國立交通大學

電子工程學系電子研究所

博士論文

用於先進音訊編碼之高效率編碼策略

Efficient Coding Strategies for Advanced Audio Coding

研究生：楊政翰

指導教授：杭學鳴

中華民國九十四年七月

用於先進音訊編碼之高效率編碼策略

Efficient Coding Strategies for Advanced Audio Coding

研究生：楊政翰　　Student：Cheng-Han Yang

指導教授：杭學鳴博士　　Advisor：Dr. Hsueh-Ming Hang

國立交通大學

電子工程學系電子研究所

博士論文

A Dissertation

Submitted to Department of Electronics Engineering

& Institute of Electronics

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Electronic Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

用於先進音訊編碼之高效率編碼策略

研究生 ： 楊政翰　　　　　　指導教授 ： 杭學鳴博士

國立交通大學 電子工程學系 電子研究所博士班

摘　　要

先進音訊編碼(Advanced Audio Coding, AAC)是 ISO/IEC MPEG 標準委員會所訂定新一代高效能而且複雜的音訊壓縮標準。由於先進音訊編碼器的設計並不在標準的規範內，因此編碼器中編碼模組的設計對編碼效能有很大的影響。而其中，一個適當的位元率–失真(rate-distortion)控制演算法就是能促成一個良好的先進音訊編碼器的關鍵要素。位元率–失真控制演算法和其相關的議題將會是本論文的重點。

籬柵圖(trellis-based)演算法是一個用於先進音訊編碼有名的位元率–失真控制演算法。它是利用籬柵圖搜尋整個訊框(frame)方法來找尋適當的編碼參數。籬柵圖演算法可以達到令人讚賞的效能，但是它的運算量非常之高。本論文的第一個貢獻是為先進音訊編碼設計了兩種型式的低複雜度、高效能的位元率–失真控制演算法，分別是串聯式籬柵圖(cascaded trellis-based, CTB)演算法以及增強型 BFOS (enhanced BFOS, EBFOS)演算法。在所提出的第一種型式演算法中(串聯式籬柵圖演算法)，我們把籬柵圖方法中一個運算量非常大的單一步驟分成兩個運算量較少的連續步驟來實現。藉由這種方式，我們有效的減低籬柵圖方法中的運算負擔。除此之外，我們可以藉由大幅減少籬柵圖搜尋的候選人這個方法，再更進一步減低運算量。在所提出的第二種型式演算法中(增強型 BFOS 演算法)，我們是以將位元一步一步分配到最被需要頻帶(band)的方法取代籬柵圖搜尋整個訊框方法。在這個方法中，頻帶層次(band-level)的位元使用效率以及先進音訊

編碼中編碼程序的頻帶間相互依賴性兩項我們都考慮到了。模擬結果顯示，我們所提出的兩種型式位元率–失真控制演算法的編碼效能明顯比 MPEG-4 先進音訊編碼驗證原型(Verification Model)來得好，而且很接近原始籬柵圖演算法的效能。此外，與籬柵圖演算法相比，所提出的演算法需要小於 1/140 的運算複雜度。

儘管現行的音訊編碼技術是這麼的成功，在減少多聲道音訊內的聲道間多餘資訊上卻沒有太多的成果。本論文的第二個貢獻是發展了一個用於知覺音訊編碼中用來移除聲道間多餘資訊(redundancy)的有效演算法。在我們的方法中，知覺比重聲道間預測技術被用在改良式離散餘弦轉換(Modified Discrete Cosine Transform, MDCT)係數上。以這個基本架構作為基礎，有兩種型式的預測器(predictor)被採用，分別是時間訊號式(time-signal based)預測器以及頻譜係數式(spectral-coefficient based)預測器。和 INT-DCT 方法相似，我們的方法並不需要額外的知覺遮罩控制，同時也並不會造成音訊品下降。另外，對於大多數典型的音訊訊號，我們的方法在減少位元率的效能上比 INT-DCT 方法來得好約 10%。

Efficient Coding Strategies for Advanced Audio Coding

Student: Cheng-Han Yang Advisor: Dr. Hsueh-Ming Hang

Department of Electronics Engineering and Institute of Electronics
National Chiao Tung University

ABSTRACT

The Advanced Audio Coding (AAC) is a recent, high performance and sophisticated audio coder specified by the ISO/IEC MPEG Standard Committee. Because the design of encoder in AAC standard is non-normative, the coding performance is greatly influenced by the design of the coding modules (tools) in an AAC encoder. One critical element contributing to a good AAC encoder is a properly designed rate-distortion (R-D) control algorithm. This and its related issues will be the focus of this dissertation.

One well-known R-D control algorithm designed for AAC is the trellis-based algorithm. It performs the trellis search through entire frame for finding proper coding parameters. It can achieve a praiseworthy performance, but their computational complexity is extremely high. The first contribution of this dissertation is the design of two types of low complexity and high performance rate-distortion control algorithms, which are Cascaded Trellis-Based (CTB) algorithm and Enhanced BFOS (EBFOS) algorithm. In the first type of the proposed algorithms, CTB, we efficiently reduce the computational burden of the trellis-based algorithms by splitting the heavy calculation stage in the trellis-based approach into two consecutive steps with much less computation. In addition, the complexity is further reduced by decreasing significantly the number of candidates in the trellis search. In the second type of proposed algorithms, EBFOS, instead of performing the trellis search through the entire frame, we allocate the bits to the most needed band step by step. In this approach, we consider both the “bit-use efficiency” at band-level and the inter-band dependency of the coding process in AAC. Simulation results show that the coding performance of the proposed two types of rate-distortion control algorithms is significantly better than that of the AAC Verification Model and is close to that of the original high-cost trellis-based algorithms. Roughly, the proposed algorithms require less than 1/140 complexity in computation when it is compared to the original trellis-based algorithms.

Despite the success of current audio coding techniques, little effort has been made to reduce the inter-channel redundancy inherent in multichannel audio compression. The second contribution of this dissertation is to develop an efficient algorithm for removing inter-channel redundancy in perceptual audio coding. In our approach, the perceptually weighted inter-channel prediction is applied to the Modified Discrete Cosine Transform (MDCT) coefficients. Based on this basic structure, two types of inter-channel predictor are proposed, the time-signal based predictor and the spectral-coefficient based predictor. Similar to the existing INT-DCT based approach, no extra perceptual masking control is needed for our approach; in the meanwhile, no audio quality degradation will be induced by our method. The bit rate reduction of our method is about 10% or higher than that of the INT-DCT based approach for most typical audio sequences.

誌謝

能完成論文，首先要感謝杭學鳴老師這兩年多來的悉心指導。杭老師不僅在做研究上給予我們指導，讓我在知識的探索上獲益良多，同時也關心我們的日常生活。在杭老師的指導下，我真的學習到很多很多東西。

感謝學長們的幫忙及指導，讓我能夠使用實驗室內充足的資源。感謝繼大、俊毅、郁男等同學，能和他們一同研究、討論及獲得他的幫忙，讓我在研究及生活上都受益良多。同時，身為通訊電子與訊號處理實驗室的一員，我也很感謝實驗室的所有成員們，以及所有曾給予我支持、鼓勵及幫忙的朋友和同學們。能在這個充滿活力及歡笑的實驗室作研究，讓我的研究生活一點也不枯燥。

謝謝我親愛的太太-子菁的陪伴與支持，讓我的求學生涯不孤單。有了她的關懷及照顧，我才能無後顧之憂的進行研究。最後，我要感謝我的父母。不論在生活上，求學上總是給予我最大的自由及支持。可以說沒有他們的鼓勵及支持，也就沒有我今天的成就。因此，謹以這本論文獻給我最親愛的家人以及所有我所感謝的人。

政翰

謹誌於國立交通大學

西元 2005 年七月

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

In the early 1980's, CD (Compact Disk) was developed. This new device induced a big revolution in audio industry – the storage media changes from tapes to CD, which also implies that the digital audio gradually replaces the analog audio. Moreover, to meet the demand of efficient transmission and storage of digital audio for diversified multimedia applications, many high-efficient audio coding schemes have been developed, such as MPEG-1/2/4 audio coding standards and Dolby AC-3 [1]. The MPEG-4 Advanced Audio Coding (AAC) is one of the most recent-generation audio coders specified by the ISO/IEC MPEG standards committee [2]. The core part of the MPEG-4 AAC is based on the MPEG-2 AAC technology. The MPEG-4 AAC features a number of additional coding tools and coder configurations comparing to MPEG-2 AAC [3][4]. Consequently, the MPEG-4 AAC is a very efficient audio compression algorithm aiming at a wide variety of different applications, such as internet, wireless, and digital broadcast arenas.

Two important issues of modern digital audio compression algorithms are a proper design of rate-distortion (R-D) control algorithm and an efficient design of inter-channel redundancy removal algorithm for multichannel audio signals. The first contribution of this dissertation is the design of two types of low complexity and high performance rate-distortion control algorithms, which are Cascaded Trellis-Based algorithm and Enhanced BFOS algorithm. The second contribution of this dissertation is to develop an efficient algorithm for removing inter-channel redundancy in perceptual audio coding.

