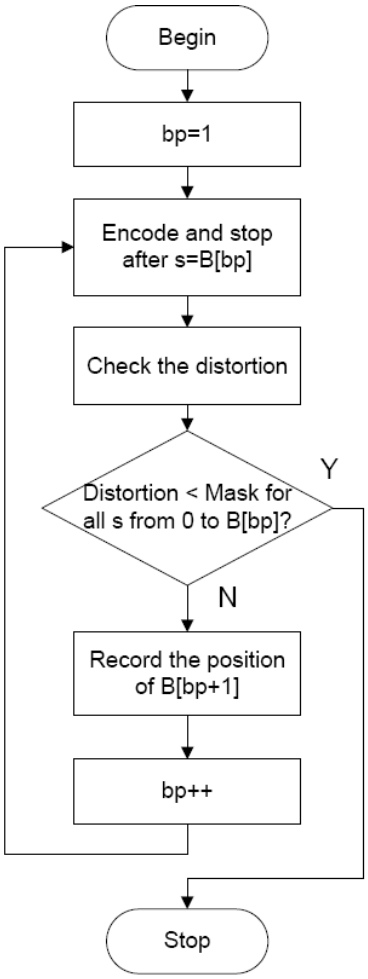


Figure 6.7: The flowchart of the smart enhancing process.

bit-planes for the previous sfbs are coded. These sfbs are also insignificant in perceptual meaning because even if they are not coded at all, the distortion will not exceed the mask (see Eqn. (6.1)).

The encoding for the first *bp* temporarily stops and it is followed by a bit-plane reconstruction for distortion check. The bit-plane reconstruction comprises two steps:

- Encoded bit-plane reconstruction
- Following bit-plane estimation

Given that $\bar{e}[k][T]$ is the reconstructed amount for residual frequency element k in sfb s, it can be computed by using encoded bit-planes from $bp = 1$ to $bp = T$ (terminating bit-plane where the encoding stops):

$$\bar{e}[k][T] = (2\hat{\varepsilon}[k] - 1) \cdot \sum_{bp=1}^{T} \left(b[k][bp] \cdot 2^{(b^M[s]-bp)} \right) \tag{6.4}$$

where $\hat{\varepsilon}[k]$ is the reconstructed sign symbol (0 or 1), $b[k][bp]$ is the bit symbol (0 or 1) and $b^M[s]$ is the total levels of bit-planes for the current sfb.

If $\bar{e}[k][T] \neq 0$, the reconstruction is further enhanced by an estimation process. Although the bit-planes below the current $bp = T$ are not coded yet, they can be estimated based on the Laplacian distribution feature of the frequency elements in SLS coding. This reconstruction enhancement is supposed to be performed in the SLS decoder too. Specifically, the estimated extra amplitude for the following bit-planes can be estimated using probability assignment in Eqn.(2.26) by

$$\tilde{e}[k][T] = (2\hat{\varepsilon}[k] - 1) \sum_{bp=T+1}^{b^M[s]} \left(Q_{bp}^{L[s]} \cdot 2^{(b^M[s]-bp)} \right), \tag{6.5}$$

and the final reconstructed spectrum coefficient $\hat{e}[k][T]$ is obtained as

$$\hat{e}[k][T] = \begin{cases} \bar{e}[k][T] + \tilde{e}[k][T], & \forall\ T < L[s] \\ \bar{e}[k][T], & \text{otherwise} \end{cases} \tag{6.6}$$

provided k is a coefficient in sfb s.

The total distortion energy for the sfb s, $d[s][T]$ (dB) is then computed as

$$d[s][T] = 10 \log_{10} \left[\sum_{k=O[s]}^{O[s+1]-1} \left(e[k] - \hat{e}[k][T] \right)^2 \right]. \tag{6.7}$$

For $bp = 1$, $T = 1$ and the encoding temporarily terminates at $s = B[1]$. The distortion $d[s][1]$ $(0 \leq s \leq B[1])$ is compared with the corresponding mask $M[s]$. If

$$d[s][1] < M[s], \ \forall \ 0 \leq s \leq B[1], \tag{6.8}$$

encoding of the current frame terminates and the encoder proceeds to encode the next frame. In other words, the encoding bitrate for this frame is already enough for the decoded frame to achieve transparent quality. Otherwise, a parameter $B[2]$ is obtained as

$$d[s][1] < M[s], \ \forall \ B[2] < s \leq B[1], \tag{6.9}$$

where $0 < B[2] \leq B[1]$. It can be easily understood that $B[2]$ is actually the last sfb with distortion that exceeds the mask after encoding of the first bit-plane is completed. For example, in Figure 6.8, $B[2] = 37$.

The encoding process then starts from $s = 0, s = 1,...$ up to $s = B[2]$ for $bp = 2$. The distortion $d[s][2]$, $0 \leq s \leq B[2]$, can be computed using Eqn.(6.7) with $T = 2$. Similarly, if

$$d[s][2] < M[s], \ \forall \ 0 \leq s \leq B[2], \tag{6.10}$$

the encoding process terminates for the current frame. Otherwise, the stopping sfb for the next bp, $B[3]$ is computed. In general, $B[bp]$ is computed as

$$d[s][bp-1] < M[s], \ \forall \ B[bp] < s \leq B[1] \tag{6.11}$$

where $1 < bp \leq b^M$, and b^M is the maximum bit-plane level for the whole frame.

It should be noted that if the encoding process does not terminate at $s = B[bp]$, the remaining sfbs from $s = B[bp] + 1$ to $B[1]$ in bp have to be encoded first before encoding of bit-plane $bp+1$ starts. This is to satisfy the condition that the encoded