1.1 Rate-Distortion Control

One critical element contributing to a good audio encoder is the rate-distortion (R-D) control process (or called bit allocation process) in the iteration loops. The rate-distortion control process in AAC is to determine two critical parameters, the values of scale factor (SF) and Huffman codebook (HCB), for each band so as to optimize the selected criterion under the given bit rate constraint. Because encoding these coding parameters is inter-band dependent in AAC, the proper choice of their values to maximize the coding performance becomes a difficult problem. Two-loop search (TLS) [5] is a commonly known R-D control algorithm, which is also used in the MPEG-4 AAC Verification Model (VM) [6]. VM is the encoder software developed by the MPEG committee to verify the coding syntax. However, as pointed out by [7] and [8], the poor choice of coding parameters in the TLS algorithm is one shortcoming of the current MPEG-4 AAC VM and, therefore, its compression efficiency is lower than expected particularly at low rates.

The generalized BFOS algorithm is an efficient bit allocation algorithm for subband coding and provides good performance while the inter-band dependency of coding process is ignored [9][10]. The research in [11] also shows that the generalized BFOS algorithm is a near optimal bit allocation scheme for MPEG-1 LayerⅠ/ LayerⅡ audio coding. However, our research shows that the generalized BFOS algorithm becomes less efficient for MPEG-4 AAC in which the inter-band dependency of coding process exists.

Two trellis-based high performance R-D control algorithms for AAC are proposed by [7][8]. One distinct feature of these R-D control algorithms, as comparing to TLS, is that both bit rate and distortion are controlled simultaneously and the inter-band relationship of coding parameters, SF and HCB, is also counted in choosing their values. These R-D control algorithms are formulated as Viterbi search through the trellis diagram [12][13] to find the optimal coding parameters and, therefore, are called *trellis-based optimization*. As discussed in [8], the subjective quality of the trellis-based optimization scheme is significantly better than that of TLS. However, its computational complexity is extremely high and thus it is not suitable for practical applications, such as real-time encoding with power constraint.

Therefore, it is very desirable to achieve a similar performance at a much lower complexity.

In this dissertation, two types of low complexity and high performance rate-distortion control algorithms are proposed for MPEG-4 AAC. In the first type of algorithms, two kinds of techniques are introduced to speed up the trellis-based optimization procedure. In the first kind of fast algorithms, we break the combined SF and HCB parameter selection stage into two sequential steps and thus call it *cascaded trellis-based* optimization. In the second kind of fast algorithms, by observing the audio signal characteristics and statistics we develop a few rules that can reduce significantly the number of candidates in the trellis. These two techniques are fairly independent. Together, the overall computational complexity is dramatically reduced while the coding performance degradation is small.

In the second type of proposed algorithms, instead of performing the trellis search through entire frame, we allocate the bits to the proper band step by step. In this approach, we both consider the “bit-use efficiency” at band-level and the inter-band dependency of coding process in AAC. The “bit-use efficiency” here means the distortion improvement due to receiving bits. The basic idea behind this approach is “give bits to the band with the maximum NMR gain per bit” or “retrieve bits from the band with the maximum bits per NMR loss”, which is similar to the basic concept of generalized BFOS algorithm. Otherwise, a fast version of this bit allocation scheme is also proposed for reducing calculations while the coding performance degradation is small.

1.2 Redundancy Inherent in Multichannel Audio

Despite the success of AC-3 and AAC, not much effort has been made to reduce the inter-channel redundancy inherent in multichannel audio. The only technique used in AC-3 or AAC is called "Joint Stereo Coding", which consists of Intensity/Coupling and Mid/Side (M/S) stereo coding. Intensity/Coupling is adopted based on the psychoacoustic evidence that the perception of high-frequency sound component (above approximately 2kHz) mainly relies on the energy-time envelopes. Instead of encoding the original signals, the M/S stereo coding encodes the sun (middle) and the difference (side) of the signals of a channel-pair. However, this may cause audio quality degradation if a proper perceptual masking control for the sum and difference signals is not in place.

In addition to M/S technique, several other algorithms, such as the Karhunen-Loeve Transform (KLT) based approach [14] and the Integer-to-Integer Discrete Cosine Transform (INT-DCT) based approach [15], have been suggested for removing inter-channel redundancy. As shown in Fig. 1.1, in the KLT based approach, KLT is applied to the “un-quantized” MDCT coefficients of multiple channels. However, as discussed in [15], how to derive proper perceptual masking thresholds for the KLT-transformed signals is a challenge yet to be solved. Without appropriate perceptual model in the transform domain, this approach may result in uncontrolled degradation in audio quality. In contrast, the INT-DCT based approach conducts a lossless transformation on the “quantized” MDCT coefficients of multiple channels. Because the quantized MDCT coefficients can then be perfectly reconstructed, there is no quality loss in using the INT-DCT based approach but its ability in removing redundancy is relatively low.

In order to achieve a higher efficiency in removing the inter-channel redundancy and, in the meanwhile, to maintain good audio quality, an efficient inter-channel prediction algorithm is proposed in this paper. Different from the M/S stereo coding or the KLT based approach, our approach does not propagate the quantization noise from one channel to other channel. Therefore, no extra perceptual masking control is needed. Similar to the INT-DCT based approach, no audio quality degradation will be induced by our method. Moreover, the bit rate reduction performance of our new approach is 10% better than that of the INT-DCT based approach on the average.

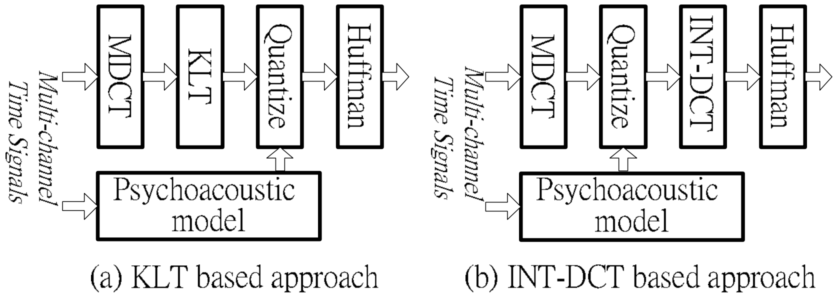


Fig. 1.1: Simplified block diagram of (a) KLT based approach. (b) INT-DCT based approach.

1.3 Outline of the Dissertation

The organization of this dissertation is as follows. In Chapter 2, we introduce the brief history of MPEG Audio and describe the main feature in MPEG-4 AAC. The proposed first type of R-D control algorithms, cascaded trellis-based scheme, and its variations are described in Chapter 3. In Chapter 4, we describe the proposed second type of R-D control algorithms and its fast version. The proposed efficient inter-channel prediction scheme for removing inter-channel redundancy is described in Chapter 5. Finally, all main results achieved in this dissertation are summarized in Chapter 6.

Chapter 2
Advanced Audio Coding

In this chapter, we will briefly describe the history of MPEG Audio and MPEG-2/MPEG-4 AAC standards. Moreover, we will describe the main features in MPEG-4 AAC system, which are mainly based on MPEG-4 AAC Version 1 [2].

2.1 Brief History of MPEG Audio

The standardization body **M**oving **P**icture **E**xpert **G**roup (MPEG) was established in 1988 to specify digital video and audio coding schemes. MPEG provided the first digital audio compression standard, MPEG-1 audio layers I, II and III, which was adapted by the International Organization for Standards and the International Electrotechnical Commission (ISO/IEC) at the end of 1992 [16]. MPEG-1 audio coding system only supports single-channel or two-channel coding with sampling rate at 32, 44.1, and 48 kHz.

Following the first phase of its standardization efforts, MPEG-1 audio standard was complemented by MPEG-2 audio in 1994 [17], which provided a backward compatible (BC) multichannel coding and extensions to lower sampling rates. The basic coding technology in MPEG-2 BC is the same as that in MPEG-1 audio. For Layers I and II, the time to frequency mapping is implemented by a 512-tap polyphase quadrature mirror filter (PQMF) with 32 channels [18]. The masking thresholds are computed by Psychoacoutsic Model 1 with a 512-point FFT. The quantization is performed by block companding groups. For increasing the frequency resolution and coding efficiency, Layer III employs a hybrid filter bank composed by the PQMF followed by a modified cosine transform (MDCT) [19]. The masking thresholds are computed by Psychoacoutsic Model 2 with a 1024-point FFT. The quantization is performed by employing non-uniform quantization followed by Huffman coding. The performance and implementation complexity rises as the layer number goes up. Layer II is widely used for Digital Audio Broadcast (DAB) in Europe, audio in Video Compact Disc (VCD), and broadcast delivery systems. Layer III has become the most popular standard for transmission and storage of compressed audio for both internet (World Wide Web, WWW) and consumer electronics (such as handheld media applications).

For better coding efficiency for multichannel coding, MPEG began standardization activities for a non-backward compatible (with MPEG-1) audio coding system. As a result of this effort, the IS13818-7 MPEG-2 Non-backward Compatible/Advanced Audio Coding (NBC/AAC) algorithm was finalized in 1997 April [20]. The coding efficiency of MPEG-2 AAC is greatly improved and only needs data rates 384 kbps for five full bandwidth channels with transparent quality. In parallel, the MPEG-4 standardization activities had been started in 1994/1995. Version 1 of the MPEG-4 audio standard was adopted in December of 1998. MPEG-4 audio encompasses more functionality than just perceptual coding. It contains an integrated family of algorithms with wide ranging provisions for scaleable, object-based speech and audio coding. Relative to its predecessors, the distinguishing features of MPEG-4 are extensive scalability, object-based representations, user interactivity manipulation, and a comprehensive set of coding tools. Very low bit rates can be achieved by using the structured representations for synthetic speech and music. For higher bit rates and “natural audio” speech and music, the standard provides integrated coding tools. The coding tools are specified in terms of MPEG-4 “profiles” which essentially recommend tool sets for a given level of functionality and complexity. Version 2 of the MPEG-4 audio [21], which is fully backward compatible to Version 1, was technically frozen in December 1999. Version 2 contains various extensions to Version 1, such as Error Robustness, Low-Delay AAC (AAC-LD), Bit Slice Arithmetic Coding (BSAC) for fine grain bit rate scalability and Harmonic and Individual Lines plus Noise (HILN) in stand alone mode as well as in combination with the parametric speech coding scheme HXVC.

2.2 Overview of MPEG-2/MPEG-4 AAC

MPEG-2 AAC development officially started from 1994 and was finalized in 1997. The goal of MPEG-2 AAC development is a new powerful state-of-the-art multichannel coder without compatibility constraints. Like most digital audio coding schemes, MPEG-2 AAC algorithm compresses audio signals by removing the redundancy between samples and the irrelevancy within audio signals. Removing the redundancy between samples is accomplished by time-frequency analysis. Making use of the acoustic masking properties (perceptual model) of human hearing system to remove unperceivable signal parts (irrelevancy removal). MPEG-2 AAC combines the coding efficiencies of a high-resolution filter bank, backward-adaptive prediction, joint stereo coding, and Huffman coding with a flexible coding architecture to permit application-specific functionality. Because of the high performance of the MPEG-2 AAC, it was adopted as the core of the MPEG-4 General Audio (GA) Standard.

MPEG-4 GA is built based on the core kernel provided by MPEG-2 AAC, which is extended by several additional coding tools and coder configurations. Therefore, MPEG-4 GA is usually called MPEG-4 AAC. These additional coding tools are the Perceptual Noise Substitution (PNS), the Long-Term Prediction (LTP) and the Transform-domain Weighted Interleave Vector Quantization (TwinVQ). The PNS and LTP tools are available to enhance the coding performance for the noise-like and tonal signals. The TwinVQ tool is a special coder kernel to cover extremely low bit rates.

In order to allow the tradeoff between quality and complexity, MPEG-2 AAC system offers three profiles: the Main Profile, the Low Complexity (LC) Profile, and the Scalable Sampling Rate (SSR) Profile. Each profile contains a subset of a single toolset, with the tools among it to compose the coder. Similarly, the profiles in MPEG-4 mean subsets of toolset but with more toolsets supported. That is to say, a profile may not be another level of implementation, but may be another toolset. One should note that, once the implementation level is defined, the corresponding complexity will be known.

MPEG-4 AAC takes the word ‘tool’ to represent a specific function, and the combination of tools constructs the entire coding. There are two tools belong to quantization in MPEG-4 GA – one is quantization for conventional AAC; the other is interleaved vector quantization. With these two quantization tools, MPEG-4 GA can be divided into two parts of coding schemes – AAC and TwinVQ. We are focus on MPEG-4 AAC in our dissertation. Fig. 2.1 gives an overview of the MPEG-4 AAC encoder and the detail of each tool is described in the following subsections.

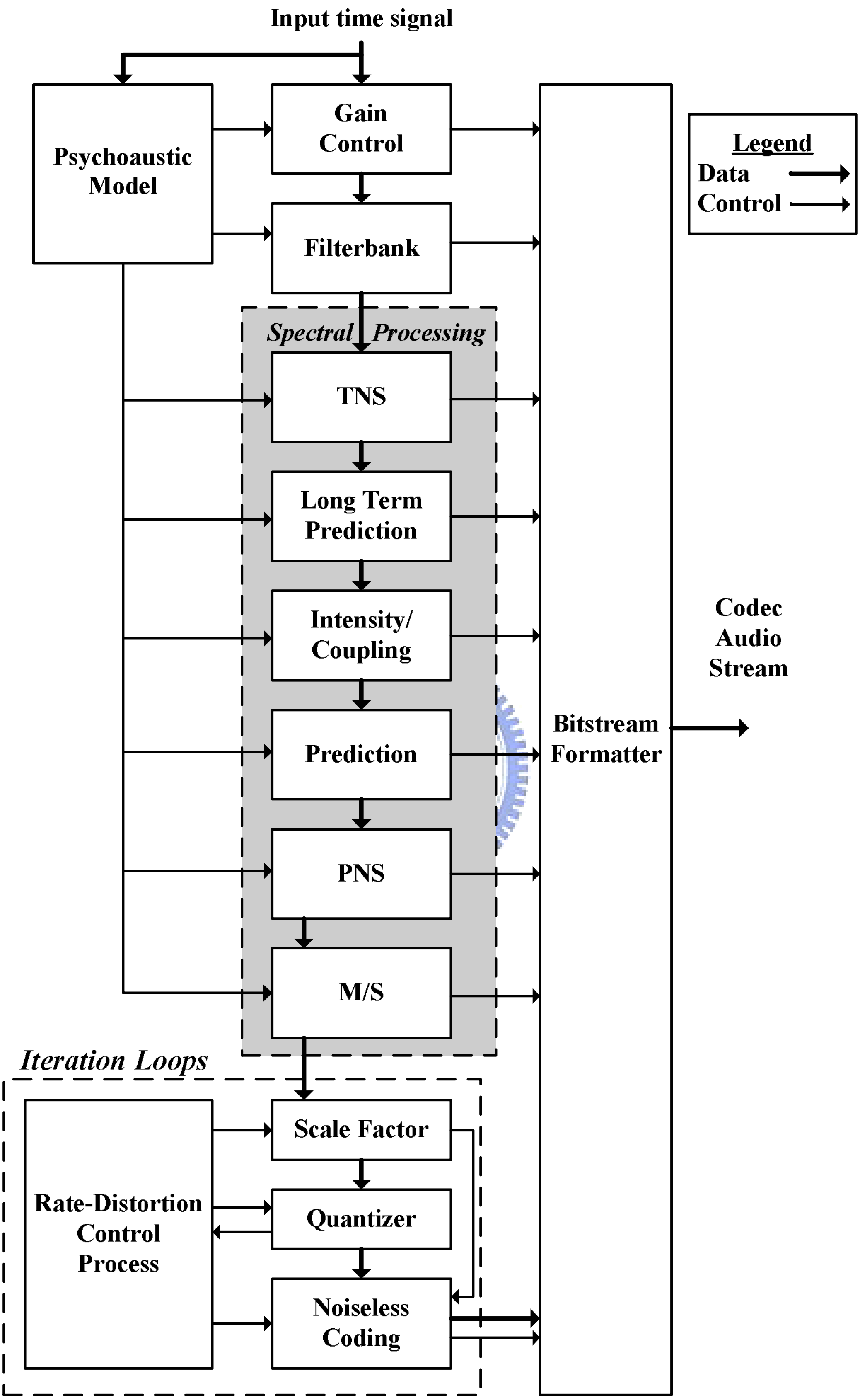


Fig. 2.1: MPEG-4 AAC Encoder Block Diagram. [2]

2.3 Psychoaustic Model

Most current audio coders achieve compression by exploiting the fact that "irrelevant" signal is not detectable by human ear. Thus, psychoacoustics plays an important role in reducing the bit rate in audio coding. Psychoacoustic model is basically based on the critical band and acoustic masking principle of the human ear/brain aural perception system. The irrelevant information is identified during signal analysis incorporating with several acoustic masking principles, such as the absolute threshold of hearing, simultaneous masking and temporal masking. In this section, we will first illustrate the critical band and acoustic masking principle of the human aural perception system. Then we will introduce an example of psychoacoustic model in MPEG-4 AAC, of course, it is not normative.

2.3.1 Critical Band

In the physic measurement of sound, the unit of the frequency is hertz (Hz). Nonetheless the actual unit of frequency represented in inner ear is another nonlinear frequency scale. In psychoaustics, the unit to represent the frequency of a signal is ‘bark’. This unit comes from the critical band phenomenon of human ear. Generally, hearing system of human is composed of several sensors, with each sensor detecting the signals in different frequency range. These sensors can be model as several overlapping bandpass filters. These bandpass filters, which are also called critical bands, are successive arranged about 2 ~ 22k Hz with different central frequency and bandwidth. Table 2.1 lists the frequency range of critical bands. One can easily find that the bandwidths of critical bands are not the same.

The critical band concept strongly influences the masking effect. Phenomenon shows that, if a tone and noise are within the same critical band, the masking effect will occur, even if they do not intersect. Moreover, within the same critical band, the threshold of masking effect will be almost the same. Otherwise, the masking effect also exists outside the critical band because of the spreading effect. However, the influence would greatly decay according to the distance.

Critical band concept greatly influences the computation of perceptual model, since it is the actual sound representation of inner ear. In the computation of perceptual model, the frequency domain signals should be transformed to bark domain first. After that, all the other calculations would be performed in bark domain.