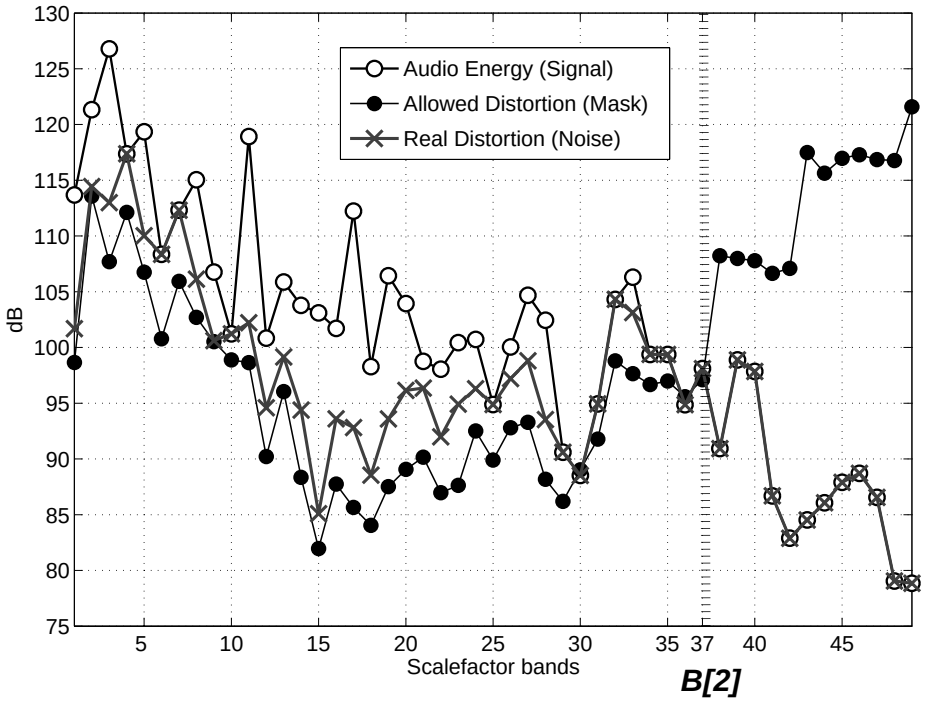


Figure 6.8: An example plot of the signal, mask and noise for one frame in ave-maria.wav which is encoded by MPEG-4 AAC at 64kbps.

format can be further enhanced to lossless.

6.4.3 Smart Enhancing Bitrate Estimation

From the encoding process proposed in Section 6.4.2, it can be observed that bitrate needed for smart enhancement roughly depends on two parameters which can be extracted from the original and the AAC coded signals. In this subsection, a bitrate estimation model is established. By using this model, the enhancing bitrate necessary for a low quality perceptual audio input to achieve transparent quality can be estimated without actual encoding. This bitrate can be used for SLS lossless format to perform a smart truncation (as indicated in Figure 6.5).

Supposing the smart enhancing process can be represented as a matrix (see Figure 6.6), the depth of the matrix can be estimated by the total residual energy

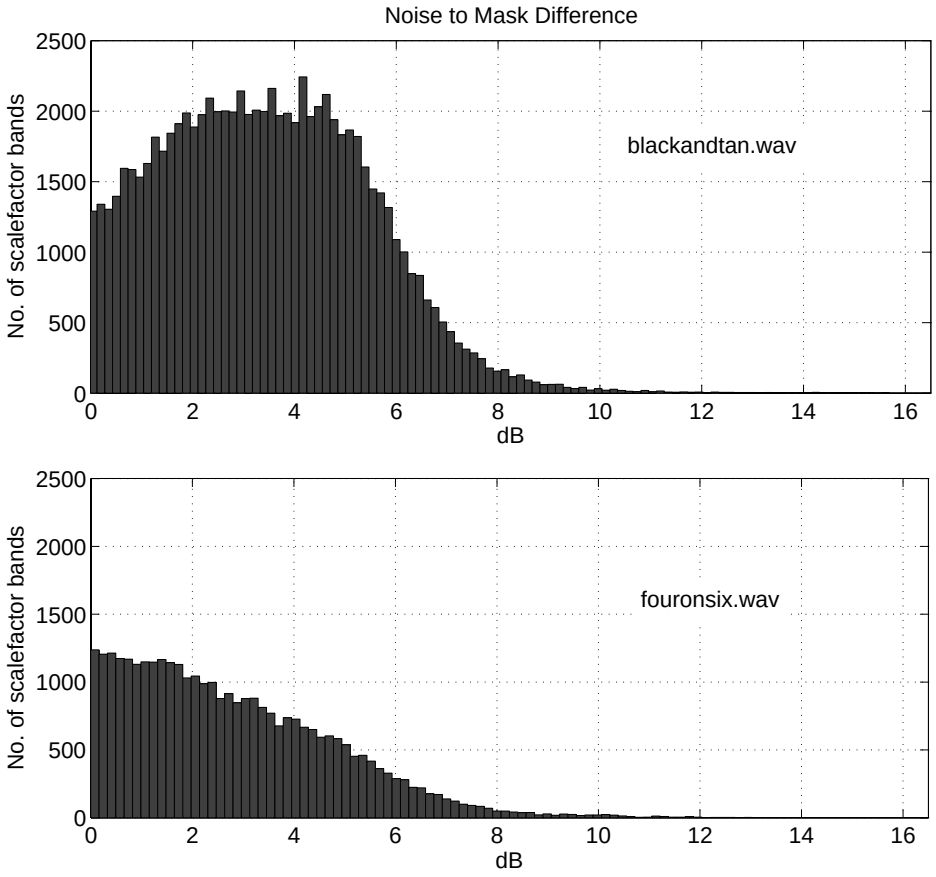


Figure 6.9: The noise to mask difference plot for two excerpts, blackandtan.wav and fouronsix.wav.

(or noise for perceptual coding) to mask difference and its length can be estimated by the percentage of low energy sfbs (position of $B[1]$). Specifically, the noise to mask difference, $D[s]$, for each sfb s in terms of dB can be simply computed as

$$D[s] = 10 \log_{10} \left(\frac{E[s]}{M[s]} \right). \tag{6.12}$$

$E[s]$, the residual signal energy for s is defined as

$$E[s] = 10 \log_{10} \left(\sum_{k=O[s]}^{O[s+1]-1} e[k]^2 \right) \tag{6.13}$$

where $e[k]$ is the residual coefficient as defined in Eqn.(2.2). An example plot on the histogram of noise to mask difference for two excerpts, blackandtan.wav and fouronsix.wav (48kHz/16bit), are shown in Figure 6.9. Only the positive values are shown. This is because that in the smart enhancing process, the residual signal has to be coded only if the residual signal is above the mask. Suppose that there are a total of F frames in an excerpt, the first parameter, D^T (in terms of 10^3dB), can be calculated as

$$D^T = \sum_{f=0}^{F-1} \sum_{s=0}^{S-1} D[s]/1000, \tag{6.14}$$

which is actually the area of the noise to mask difference histogram in Figure 6.9.

The second parameter, which is the length of the enhancing matrix, is the percentage of non-low energy sfbs in each frame. It is denoted by $P_{\bar{L}}$. From Figure 6.6, it can be observed that the enhancing bitrate is proportional to the number of non-low energy sfbs (from 0 to $B[1]$). Training on a large set of data shows that in general, an sfb is treated as "low energy" if the respective residual energy is lower than 20dB. Therefore, if $N_{\bar{L}}$ is the total number of sfbs with $E[s] \geq 20dB$ for a excerpt, then $P_{\bar{L}}$ can be computed as

$$P_{\bar{L}} = \frac{N_{\bar{L}}}{F \cdot S}. \tag{6.15}$$

Figure 6.10 shows an example plot of the residual signal energy for two excerpts, broadway.wav and cymbal.wav (48kHz/16bit). It can be seen that compared to broadway.wav, cymbal.wav has relatively larger percentage of low energy sfbs, which explains why cymbal.wav requires much less bitrate for smart enhancement (Figure 6.11 in Evaluation Section).