Table 2.1: Critical bands bandwidth. [1]

Band No.	Central Freq. (Hz)	Bandwidth (Hz)	Band No.	Central Freq. (Hz)	Bandwidth (Hz)
1	50	0 ~ 100	14	2150	2000 ~ 2320
2	150	100 ~ 200	15	2500	2320 ~ 2700
3	250	200 ~ 300	16	2900	2700 ~ 3150
4	350	300 ~ 400	17	3400	3150 ~ 3700
5	450	400 ~ 510	18	4000	3700 ~ 4400
6	570	510 ~ 630	19	4800	4400 ~ 5300
7	700	630 ~ 770	20	5800	5300 ~ 6400
8	840	770 ~ 920	21	7000	6400 ~ 7700
9	10000	920 ~ 1080	22	8500	7700 ~ 9500
10	1175	1080 ~ 1270	23	10500	9500 ~ 12000
11	1370	1270 ~ 1480	24	13500	12000 ~ 15500
12	1600	1480 ~ 1720	25	19500	15500 ~
13	1850	1720 ~ 2000			

2.3.2 Acoustic Masking Principle

Acoustic masking is an important characteristic of human ear for compression and there are three major masking principles, absolute threshold of hearing, simultaneous masking and temporal masking.

Absolute threshold of hearing, also called threshold in quiet, is the threshold that a signal is just perceivable. In other words, a signal is inaudible if the power density of it is less than the threshold. The unit of absolute threshold is SPL (sound pressure level), which is the unit that how large a sound is. An example of the curve of threshold in quiet is shown in Fig. 2.2.

The masking phenomenon is that one signal (called maskee) is masked by another signal (called masker). If the masking effect works in frequency domain, it is called simultaneous masking. If the masking effect works in time domain, it is called temporal masking. A typical example of simultaneous masking is shown in Fig. 2.2. When a signal (masker) is present at a particular frequency, you cannot perceive the signal (maskee) at nearby frequencies that are sufficiently low in intensity. For the temporal masking, if the intensity of a signal at a time is larger than that at neighboring time, it may mask the signals near it. There are two categories of temporal masking according to whether the masking effect is prior or posterior to the masker at time domain. The masking effect prior to the masker time is called pre-masking; in other case, the masking effect posterior to the masker time is called post-masking. Fig. 2.3 shows an example of temporal masking.

By exploiting the masking effects, we can remove the inaudible signals to reduce the signals that is necessary to be encoded. This concept is also extended to the idea of “maximum allowable energy level” for coding distortion. This idea means that we can neglect the quantization error if this error is lower than the absolute threshold or the masking threshold. Therefore, we could have the chance of transparent coding even if we perform quantization in Encoder.

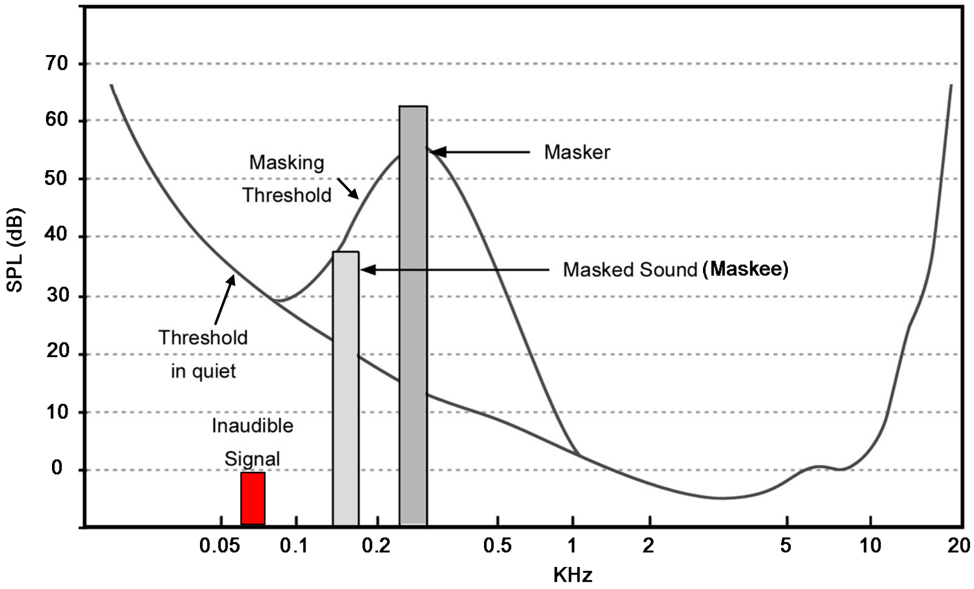


Fig. 2.2: Simultaneous masking effect. The masker at 300 Hz masks the 150 Hz signal.

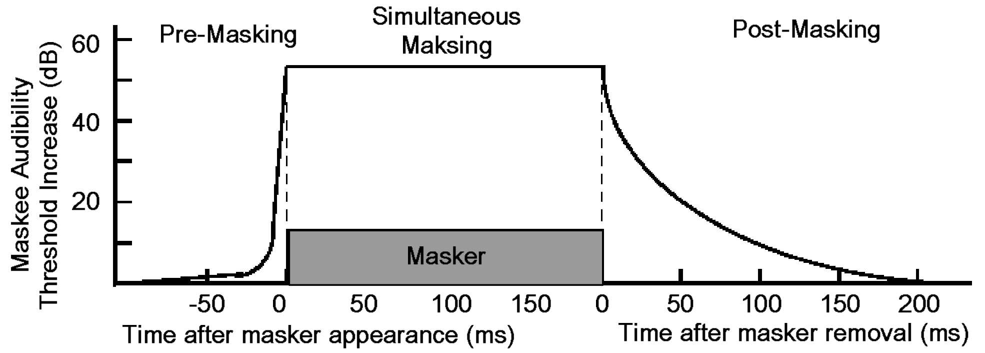


Fig. 2.3: Temporal masking effect. [1]

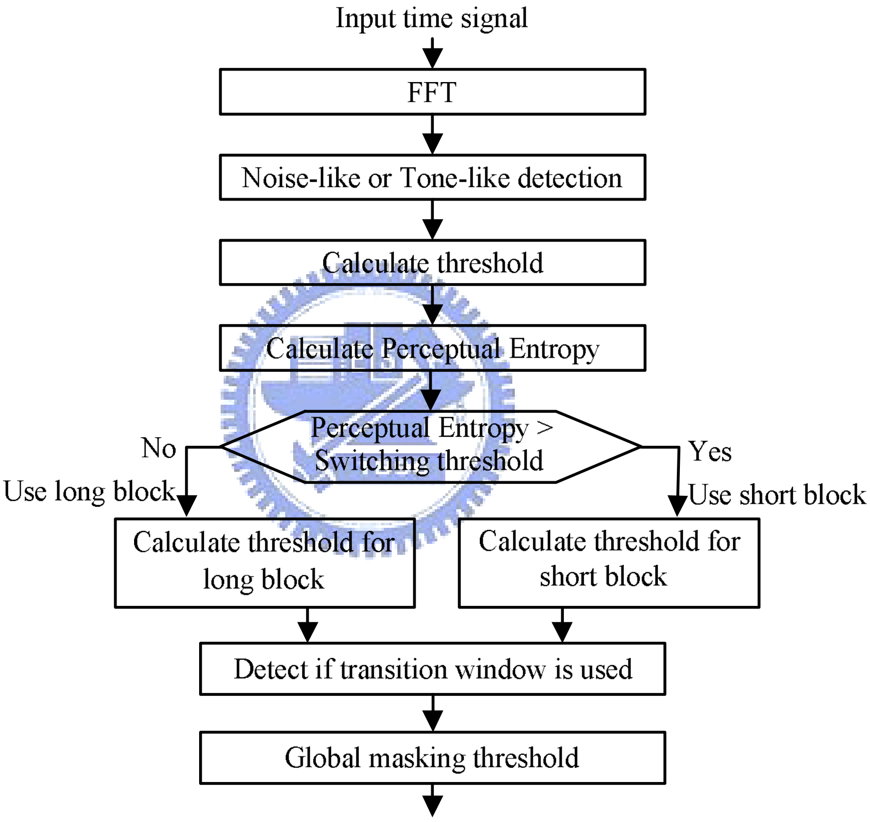


Fig. 2.4: Block diagram of a typical Psychoacoustic Model. [2]

2.3.3 Psychoacoustic Model in MPEG-4 AAC

Fig. 2.4 shows the block diagram of psychoacoustic model. Frequency domain transformation is essential since representation of signal in inner ear bases on frequency domain. Because the masking effects of noise-like and noise-like signals are much different, the detection of noise-like signal or tone-like signal is performed at second procedure. According to these two sorts of masking effect, the initial masking threshold is calculated. Perceptual entropy calculated from initial masking threshold decides which window is suitable. It also can be applied to quantization tool to control the number of bits used in this frame. After deciding the widow type, we calculate the actual masking threshold by corresponding window type. Note that, the transition window between long/short windows is necessary for smooth coding, so a procedure is essential to detect if the transition window is needed.

2.4 Filterbank

A fundamental component in AAC is the conversion of time-domain signals into frequency representation (spectral coefficients) and is done by a forward modified discrete cosine transform (MDCT) in the encoder. Its reverse process, inverse modified discrete cosine transform (IMDCT), is done in the decoder. Moreover, the MDCT and IMDCT employ a technique called time domain aliasing cancellation (TDAC) [19].

For reducing the boundary effect, each block of input samples is overlapped by 50% with the preceding block and the following block. The overlapped analysis and overlap-add synthesis processes are illustrated in Fig. 2.5.

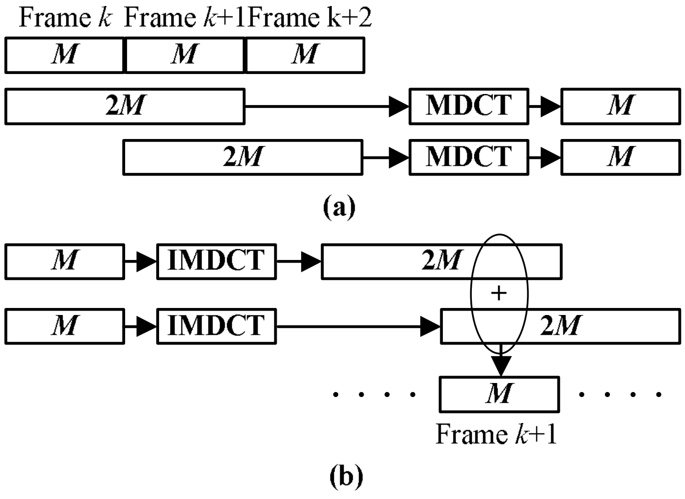
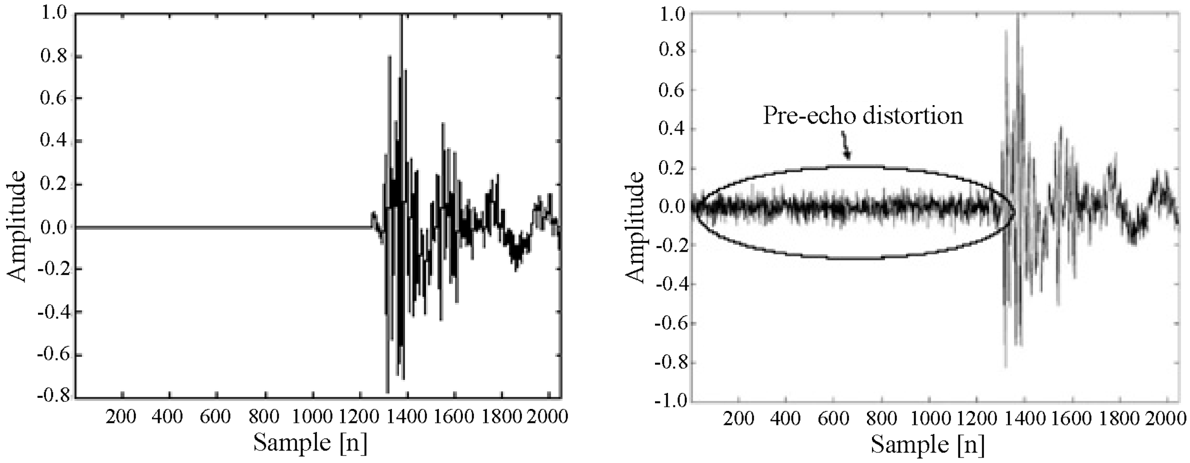


Fig. 2.5: Overlapping phenomenon of (a) MDCT and (b) IMDCT. [1]

2.5 Temporal Noise Shaping (TNS)

The handling of transient and pitched signals is a major challenge in the perceptual audio coders. This is mainly due to the problem of maintaining the masking effect in the reproduced audio signals. While coding the transient signals, a so-called pre-echo problem may occur because of the temporal mismatch between masking threshold and quantization noise. Although the quantization noise that stretches out the entire window is small compared to the abrupt signal region, it is much larger than quiet or nearly quiet signal region. Fig. 2.6 illustrates this phenomenon.

In AAC, the TNS tool is designed to solve this problem by controlling the temporal shape of the quantization noise within each window of the transform. Basically, the TNS technique is based on two main considerations, 1.Consideration of the time/frequency duality and 2.Shaping of quantization noise spectra by means of open-loop predictive coding [8]. In addition to pre-echo protection, TNS permits for a better encoding of pitch-based signals, such as speech.



(a) Uncoded signal

(b) Decoded signal

Fig. 2.6: Pre-echo example. [1]

2.6 Prediction/Long Term Prediction (LTP)

Prediction is used to further remove the redundancy especially for more or less stationary signals. The backward adaptive predictors are adopted so that no additional side information is required. A second order backward adaptive lattice structure predictor is used and the predictor parameters are adapted to the current signal statistics, using an LMS-based adaptation algorithm.

Long term prediction is a well-known technique from speech coding and has been used to exploit the long-term correlation in the speech signals. It is adopted as a new tool in MPEG-4 AAC to improve the coding efficiency. LTP exploits time redundancy between the current and the preceding frame (backward prediction). First, the quantized spectral coefficients of the preceding frame are mapped (using IMDCT) into the time signals and are matched to the current time signals to get the best prediction parameters, delay and gain. Then, the predicted time signals are constructed and mapped into the spectral representations (using MDCT). Finally, the predicted coefficients and the current coefficients are subtracted from each other to get the residual coefficients. The encoding process of LTP is shown in Fig. 2.7. LTP tool provides considerable coding gain for stationary harmonic signals as well as some non-harmonic tonal signals. Besides, because the complexity of LTP tool is much lower compared to Prediction tool, LTP replaces Prediction tool in MPEG-4 AAC.

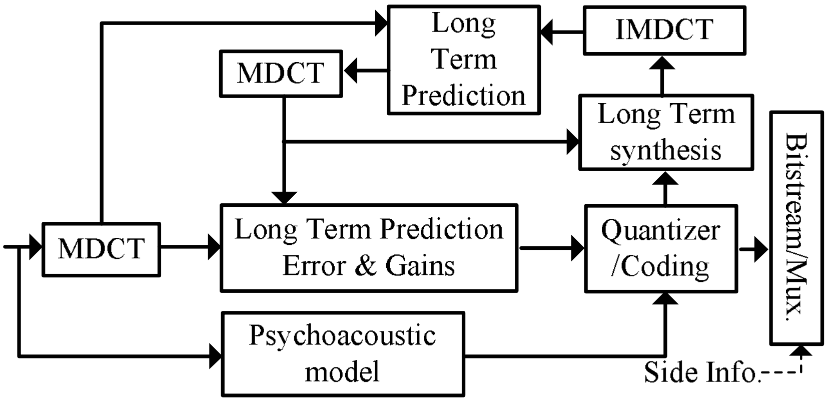


Fig. 2.7: Block diagram of the encoding process of LTP. [2]

2.7 Joint Stereo Coding

AAC joint stereo coding reduces the needed bit rate for stereo or multichannel signals more efficiently than separate coding of several channels. There are two different joint stereo coding approaches in AAC: M/S (middle/side) stereo coding and Intensity/Coupling.

Instead of encoding the original signals, the M/S stereo coding encodes the sun (middle) and the difference (side) of the signals of a channel-pair. If the left and right signals are highly correlated (as is usually true), coding the M/S signals will require less bits than coding the original signals. Intensity/Coupling tool exploits the fact that the perception of high-frequency sound component mainly relies not on each sample but on the analysis of energy-time envelopes. Thus for certain types of signals, only a single set of spectrum coefficients which is shared among several audio channels need to be transmitted with virtually no loss in audio quality. The original energy envelopes of the coded channels are preserved approximately such that each channel signals can be reconstructed with its original level after decoding.

2.8 Perceptual Noise Substation (PNS)

Conventional coding method encodes the audio without separating the tone-like and noise-like signals. However, it is inefficient to encode noise-like signal in such way. The PNS tool in MPEG-4 AAC gives a very compact representation of noise-like signals. In this way, the PNS tool further increases the compression efficiency for some types of input signals since only a flag and the energy information is coded and transmitted. In the encoder, the noise-like components of the input signals are detected. If the coefficients are detected as noise-like signals, then the coefficients will not be quantized and entropy coded as usual. Instead, only a noise substitution flag and the total power of the noise-like signals are coded and transmitted. In the decoder, the pseudo noise with desired total power is inserted for the substituted spectrum coefficients.

2.9 Rate-Distortion Control Process

Motivated by the human auditory system, the spectral coefficients are grouped into a number of bands, called *scale factor bands* (SFB). The spectral coefficients in one SFB are quantized by a non-uniform quantizer. The non-uniform quantizer in AAC is formulated in (2.1). The *common_scalefactor* is the common quantizer step size information for all the SFB. The quantizer step size which determines the quantization distortion (noise-to-masking ratio, NMR) is controlled by the parameter, Scale Factor (SF). Note that, the parameter, Scale Factor, here and in the following discussions is equal to (*scalefactor–common_scalefactor*) in (2.1).

$$x\_q = \mathrm{int}\left((abs(mdct\_line)^{3/4} \times 2^{\frac{3\times(saclefactor - common\_scalefactor)}{16}} + 0.4054 \right) \qquad (2.1)$$

The quantized coefficients in one band are then entropy-coded by one of the twelve pre-designed Huffman CodeBooks (HCBs). Each SFB can have its own quantization step size and HCB. In addition, the indices of SFs and HCBs have to be coded and transmitted as side information. In AAC, the SFs are differentially coded relative to the previous SF and then Huffman coded using a pre-designed codebook [2]. Taking Fig. 2.8 as example, instead of encoding the SF value of the *2*nd SFB, 65, the difference between the *2*nd SFB and the *1*st SFB, 5, is coded. The indices of HCBs are coded by run-length codes [22]. A run-length code in AAC is 9 bits long, which is composed of a 4-bits codebook index and a 5-bits run index. For example, as shown in Fig. 2.8, the *3*rd HCB is used from the *1*st SFB to the *10*th SFB; therefore, these 10 HCB indices (same value) are coded together by one run-length code, in which the codebook index is 3 and run index is 10. **It is obvious that the differential and run-length coding induce the inter-band dependency in coding process**. The R-D controller, our focus in this paper, is to determine two critical parameters, the values of SF and HCB, for each SFB so as to optimize the selected criterion under the given bit rate constraint. In the following discussions, if the context is clear, the abbreviation “SF” is also referred to the value of SF and “HCB” is also referred to the index of HCB.