The total smart enhancing bitrate, R'_s (in terms of kbps), can thus be estimated

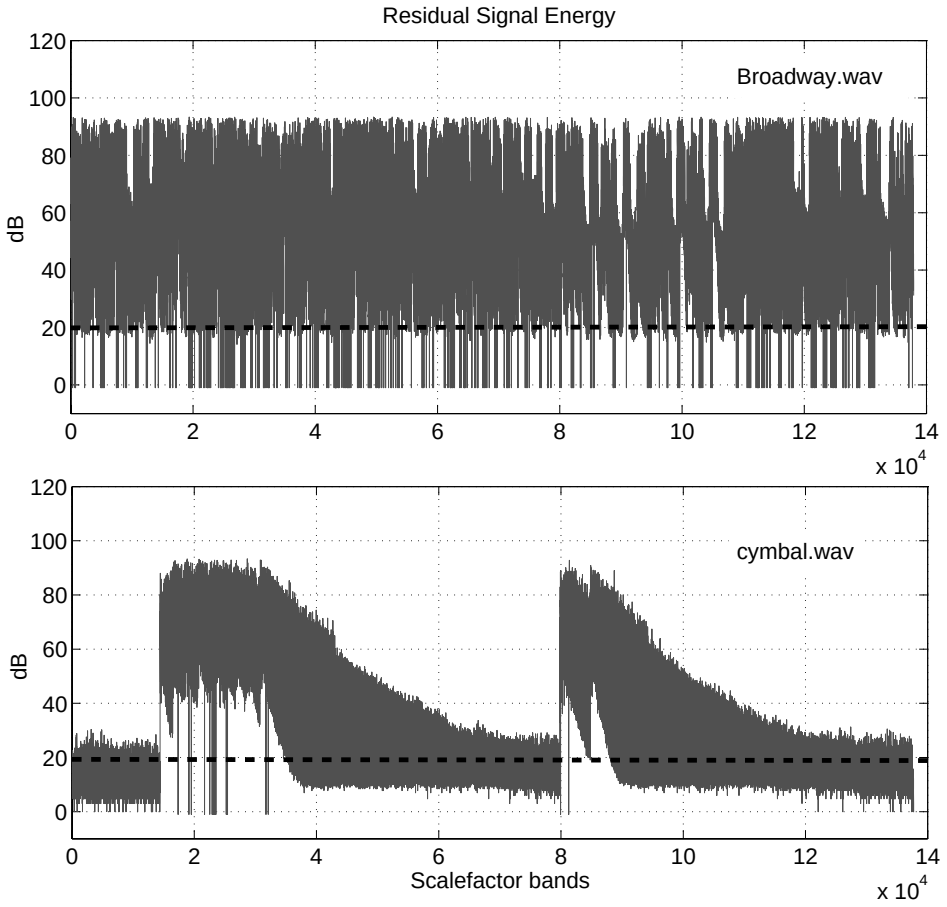


Figure 6.10: The residual signal energy plot for two excerpts, broaday.wav and cymbal.wav.

as

$$R'_s = \omega_1 \cdot (D^T \cdot P_{\bar{L}}) + \omega_2 \tag{6.16}$$

where ω_1 and ω_2 are the modulation constants. By substituting two sets of experimental data for R'_s and $D^T \cdot P_{\bar{L}}$, it is obtained that

$$\begin{cases} \omega_1 = 0.50 \\ \omega_2 = 132.20 \end{cases} . \tag{6.17}$$

By substituting (6.17) into (6.16), R'_s can be finally represented by

$$R'_s = 0.5 \cdot (D^T \cdot P_{\bar{L}}) + 132.2. \tag{6.18}$$

In this way, the smart enhancing bitrate can be estimated by analyzing the original and the perceptually encoded audio formats without a real enhancing process.

6.5 Performance Evaluation

Three sets of evaluation results, including the experimental smart enhancing bitrates, the estimated bitrates and the subjective quality evaluation, are presented in this section. The standard MPEG-4 audio test sequences including 15 stereo music files (with different genres) sampled at 48 kHz, 16 bits/sample as listed in Table 2.2 are used.

6.5.1 Experimental Smart Enhancing Bitrates

The experimental bitrate result is shown in Figure 6.11. Besides the bitrate at transparent quality, the bitrate required for AAC+LLE at different levels are also plotted for reference. Specifically, $bp = 1$ indicates the first bit-plane for each sfb is coded, and so on.

It can be seen from the Figure 6.11 that

- for different test items, the bitrate required for transparent quality varies. The maximum difference among the smart enhancing bitrates is around 140kbps.
- for most of the items, transparent quality can be achieved after the 2nd bit-plane is coded.

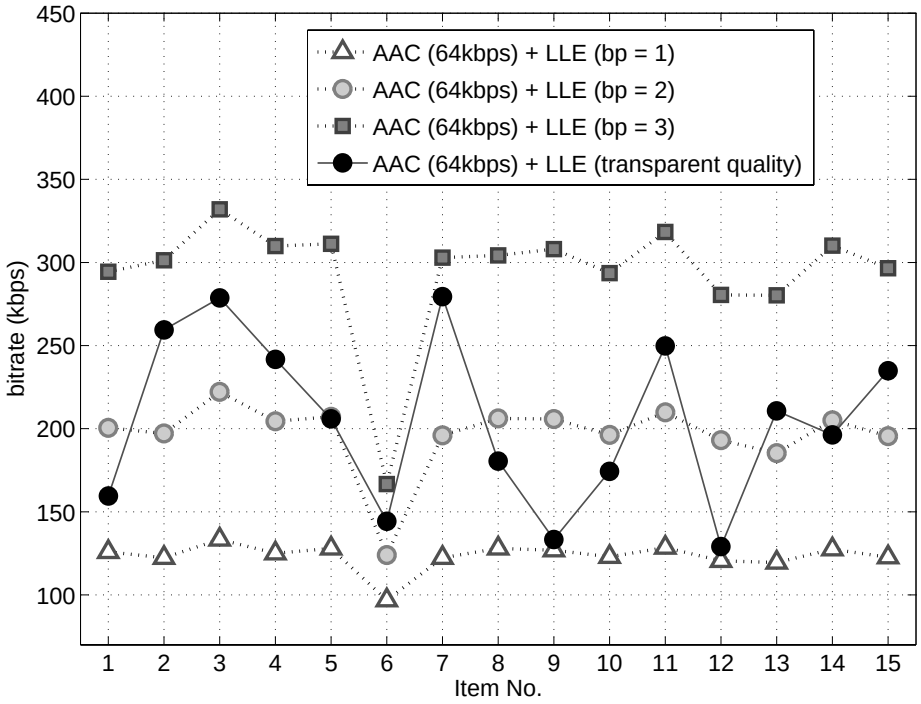


Figure 6.11: The SLS encoding bitrates for the 15 test excerpts.

- the average transparent bitrate for all the 15 items is 205.2kbps. According to Chapter 4, most of the test sequences can achieve transparent quality at 256kbps and 256kbps is thus the standard transparent bitrate setting in SLS music manager model. The average bitrate reduction compared to standard 256kbps is 19.8%, with the maximum bitrate reduction of 49.6% (item 12).