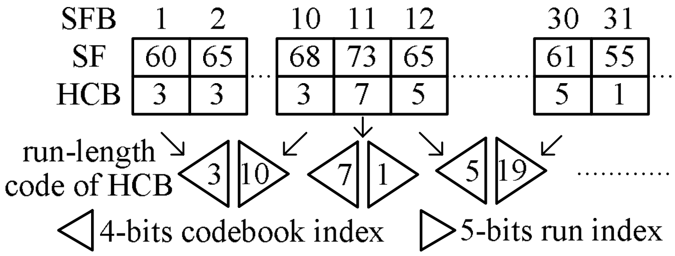


Fig. 2.8: An example of values of SF and HCB.

A typical rate-distortion (R-D) control process in the MPEG audio encoder has two nested iteration loops, the outer iteration loop and the inner iteration loop. Thus, it is often called the *two-loop search* (TLS). The outer iteration loop is the *distortion control loop* that handles the distortion associated with each band. The inner iteration loop, also called the *rate control loop*, adjusts coding bits to fit the target bit budget for a frame. The flowcharts of the outer loop and the inner loop specified in AAC are shown in Fig. 2.9.

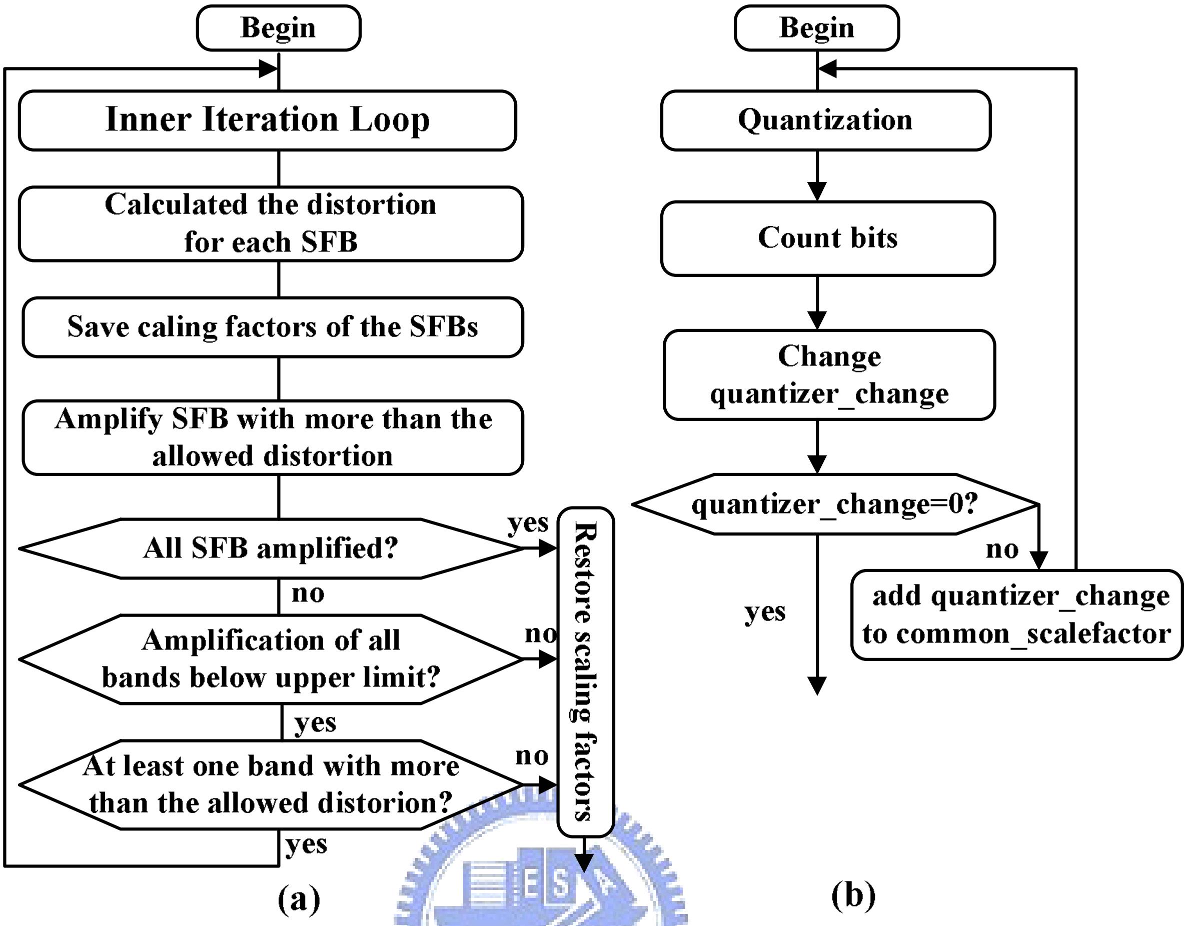

Begin
Inner Iteration Loop
Calculated the distortion for each SFB
Save caling factors of the SFBs
Amplify SFB with more than the allowed distortion
All SFB amplified?
yes
no
Amplification of all bands below upper limit?
no
yes
At least one band with more than the allowed distorion?
no
yes
Restore scaling factors
(a)
Begin
Quantization
Count bits
Change quantizer_change
quantizer_change=0?
no
add quantizer_change to common_scalefactor
yes
(b)

Fig. 2.9: AAC (a) outer iteration loop and (b) inner iteration loop. [2]

Chapter 3
Cascaded Trellis-Based Rate-Distortion Control Algorithm

In this chapter, we describe the proposed first type of R-D control algorithms, called *cascaded trellis-based* (CTB) scheme. The proposed CTB algorithm and its variations are described in Section 3.1. The proposed fast trellis search schemes are described in Section 3.2. The complexity analysis of the proposed R-D control algorithms and the simulation results with quality evaluation are summarized in Section 3.3.

3.1 Cascaded Trellis-Based Optimization Scheme

We start with the problem formulation of the R-D control algorithm for AAC in Section 3.1.1. The trellis-based (TB) procedures for SF optimization and HCB optimization in the CTB scheme are described in Sections 3.1.2 and 3.1.3, respectively. One key element in the trellis-based optimization process on SF, so-called “pseudo HCB”, is explained in Section 3.1.4. Finally, the procedure of the complete CTB optimization scheme is summarized in Section 3.1.5.

3.1.1 Problem Formulation

For the perceptual audio coders, noise-to-masking ratio (NMR) is the most widely used objective measure in the R-D control module for modeling the subjective perceptual distortion. Based on NMR, there are two commonly used criteria for R-D optimization, the *average noise-to-mask ratio* (ANMR) and the *maximum noise-to-mask ratio* (MNMR) [23]. In AAC, the differential coding of SFs and the run-length coding of HCBs introduce inter-band dependence in parameter selection. In order to take into account the inter-band dependence in encoding SFs and HCBs, we need to consider all their possible combinations for all SFBs and examine the bits and distortion produced by each combination. If such inter-band dependence does not exist, we can decide SF and HCB for each SFB separately and add all bands together to find the global optimal solution.

Mathematically, the R-D optimization problems for minimizing ANMR and MNMR under a given bit rate constraint are formulated by (3.1) and (3.2), respectively.

$$\text{minimize} \sum_i w_i d_i \ \text{subject to (s.t.)} \ \sum_i (b_i + D(s_i - s_{i-1}) + R(h_{i-1}, h_i)) \le PB \tag{3.1}$$

$$\text{minimize} \, (\max_i w_i d_i) \ \text{subject to (s.t.)} \sum_i (b_i + D(s_i - s_{i-1}) + R(h_{i-1}, h_i)) \le PB \tag{3.2}$$

where i is the SFB index, w_i is the inverse of the masking threshold, and d_i is the quantization distortion, the mean squared quantization errors. In (3.1), $\Sigma_i \, w_i d_i$ is the sum of NMR over all SFBs in a frame and in (3.2), $\max_i \, w_i d_i$ is the maximum NMR in a frame. The parameter values of SF and HCB for the ith SFB are denoted by s_i and h_i, respectively. Symbol $D()$ is a function of SF, representing the number of bits produced by differential coding of SF. Symbol $R()$ is a function of HCB, representing the number of bits produced by run-length coding of HCB. The returned function values in both cases are numbers of bits to encode the arguments. Parameter b_i is the number of bits for coding the quantized spectral coefficients (QSCs) and the parameter PB is the prescribed bit rate for an audio frame.