- for items 1, 6, 9 and 12, the bitrates required for transparent quality is relatively low. This may be caused by

  - the dynamic range of the signal energy is low.

  - the signal has low energy level in the high frequency bands.

  - the excerpt contains certain period of silence (very low energy) frames.

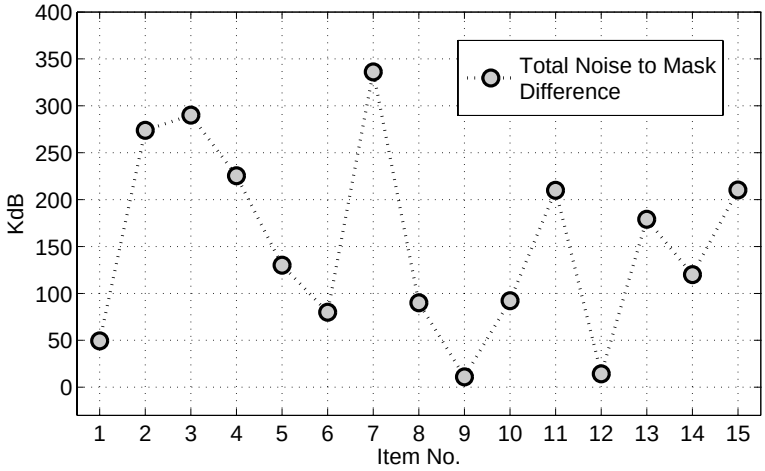


Figure 6.12: The total noise to mask difference (D^T) in terms of 10^3 dB for the 15 test excerpts.

6.5.2 Smart Enhancing Bitrate Estimation

As described in Section 6.4.3, the smart enhancing bitrate can be estimated using two parameters which comprises the total noise to mask difference, D^T, and the percentage of non-low energy sfbs, $P_{\bar{L}}$, using Eqn.(6.18).

Figure 6.12 shows the plot of the D^T values for all test sequences. It can be observed that the pattern roughly follows the experimental bitrate; however, it is still not accurate enough to perform the estimation. For example, the D^T value in Figure 6.12 for the 6th excerpt is higher than the 1st excerpt; however, in Figure 6.11 it behaves in the opposite way.

Another parameter, $P_{\bar{L}}$, for the 15 test excerpts are plotted in Figure 6.13. As mentioned in Section 6.4.3, the 6th excerpt (cymbal.wav) has large percentage of low-energy sfbs. This is caused by a long period of silence in this excerpt.

These two parameters are combined using Eqn.(6.6) to perform transparent bitrate estimation. The comparison between the experimental smart enhancing bitrate and the estimated bitrate is shown in Figure 6.14 and Table 6.1. It can

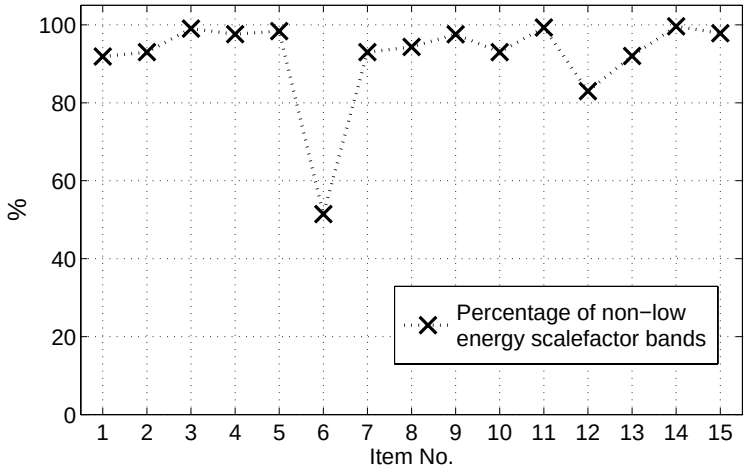


Figure 6.13: The percentage of non-low energy sfbs ($P_{\bar{L}}$) for the 15 test excerpts.

Table 6.1: Comparison between the experimental enhancing bitrates and the estimated bitrates (Diff = Experimental - Estimated).

Item	Experimental	Estimated	Diff	Item	Experimental	Estimated	Diff
1	159.47	154.84	4.63	9	133.33	137.57	-4.23
2	259.47	259.47	0.00	10	174.40	175.10	-0.70
3	278.67	275.71	2.95	11	249.87	236.43	13.44
4	241.60	242.15	-0.55	12	129.07	138.12	-9.06
5	205.87	196.11	9.76	13	210.67	218.52	-7.85
6	144.27	152.76	-8.49	14	196.27	191.93	4.34
7	279.47	288.35	-8.88	15	234.93	234.93	0.00
8	180.53	174.62	5.92				

be seen that the smart enhancing bitrate needed for transparent quality can be well estimated by using the two parameters extracted from the original and the AAC coded audio data without a real encoding process. The difference between the estimated and the experimental bitrates is less than 15 kbps for all the test sequences. Therefore, to ensure transparent quality, an additional 15 kbps should be added to all the estimated bitrates. The smart bitrates can thus be estimated for lossless SLS formats.

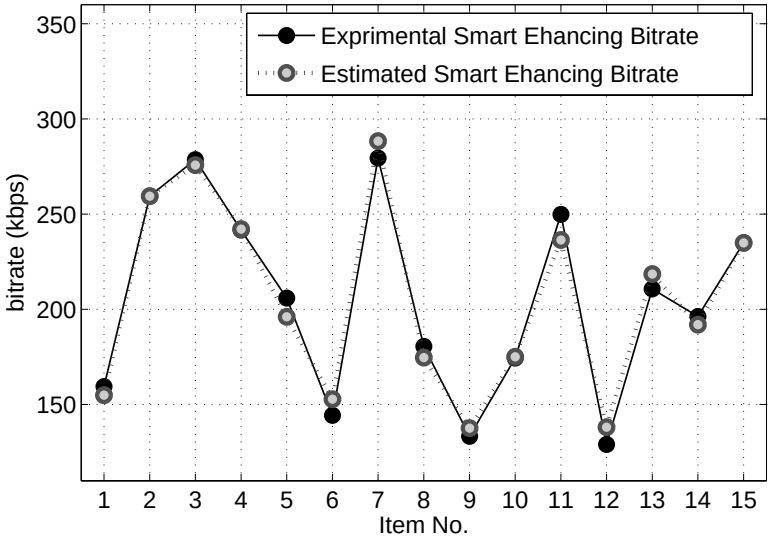


Figure 6.14: Comparison between the experimental enhancing bitrates and the estimated bitrates.