To solve (3.1) and (3.2), the straight-forward joint optimization of SF and HCB for all SFBs is exorbitantly complex. For one frame in AAC, the number of SF values is 60, the number of HCB indices is 12, and there are 49 SFBs in total. Therefore, to find the optimal solution of all combinations, the complexity of brute force search is $O((60 \cdot 12)^{49})$. In [7][8], a dynamic programming approach, called *joint trellis-based* (JTB) scheme in this paper, is proposed to find the optimal SF and HCB for all SFBs jointly at a reduced complexity. As shown in [7][8], the problem of minimizing ANMR in (3.1) can be reformulated as minimizing the unconstrained cost functions, C_{ANMR}, with the Lagrangian multiplier λ:

$$C_{\mathrm{ANMR}} = \sum_i w_i d_i + \lambda \cdot (b_i + D(s_i - s_{i-1}) + R(h_{i-1}, h_i)) \tag{3.3}$$

Likewise, the problem of minimizing MNMR in (3.2) can be reformulated as minimizing the cost functions, C_{MNMR}, under the constraint: $w_i d_i \le \lambda$, $\forall i$, for a certain value of λ.

$$C_{\mathrm{MNMR}} = \sum_i b_i + D(f_i - f_{i-1}) + R(h_{i-1}, h_i) \tag{3.4}$$

The research in [7][8] shows that the problem of minimizing C_{ANMR} and C_{MNMR} can be efficiently solved by the Viterbi search through the trellis, in which we compute only the legal transitions from the previous state to the current state [12][13]. Although, the search complexity of JTB scheme [8], $O((60 \cdot 12)^2 \cdot 49)$, is much lower than that of brute force search, it is still extremely high for practical applications.

As shown in Fig. 3.1, a simplification of the JTB scheme is to search for the SF and the HCB values in two consecutive steps without going through all possible combinations. Ideally, the order of complexity of our CTB scheme goes down to $O((60^2+12^2) \cdot 49)$. However, because these two steps are strongly correlated, we need to design the cascaded algorithm with special treatment on this issue to reduce performance degradation. This is the main point of this section.

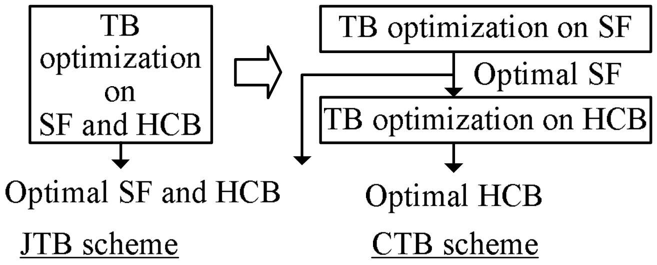


Fig. 3.1: Joint trellis-based scheme vs. cascaded trellis-based scheme.

3.1.2 Trellis-Based Optimization on SF

In this sub-section, the procedures of trellis-based optimization on SF aiming at two criterions, ANMR and MNMR, are described.

1) Trellis-Based Procedure for ANMR Minimization:

The problem of minimizing ANMR in the JTB scheme is formulated as minimizing the unconstrained cost functions, C_{ANMR}, in (3.3). However, to break the combined one step into two consecutive steps in our CTB scheme, this problem is reformulated as minimizing two unconstrained cost functions, $C_{\mathrm{SF\_ANMR}}$ and C_{HCB}, as follows.

$$C_{\mathrm{SF\_ANMR}} = \sum_i w_i d_i + \lambda \cdot (b_i + D(s_i - s_{i-1})) \tag{3.5}$$

$$C_{\mathrm{HCB}} = \sum_i b_i + R(h_{i-1}, h_i) \tag{3.6}$$

The minimization of $C_{\mathrm{SF\_ANMR}}$ is described in this sub-section, and the minimization of C_{HCB} will be described in Section 3.1.3. Because $C_{\mathrm{SF\_ANMR}}$ and C_{HCB} are minimized in two separate steps, the global optimality of C_{ANMR} is not guaranteed although the computation is significantly reduced. Our contribution described hereafter is to develop techniques that would come close to the global optimality.

Similar to the approach in the JTB scheme, the goal of finding proper SFs that minimize $C_{\mathrm{SF\_ANMR}}$ can be achieved by looking for the optimal path through the trellis. Each stage in the trellis corresponds to an SFB and there are *N_SFB* stages in total. However, different from JTB, each state at the *i*th stage in our scheme only represents an SF candidate for the *i*th SFB. In other words, at the *i*th stage, if a path passes through the *m*th state, it means that the *m*th SF candidate is used to encode the *i*th SFB.

For a given value of λ, the Viterbi search procedure for finding a proper set of SFs that minimize $C_{\mathrm{SF\_ANMR}}$ is outlined below. We denote $\Upsilon_{k,i}$ as the *k*th state at the *i*th stage and denote $C_{k,i}$ as the minimum accumulative-partial cost ending at $\Upsilon_{k,i}$. The state-transition cost, $T_{l,i\text{-}1\rightarrow k,i}$, from $\Upsilon_{l,i\text{-}1}$ to $\Upsilon_{k,i}$ is $\lambda{\cdot}D(s_{k,i} - s_{l,i\text{-}1})$, where $s_{k,i}$ is the SF value associated with the state $\Upsilon_{k,i}$.

1) Initialize all the states and start trellis search from the first stage. $C_{k,0}$ = 0, $\forall k$ and i =1.

2) For each state at the ith stage, find the best path from the previous stage by examining all the states at the (i-1)th stage leading to the current state. The best path ending at $\Upsilon_{k,i}$ is the one that has the minimum accumulative-partial $C_{k,i}$. That is, we look for the minimum value of $C_{k,i}$, $\forall k$;

$$C_{k,i} = \min{}_l \{C_{l,i-1} + (w_i d_{k,i} + \lambda \cdot b_{k,i} + T_{l,i-1\rightarrow k,i})\} \tag{3.7}$$

3) Check the index, i. If i < N_SFB, set i = i+1 and go to step 2.

2) Trellis-Based Procedure for MNMR Minimization:

The problem of minimizing MNMR in the JTB scheme is formulated as minimizing the cost functions, C_{MNMR}, in (3.4). In our CTB scheme, this problem is reformulated as the minimization of two cost functions, $C_{\mathrm{SF\_MNMR}}$ in (3.8) and C_{HCB} in (3.6), under the constraint: $w_i d_i \le \lambda$, $\forall i$, for a certain value of λ.

$$C_{\mathrm{SF\_MNMR}} = \sum_i b_i + D(s_i - s_{i-1}) \tag{3.8}$$

Similar to the trellis-based ANMR optimization on selecting SF described above, an "SF trellis" is constructed for minimizing $C_{\mathrm{SF\_MNMR}}$. For a given value of λ, the Viterbi search procedure for finding proper SFs that minimize $C_{\mathrm{SF\_MNMR}}$ is outlined below. The state-transition cost, $T_{l,i\text{-}1\rightarrow k,i}$, is $D(s_{k,i} - s_{l,i\text{-}1})$.

1) Initialize. $C_{k,0}$ = 0, $\forall k$ and i =1.

2) For the ith stage, only the particular state, which the NMR ($w_i d_{k,i}$) associated with is less than or equal to λ, is valid for trellis search. Therefore, before staring the trellis search, we must find the valid states for the ith stage, $\Upsilon_{k,i}$, $\forall k$.

3) For each valid state at the ith stage, find the best path from the previous stage by

examining all the valid states in the (i-1)th stage leading to the current state. That is, we compute and find the $C_{k,i}$ such that;

$$C_{k,i} = \min{}_l \{ C_{l,i-1} + (b_{k,i} + T_{l,i-1\rightarrow k,i}) \} \tag{3.9}$$

4) If $i < N\_SFB$, set $i = i$+1 and go to step 2.

After completing the forward “search and expansion” step through the trellis, the optimal path in the trellis can be extracted by tracing backward from the state with minimum C_{k,N_SFB} at the last stage. Consequently, the optimal SFs for all SFBs that minimize $C_{\mathrm{SF\_MNMR}}$ (or $C_{\mathrm{SF\_ANMR}}$) are determined.

As described in [7][8], to a band below the masking threshold, any values of SF can be assigned. Therefore, its associated state in the trellis is split into two consecutive states. At the first state, the spectral coefficients are quantized using the assigned valid SF, and at the second state, all quantized values of spectral coefficients are set to zero.

3.1.3 Trellis-Based Optimization on HCB

The HCB optimization is performed under the assumption that the SF (value) for each SFB has already been decided. In our CTB scheme, SF is determined by the trellis-based optimization on SF described in Section 3.1.2. With a determined SF, QSCs for each SFB are fixed and thus the b_i term in the cost function C_{HCB} (see (3.6)) depends only on the selection of HCB. Therefore, C_{HCB} can be restated as (3.10).

$$C_{\mathrm{HCB}} = \sum_i H_{h_i}(\boldsymbol{q}_i) + R(h_{i-1}, h_i) \tag{3.10}$$

where $\boldsymbol{q}_i$ (vector) contains the QSCs for the ith SFB and symbol H_h () is a function of QSCs, representing the number of bits produced by Huffman-coding of QSCs using the hth HCB. The goal of the optimization procedure here is to find the HCBs for all SFBs that minimize the cost function C_{HCB} and this can be achieved again by finding the optimal path through the trellis with states now being HCB.