6.5.3 Subjective Quality

The MUSHRA [100] test for the subject qualities of the encoded sequences by using smart enhancement and fixed bitrate coding is shown in Figure 6.15 with 5% confidence level. It can be seen that for most of the items, smart enhancement brings a similar and near-transparent quality as the standard 256kbps does with a considerable amount of bitrate reduction. For those audio excerpts which require more than 256kbps, apparent improvements are achieved.

6.6 Conclusion

A smart enhancing structure based on MPEG-4 scalable lossless audio coding is proposed in this chapter. Evaluation results show that with the proposed encoding process, transparent quality audio can be achieved with an average bitrate reduction of approximately 20% compared to the fixed bitrate setting of 256kbps. Moreover, the bitrate estimation model proposed enables the smart truncation

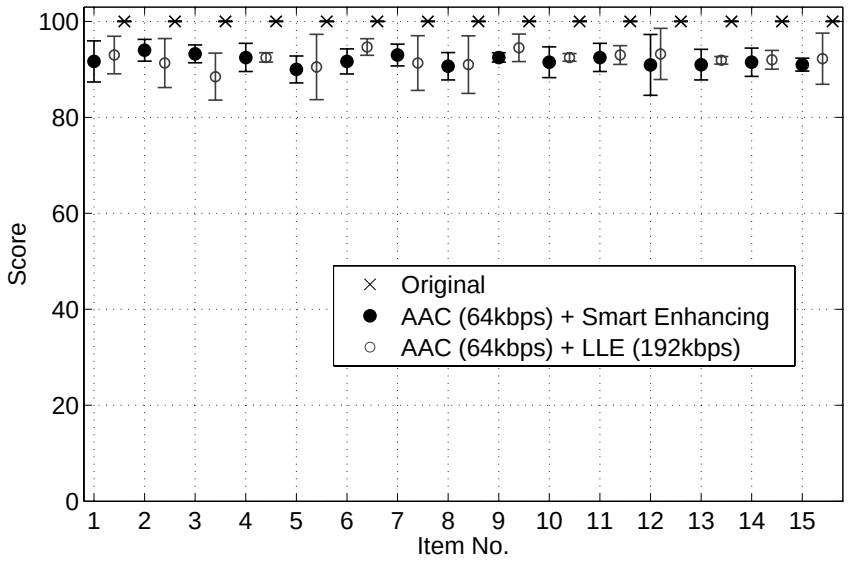


Figure 6.15: Comparison on subjective qualities for the smart enhancing and the fixed bitrate encoding.

from the lossless audio format to the transparent quality format without going through a real encoding process. The proposed functions enable the latest multi-quality SLS music manager to provide a transparent quality version of music with the minimum enhancing bitrate, while retaining the compatibility with the state-of-the-art perceptual audio format, lossless audio format and the ISO standardized SLS decoder.

Chapter 7

Conclusions and Recommendations For Future Research

In this chapter, a summary of conclusions and recommendations for future work for Chapters 3, 4, 5 and 6 are given in the first section. Section 2 gives a discussion on possible future work for several new applications.

7.1 Conclusions

The main contribution of this dissertation, Chapters 3-6, study and optimize the key components of the SLS codec structure, including the IntMDCT filterbank, M/S coding and bit-plane coding. The detailed conclusions for each chapter are summarized as follows.

- **IntMDCT in SLS**

  To enable efficient lossless coding, IntMDCT was adopted in MPEG-4 SLS. Notwithstanding the fact that MDCT is generally employed for lossy coding, SLS uses IntMDCT as the only filterbank for both lossy and lossless coding scenarios. With concerns about the potential problems which may be caused

by IntMDCT under perceptual audio coding, Chapter 3 conducts analysis and testings on the influence of the rounding errors introduced by IntMDCT under lossy operation. Based on the results, it is found that the rounding errors of IntMDCT will not degrade the perceptual quality of decoded audio under standard playback circumstances. It is therefore concluded that MDCT and IntMDCT filterbanks are interchangeable in a lossy coding scenario. This conclusion proves the validity of using only IntMDCT in scalable lossless audio coder.

- **Perceptually enhanced bit-plane coding**

  The lossy quality performance of the current SLS structure at low or non-core bitrates are shown to be inefficient compared to the performance at high core bitrates. Inspired by observations on the energy distribution of the residual spectrum, a frequency-region based prioritized bit-plane coding has been proposed along with an analysis on parameter optimization. Based on the statistical modelling of the spectrum, a much more simplified implementation is designed for SLS. The results of experiments show that the new SLS with low core bitrates is significantly improved by the proposed system with variable intermediate bitrate combinations. This is achieved with zero extra bit (as only one reserved bit is used to indicate the coding model) and merely trivial added complexity. This is especially important for the non-core mode of SLS as perceptually more efficient fully-scalable coding is achieved. The generalized frequency-region based prioritized bit-plane coding can also be applied in other bit-plane coding scenarios besides SLS.

- **Stereo bitrate allocation in SLS**

  Efficient stereo bitrate allocation algorithms are proposed to enhance the

quality of the non-core SLS or the fully scalable audio. These algorithms are specially designed for pure bit-plane coding with low complexity. Test results show that for sequences with relatively high correlations between the L and R channels, the proposed methods significantly enhance the M/S stereo coding scheme for fully scalable audio at various intermediate bitrates.

- **Smart enhancer for SLS**

  A smart enhancing structure based on MPEG-4 scalable lossless audio coding is proposed. Evaluation results show that with the proposed encoding process, transparent quality audio can be achieved with an average bitrate saving of 19.8% compared to the fixed bitrate setting of 256kbps. Moreover, the bitrate estimation model proposed enables the smart truncation from the lossless audio format to the transparent quality format without going through a real encoding process. The proposed functions enable the latest multi-quality SLS music manager to provide a transparent quality version of music with the minimum enhancing bitrate, while retaining the compatibility with the state-of-the-art perceptual audio format, lossless audio format and the ISO standardized SLS decoder.

7.2 Future Research

By looking at the potential applications of SLS, some suggestions for future research work are summarized as follows.

- **Perceptually enhanced bit-plane coding**

  In the work described in Chapter 4, the optimized parameter setting is based on the properties of the test sequences used. It will be our future work to

investigate how the parameter setting could be automated with any audio input.

- **SLS streaming**

  One of the most significant applications of SLS is streaming, where the advantage of FGS feature can be fully utilized. In an unstable network with variable capacity/speed, the best quality audio can be achieved by using SLS with FGS to adapt to the network conditions. However, as a relatively new codec, issues pertaining to how the SLS bitstream can be packed, error protected and transmitted in the optimized way have to be examined.

- **From fine granular scalability to medium granular scalability (MGS)**

  One application of SLS is to combine with MPEG *scalable video coding* (SVC) [101] to achieve scalable multimedia. Due to its target application, SVC only has two modes, MGS and *coarse granular scalability* (CGS). Moreover, only a limited number of layers are available. Therefore, in order to efficiently combine SLS with SVC, there is a need to extend FGS based SLS to the MGS level.

- **Full compatibility to any perceptual audio coders**

  Currently, SLS only supports AAC, scalable AAC and BSAC as the perceptual core. To be an unified enhancer for all perceptual audio formats, it is important for SLS to support another two commonly used audio formats, i.e., MP3 and *high-efficiency AAC* (HE-AAC). However, MP3 and HE-AAC use pseudo quadrature mirror filter instead of MDCT. Up to now, there is not a simple and efficient approach to map the spectrum of MP3 and HE-AAC to an MDCT spectrum in the compressed domain. Thus it is meaningful and interesting to explore the solution for this issue.

Author's Publications

Journals

1. T. Li, S. Rahardja and S.N. Koh, "Smart audio enhancer based on scalable lossless coding," *IEEE Transactions on Multimedia*, accepted and to be published in Apr. 2009.

2. T. Li, S. Rahardja and S.N. Koh, "Frequency region based prioritized bit-plane coding for scalable audio," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 1, pp. 94–105, Jan. 2008.

3. T. Li, S. Rahardja and S.N. Koh, "On IntMDCT for perceptual audio coding," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, No. 8, pp. 2236–2248, Nov. 2007.

4. T. Li, S. Rahardja and S.N. Koh, "Perceptually adaptive bit-plane coding for scalable audio," *IEE Electronics Letters*, Vol. 43, No. 1, pp. 60–61, Jan. 2007.

Conferences

1. T. Li and S. Rahardja, "Transparent bitrate estimation for perceptual audio coding," accepted in *International Conference on Industrial Electronics and Applications (ICIEA 09)*, Xi An, May 2009.

2. T. Li and S. Rahardja, "MPEG-4 scalable lossless audio transparent bitrate and its application," accepted in *International Conference on Acoustic, Speech and Signal Processing (ICASSP 09)*, Taiwan, Apr. 2009.

3. T. Li, S. Rahardja and S.N. Koh, "Efficient stereo bitrate allocation for fully scalable audio codec," in *IEEE International Workshop on Multimedia Signal Processing (MMSP 2008)*, pp. 921–926, Australia, Oct. 2008.

4. T. Li, L. Liew and S. Rahardja, "A multi-quality music managing system for internet music store," in *IEEE 9th International Conference on Signal Processing (ICSP 2008)*, pp. 2685–2688, China, Oct. 2008.

5. T. Li, S. Rahardja and S.N. Koh, "A fully scalable audio coding structure with embedded psychoacoustic model," in *International Conference on Acoustic, Speech and Signal Processing (ICASSP 08)*, pp. 205–208, USA, Apr. 2008.

6. T. Li, S. Rahardja and S.N. Koh, "Adaptive bit-plane scanning for scalable audio," in *IEEE International Workshop on Multimedia Signal Processing (MMSP 2007)*, pp. 31–34, Greece, Oct. 2007.

7. T. Li, S. Rahardja and S.N. Koh, "Perceptual enhancement for fully scalable audio," in *IEEE International Conference on Multimedia & Expo (ICME 2007)*, pp. 340–343, China, Jul. 2007.

8. T. Li, S. Rahardja and S.N. Koh, "Switchable bit-plane coding for high-definition advanced audio coding," in *the 13th International Multimedia Modelling Conference (MMM 2007)*, Singapore, Jan. 2007.

9. T. Li, S. Rahardja and S.N. Koh, "Perceptually prioritized bit-plane coding for high-definition advanced audio coding," in *IEEE International Symposium on Multimedia (ISM 2006)*, San USA, Dec. 2006.

10. T. Li, S. Rahardja and S.N. Koh, "Study on rounding errors of INTMDCT in perceptual audio coding," in *IEEE International Symposium on Multimedia (ISM 2005)*, USA, Dec. 2005.

Bibliography

[1] C. Todd, "A digital audio system for broadcast and prerecorded media," in *Proc. 75th Conv. Aud. Eng. Soc. (AES Conv.)*, Mar. 1984.

[2] E. Schroder and W. Voessing, "High quality digital audio encoding with 3.0 bits/sample using adaptive transform coding," in *Proc. 80th AES Conv.*, Mar. 1986.

[3] G. Theile, G. Stoll and M. Link, "Low-bit rate coding of high quality audio signals," in *Proc. 82nd AES Conv.*, Mar. 1987.

[4] K. Brandenburg, "OCFA new coding algorithm for high quality sound signals," in *Proc. Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP 1987)*, pp. 5.1.1–5.1.4, May 1987.

[5] J. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 314–323, Feb. 1988.

[6] W.Y. Chan and A. Gersho, "High fidelity audio transform coding with vector quantization," in *Proc. ICASSP 1990*, pp. 1109–1112, May 1999.

[7] K. Brandenburg and J. D. Johnston, "Second generation perceptual audio coding: The hybrid coder," in *Proc. 88th AES Conv.*, Mar. 1990.

[8] K. Brandenburg, J. Herre, J. D. Johnston, Y. Mahieux and E. Schroeder, "ASPEC: Adaptive spectral entropy coding of high quality music signals," in *Proc. 90th AES Conv.*, Feb. 1991.

[9] Y. F. Dehery, M. Lever and P. Urcun, "A MUSICAM source codec for digital audio broadcasting and storage," in *Proc. ICASSP 1991*, pp. 3605–3608, May 1991.

[10] M. Iwadare, A. Sugiyama, F. Hazu, A. Hirano and T. Nishitiani, "A 128 kb/s hi-fi audio codec based on adaptive transform coding with adaptive block size MDCT," *IEEE J. Select. Areas Commun.*, pp. 138–144, Jan. 1992.

[11] T. Painter and A. Spanias, "Perceptual coding of digital audio," in *Proc. of the IEEE*, vol. 88, no. 4, pp. 451–513, 2000.

[12] R. Mohan, J. Smith and C.S. Li, "Adapting multimedia internet content for universal access," *IEEE Transactions on Multimedia*, vol. 1, no. 1, pp. 104–114, 1999.

[13] "Call for proposals on MPEG-4 lossless audio coding," ISO/IEC JTC1/SC29/WG11 N5040, 2002.

[14] "Information Technology - coding of audiovisual objects, Part 3. Audio," ISO/IEC 14496-3, 1998.

[15] "Scalable lossless coding (SLS)," ISO/IEC 14496-3:2005/Amd 3, 2006.

[16] R. Geiger, T. Sporer, J. Koller and K. Brandenburg, "Audio coding based on integer transform," in *Proc. 111th Conv. AES*, New York, Sep. 2001.

[17] R. Yu, C.C. Ko, S. Rahardja and X. Lin, "Bit-plane golomb coding for sources with laplacian distributions," in *Proc. ICASSP 2003*, vol. 4, pp. 277–280, Apr. 2003.

[18] R. Geiger, Y. Yokotani and G. Schuller, "Improved integer transforms for lossless audio coding," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2003.

[19] Y. Yokotani, R. Geiger, G. Schuller, S. Oraintara and K. R. Rao, "Improved lossless audio coding using the noise-shaped IntMDCT," in *IEEE 11th DSP Workshop*, Aug. 2004.

[20] R. Yu, X. Lin, S. Rahardja, C. C. Ko and H. Huang, "Improving coding effciency for MPEG-4 audio scalable lossless coding," in *Proc. ICASSP 2005*, vol. 3, pp. 169–172, Mar. 2005.

[21] K. Brandenburg and B. Grill, "First ideas on scalable audio coding," in *Proc. 97th AES Conv.*, San Francisco, preprint 3924, 1994.

[22] B. Grill and K. Brandenburg, "A two or three-stage bit rate scalable audio coding system," in *Proc. 99th AES Conv.*, New York, preprint 4132, 1995.

[23] B. Grill, "A bit rate scalable perceptual coder for MPEG-4 audio," in *Proc. 103rd AES Conv.*, New York, preprint 4620, 1997.

[24] B. Grill and B. Teichmann, "Scalable joint stereo coding," in *Proc. 105th AES Conv.*, San Francisco, preprint 4851, 1998.

[25] N. Iwakami, T. Moriya and S. Miki, "High quality audio coding at less than 64kbps by using transform domain weighted iterleave vector quantization (TWIN-VQ) ," in *Proc. ICASSP 1995*, vol.2, pp. 3095–3098.

[26] J. Herre, E. Allamanche, K. Brandenburg, M. Dietz, B. Teichmann, and B. Grill, "The integrated filterbank based scalable MPEG-4 audio coder," in *Proc. 105th AES Conv.*, San Francisco, preprint 4810, 1998.

[27] A. Jin, T. Moriya, T. Norimatsu, M. Tsushima and T. Ishikawa "Scalable audio coder based on quantizer units of MDCT coefficients," in *Proc. ICASSP 1999*, vol. 2, pp. 897–900.

[28] M.S. Vinton and L.E. Atlas, "Scalable and progressive audio codec," in *Proc. ICASSP 2001*, pp. 3277–3280.

[29] S.H. Park, Y.B. Kim, Y.S. Seo, "Multi-layer bit-sliced bitrate scalable audio coding," in *Proc. 103rd AES Conv.*, New York, preprint 4520, 1997.

[30] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

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

[32] M. Purat and P. Noll, "Audio coding with a dynamic wavelet packet decomposition based on frequency-varying modulated lapped transforms," in *Proc. ICASSP 1996*, vol. 2, pp. 1021–1024.

[33] S. Boland and M. Deriche, "Audio coding using the wavelet packet transform and a combined scalar-vector quantization," in *Proc. ICASSP 1996*, vol. 2, pp. 1041–1044.

[34] D. Sinha and A. Tewfik, "Low bit rate transparent audio compression using adapted vavelets," *IEEE Trans. Signal Process.*, vol. 41, pp. 3463–3479, Dec. 1993.

[35] A. Aggarwal, V. Cuperman, K. Rose and A. Gersho, "Percetpual Zerotrees for Scalable Wavelet Coding of Wideband Audio," in *Proc. IEEE Workshop on Speech Coding*, pp. 16–18, Jun. 1999.

[36] Z. Lu and W.A. Pearlman, "A efficient, low complexity audio coder delivering multiple-levels of quality for interactive applications," in *Proc. IEEE Signal Process. Society Workshop on Multimedia Signal Process.*, pp. 529–534, Dec. 1998.

[37] C. Dunn, "Efficient audio coding with fine-grain scalability," in *Proc. 111th AES Conv.*, New York, preprint 5492, 2001.

[38] N.S. Jayant, J.D. Johnson and R. Safranek, "Signal compression based on model of human perception," in *Proc. of IEEE*, Vol. 81, No. 10, pp. 1385–1422, 1993.

[39] F.C. Chen and T.M. Chiu, "Scalefactor based bit shift FGS audio coding," in *Proc. of 19th Int. Conf. on Advanced Infomation Networking and Applications.*, vol. 2, pp. 235–238, Mar. 2005.

[40] J. Li, "Embedded audio coding (EAC) with implicit auditory masking," in *Proc. ACM on Multimedia*, Nice, pp. 592–601, Dec. 2002.

[41] M. Raad, A. Mertins and R. Burnett, "Scalable to Lossless Audio Compression based on Perceptual Set Partitioning in Hierarchical Trees (PSPIHT)," in *Proc. ICASSP 2003*, pp. 624–627.

[42] S. Strahl, H. Zhou and A. Mertins, "An adaptive tree-based progressive audio compression scheme," in *Proc. IEEE Workshop on App. of Signal Process. to Audio and Acoustics*, pp. 219–222, Oct. 2005.

[43] J. Johnston and A. Ferreira, "Sum-difference stereo transform coding," in *Proc. ICASSP 1992*, pp. 569–571.

[44] R. Yu, S. Rahardja, X. Lin and C.C. Ko, "A fine granular scalable to lossless audio coder," *IEEE Trans. Audio, Speech and Language Process.*, vol. 14, no. 4, pp. 1352–1363, Jul. 2006.

[45] A. Spanias, T. Painter and V. Atti, "Audio signal processing and coding," *Wiley-Interscience*, Feb. 2007.

[46] S.W. Golomb, "Run-lengh encodings," *IEEE Tran. Info. Theory*, vol. 12, pp. 399–401, Jul. 1966.

[47] L.H. Witten, R.M. Neal and J.G. Cleary, "Arithmetic coding for data compression," *comm. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[48] M. Nelson, "Arithmetic coding + statistical modelling = data compression," *Dr. Dobb's J.*, Feb. 1991.

[49] M. Hans and R.W. Schafer, "Lossless compression of digital audio", *IEEE sig. Proc. Magazine*, pp. 21–32, Jul. 2001.

[50] J. Princen, A. Johnson and A. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," in *Proc. ICASSP 1987*, vol. 12, pp. 2161–2164.

[51] H. S. Malvar, "Signal processing with lapped transforms," Artech House, 1992.

[52] P. Duhamel, Y. Mahieux and J. Petit, "A fast algorithm for the implementation of filter banks based on 'time domain aliasing cancellation," in *Proc. ICASSP 1991*, vol. 3, pp. 2209–2212.

[53] D. Sevic and M. Popovic, "A new efficient implementation of the oddly stacked princen-bradley filter bank," *IEEE Sig. Proc. Letters*, vol. 1, no. 11, pp. 166–168, 1994.

[54] R. Geiger, Y. Yokotani, G. Schuller and J. Herre, "Improved integer transforms using multi-dimensional lifting," in *Proc. ICASSP 2004*, vol. 2, pp. 1005–1008.

[55] H. Huang, R. Yu, X. Lin and S. Rahardja, "Method for realizing reversible integer type-IV discrete cosine transform," *Electronic Letters*, no. 8, vol. 40, pp. 514–515, 2004.

[56] H. Huang, S. Rahardja, R. Yu and X. Lin, "A fast algorithm of integer MDCT for lossless audio coding," in *Proc. ICASSP 1995*, vol. 4, pp. 177–180.

[57] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Tech. Rep., Bell Laboratories, Lucent Technologies*, 1996.

[58] F. Bruekers and A. Enden, "New networks for perfect inversion and perfect reconstruction," *IEEE JSAC*, vol. 10, no. 1, pp. 130–137, Jan. 1992.

[59] S. Oraintara, Y. J. Chen and T. Q. Nguyen, "Integer fast fourier transform," *IEEE Trans. Sig. Proc.*, vol. 50, pp. 607–618, 2002.

[60] J. Liang and T. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Sig. Proc.*, vol. 49, no. 12, pp. 3032–3044, 2001.

[61] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete fourier transform," *IEEE Trans. on Acoustics, Speech, Sig. Proc.*, vol. 32, no. 4, pp. 803–816, 1984.

[62] W. Givens, "Computation of plane unitary rotations transforming a general matrix to triangular form," J. SIAM 6(1), pp. 26–50, 1958.

[63] IA-32 Intel architecture optimization reference manual, ON: 248966-009, Intel Corp.

[64] "Verfication report on MPEG-4 SLS," ISO/IEC JTC1/SC29/WG11 N7687, Oct. 2005.

[65] OPERA voice/audio quality analyzer, http://www.peaq.org/.

[66] "Method for objective measurements of perceived audio quality," ITU-R BS. 1387.

[67] "User requirements for audio coding systems for digital broadcasting," Recommendation ITU-R BS. 1548-1.

[68] J. Koller, T. Sporer and K. Brandenburg, "Robust coding of high quality audio signals," in *Proc. 103rd AES Conv.*, New York, Sep. 1997.

[69] J. Koller, T. Sporer and K. Brandenburg, "Improving lossless audio coding," in *17th Int. Conf. AES*, Florence, Sep. 1999.

[70] M. Purat, T. Liebchen and P. Noll, "Lossless transform coding of audio signals," in *Proc. 102nd AES Conv.*, Munich, Mar. 1997.

[71] R. Geiger, J. Herre, J. Koller and K. Brandenburg, "IntMDCT - a link between perceptual and lossless audio coding," in *Proc. ICASSP 2002*, vol. 2, pp. 1913–1816.

[72] J. Li, "Low noise reversible MDCT (RMDCT) and its application in progresssive-to-lossless embedded audio coding," *IEEE Trans. Sig. Proc.*, vol. 53, pp. 1870–1880, May 2005.

[73] R. Geiger, J. Herre, G. Schuller and T. Sporer, "Fine grain scalable perceptual and lossless audio coding based on IntMDCT," in *Proc. ICASSP 2003*, vol. 5, pp. 445–448.

[74] "Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbits/s - Part 3: Audio," ISO/IEC 11172-3, 1992.

[75] B. Edler, J. Herre, K. Brandenburg and S. Quackenbush, "Revised core experiment methodology for MPEG-4 audio," ISO/IEC/JTC1/SC 29/WG11/N5722, Jul. 2003.

[76] R. Yu, X. Lin, S. Rahardja, C. C. Ko and H. Huang, "A scalable lossy to lossless audio coder for MPEG-4 lossless audio coding," in *Proc. ICASSP 2004*, vol. 3, pp. 1004–1007.

[77] C. Bouman and K. Sauer, "A generalized gaussian image model for edge-preserving map estimation," *IEEE Trans. Image Proc.*, vol. 2, no. 7, pp. 296–310, 1993.

[78] K. Brownlee, "Statistics Theory and Methodology in Science and Engineering," New York: Wiley, 1961.

[79] International Telecommunications Union, Radiocommunication Sector BS. 1284, "Methods for the Subjective Assessment of Sound Quality - General Requirements," Geneva, 1997.

[80] R. Yu, T. Li, and S. Rahardja, "Perceptually Ehanced Bit-Plane Coding for Scalable Audio," in *Proc. Int. Conf. Multimedia & Expo*, pp. 1153–1156, Jul. 2006.

[81] ITU-R BS.1116, "Methods for the subjective assessment of small impairments in audio systems including multi - channel sound system."

[82] ISO/IEC 11172-3, "Coding of moving pictures and associated audio for digital storage media at up to 1.5Mbit/s, Part 3. Audio", 1993.

[83] J. Johnston, "Sum-Difference stereo transform coding," in *Proc. ICASSP 1992*, pp. 569–571.

[84] J. Johnston, "Estimation of perceptual entropy using noise masking criteria," in *Proc. 89th AES Conv.*, 1990.

[85] C. Liu, W. Lee and Y. Hsiao, "M/S coding based on allocation entropy," *Digtal Audio Effects*, 2003.

[86] C. Liu, W. Lee and R. Hong, "Bandwidth proportional noise-shaping criterion and the associated fast bit allocation method for audio coding," *Digtal Audio Effects*, 2002.

[87] LAME, http://www.mp3dev.org/mp3/.

[88] "Sound Quality Assessment Material (SQAM) recordings for subjective tests," *Users' handbook for the EBU - SQAM Compact Disc*, 1988.

[89] www.soundbuzz.com/sls.

[90] H. Fletcher, "Auditory patterns," *Rev. Mod. Phys.*, pp. 47–65, Jan. 1940.

[91] D. Greenwood, "Critical bandwidth and the frequency coordinates of the Basilar membrane," *J. Acoust. Soc. Amer.*, pp. 1344–1356, Oct. 1961.

[92] J. Zwislocki, "Analysis of some auditory characteristics," in *Handbook of Mathematical Psychology*, R. Luce, R. Bush, and E. Galanter, Eds. New York: Wiley, 1965.

[93] B. Scharf, "Critical bands," in *Foundations of Modern Auditory Theory*. New York: Academic, 1970.

[94] R. Hellman, "Asymmetry of masking between noise and tone," *Percep. Psychphys.*, vol. 11, pp. 241–246, 1972.

[95] E. Zwicker and H. Fastl, "Psychoacoustics facts and models." Berlin, Germany: Springer-Verlag, 1990.

[96] E. Zwicker and U. Zwicker, "Audio engineering and psychoacoustics Matching signals to the final receiver, the human auditory system," *J. Audio Eng. Soc.*, pp. 115–126, Mar. 1991.

[97] M. Schroeder, B. Atal, and J. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *J. Acoust. Soc. Amer.*, pp. 1647–1652, Dec. 1979.

[98] J. Johnston, "Estimation of perceptual entropy using noise masking criteria," in *Proc. ICASSP 88*, pp. 2524–2527, May 1988.

[99] J. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 314–323, Feb. 1988.

[100] ITU-R BS.1354, "Method for the subjective assessment of intermediate quality levels of coding systems," 2003.

[101] "Information Technology-coding of audio-visual objects-part 10: advance video coding; Amendment 3 scalable video coding," ISO/IEC 14496-10:2005/AMD3.