An “HCB trellis” is thus constructed for searching for the minimum C_{HCB}. Each state at the ith stage represents an HCB candidate for the ith SFB. The state-transition cost, $T_{n,i\text{-}1\rightarrow m,i}$, from $\mathrm{Y}_{n,i\text{-}1}$ to $\mathrm{Y}_{m,i}$ is $R(h_{n,i\text{-}1}\ ,\ h_{m,i})$, where $h_{m,i}$ is the HCB associated with the state $\mathrm{Y}_{m,i}$. According to the run-length coding rule in AAC, $R(h_{n,i\text{-}1}\ ,\ h_{m,i})$ is defined by (3.11). In other words, no extra bits are transmitted if the same HCBs are used in two neighboring SFBs.

$$R(h_{n,i-1}, h_{m,i}) = \begin{cases} 0, & if \;\; n = m \\ 9, & otherwise \end{cases} \tag{3.11}$$

The Viterbi search procedure for finding proper HCBs that minimize C_{HCB} is outlined below.

1) Initialize. $C_{m,0} = 0$, $\forall m$ and i =1.

2) For each state at the ith stage, find the best path from the previous stage by examining all the states at the (i-1)th stage leading to the current state. That is, we find the best $\mathrm{Y}_{m,i}$ by computing and find the $C_{m,i}$ such that

$$C_{m,i} = \min{}_n \{C_{n,i-1} + (H_{h_{m,i}}(\boldsymbol{q}_i) + T_{n,i-1\rightarrow m,i})\} \tag{3.12}$$

where $\boldsymbol{q}_i$ is the vector of the QSCs in the stage i.

3) If $i < N\_SFB$, set $i = i$+1 and go to step 2.

Similar to the trellis-based optimization on SF, after completing the forward search/expansion step through the trellis, the optimal path in the trellis can be extracted by tracing backward from the minimum C_{m,N_SFB} state at the last stage. Then, the optimal HCBs for all SFBs that minimize C_{HCB} are determined.

3.1.4 Pseudo HCB for SF Optimization

1) Motivation for Pseudo HCB:

We first look at the MNMR minimization case. The key problem in splitting (3.4) into (3.8) and (3.6) is to choose the correct (optimal) value of $b_{k,i}$ in (3.8). In (3.8), the $w_i d_{k,i}$ or $D(s_{k,i} - s_{l,i\text{-}1})$ term is unique for a given state or state transition in the SF trellis. However, the value of $b_{k,i}$ depends not only on $s_{k,i}$ associated with the state in the SF trellis; it also depends on the choice of HCB. In the JTB scheme, SF and HCB are chosen simultaneously. Therefore, for each candidate value of SF, all possible $b_{k,i}$ values, corresponding to 12 candidate HCBs, are evaluated. In other words, the chosen value of $b_{k,i}$ for each state $\mathrm{Y}_{k,i}$ in the trellis for JTB optimization scheme is optimal [7][8]. But in our sequential optimization scheme, the value of $b_{k,i}$ for the state $\mathrm{Y}_{k,i}$ in (3.8) is estimated based upon the available information. The estimated value of $b_{k,i}$ may not be the optimal value and this may further induce an incorrect (non-optimal) selection in SF optimization. For example, two candidate paths in the SF trellis, A and B, are shown in (3.13). Path A is better than path B because $C^{\mathrm{A}}_{\mathrm{SF\_MNMR}} < C^{\mathrm{B}}_{\mathrm{SF\_MNMR}}$, where $C^{\mathrm{A}}_{\mathrm{SF\_MNMR}}$ and $C^{\mathrm{B}}_{\mathrm{SF\_MNMR}}$ are the $C_{\mathrm{SF\_MNMR}}$ values of path A and path B, respectively. Note that $\hat{b}^{\mathrm{A}}_i$ and $\hat{b}^{\mathrm{B}}_i$ in (3.13) are the estimated values of b_i for path A and path B. If the decision on SF is made at this point, path A is chosen. Now, let us go one step further. Based on the selected SF sets of path A and path B, we can find their optimal HCBs, h_i^{A} and h_i^{B} respectively, according to the HCB optimization procedure described in Section 3.1.3. Then, their actual bits information b_i^{A} and b_i^{B}, for path A and path B, respectively, is obtained. Finally, the total costs $C^{\mathrm{A}}_{\mathrm{MNMR}}$ and $C^{\mathrm{B}}_{\mathrm{MNMR}}$ for two candidate paths are shown in (3.14). The result in (3.14) indicates that path B is actually better than path A when the bits information is correct. With a wrong estimate on b_i, our CTB algorithm would pick up path A for SFs and thus it fails to find the overall optimal path B.

$$C^{\mathrm{A}}_{\mathrm{SF\_MNMR}} = \sum_i \hat{b}^{\mathrm{A}}_i + D(s^{\mathrm{A}}_i - s^{\mathrm{A}}_{i-1}) < C^{\mathrm{B}}_{\mathrm{SF\_MNMR}} = \sum_i \hat{b}^{\mathrm{B}}_i + D(s^{\mathrm{B}}_i - s^{\mathrm{B}}_{i-1}) \tag{3.13}$$

$$\begin{aligned} C^{\mathrm{A}}_{\mathrm{MNMR}} &= \sum_i (b^{\mathrm{A}}_i + D(s^{\mathrm{A}}_i - s^{\mathrm{A}}_{i-1}) + R(h^{\mathrm{A}}_{i-1}, h^{\mathrm{A}}_i)) > \\ C^{\mathrm{B}}_{\mathrm{MNMR}} &= \sum_i (b^{\mathrm{B}}_i + D(s^{\mathrm{B}}_i - s^{\mathrm{B}}_{i-1}) + R(h^{\mathrm{B}}_{i-1}, h^{\mathrm{B}}_i)) \end{aligned} \tag{3.14}$$

Clearly, with a more accurate estimate on $b_{k,i}$, we can select better SFs. For this aim, the concept of “pseudo HCB” is proposed for the trellis-based optimization on SF. The preceding discussions on choosing HCB can be applied to the ANMR minimization case.

2) Design of Pseudo HCB:

When the trellis-based optimization on SF is performed in the pseudo HCB mode, a pseudo HCB with an index set $h^{v}_{k,i}$ needs to be constructed for the state $\mathrm{Y}_{k,i}$ to produce $b_{k,i}$ in (3.5) and (3.8). It can be constructed in several ways. For example, $h^{v}_{k,i}$ may contain only one of the 12 candidate HCBs or several codebooks. In order to improve the accuracy of the estimated values of $b_{k,i}$ and $h^{v}_{k,i}$, we analyze the data collected from the JTB optimization scheme.

For a given value of λ, using the JTB scheme, we can find a set of optimal parameters, $\boldsymbol{s}^{\mathrm{JTB}}_{opt}$, $\boldsymbol{h}^{\mathrm{JTB}}_{opt}$ and $\boldsymbol{b}^{\mathrm{JTB}}_{opt}$ that minimizes the cost function, C_{ANMR} in (3.3) or C_{MNMR} in (3.4). For comparison purposes, we also construct a reference set of QSC bits, $\boldsymbol{b}^{\mathrm{JTB}}_{\min}$, in the following way. For the ith SFB, $b^{\mathrm{JTB}}_{\min,i}$ is the minimum number of bits for encoding $\boldsymbol{q}^{\mathrm{JTB}}_{opt,i}$ and is determined by $b^{\mathrm{JTB}}_{\min,i} = \min_m \{H_m(\boldsymbol{q}^{\mathrm{JTB}}_{opt,i})\}$, where $\boldsymbol{q}^{\mathrm{JTB}}_{opt,i}$ is the QSCs quantized by using $s^{\mathrm{JTB}}_{opt,i}$. In other words, without considering the bits for coding the HCB indices, $b^{\mathrm{JTB}}_{\min,i}$ is the lowest bits number produced by any of the 12 HCBs applied to the QSCs. Because the coded bits for HCB indices, $R(h_{i-1}, h_i)$, are also included in the overall optimization procedure, when comparing coding bits for QSCs only, $\boldsymbol{b}^{\mathrm{JTB}}_{opt}$ is higher than or equal to $\boldsymbol{b}^{\mathrm{JTB}}_{\min}$.

By collecting the statistics from the simulations on ten audio sequences, the histogram of the differences between $\boldsymbol{b}_{opt}^{\mathrm{JTB}}$ and $\boldsymbol{b}_{\min}^{\mathrm{JTB}}$, denoted by Δ***b***, is shown in Fig. 3.2. We observe that over 91% of Δ***b*** is less than 3 for both ANMR and MNMR criterions. In general, we can choose the HCB that produces the minimum QSC bits, $\boldsymbol{b}_{\min}^{\mathrm{JTB}}$.

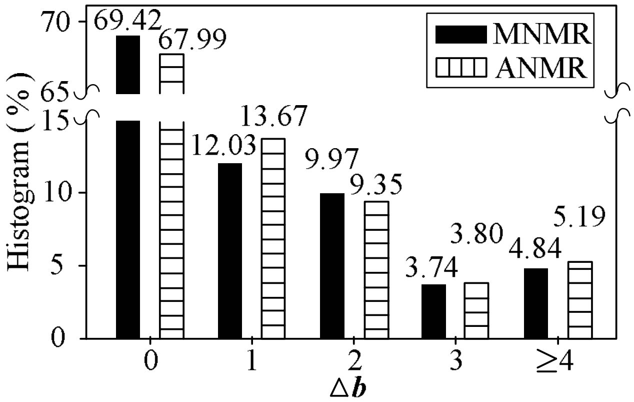


Fig. 3.2: Histogram on Δ***b***.

After examining this characteristics of $\boldsymbol{b}_{opt}^{\mathrm{JTB}}$, we derive a rule in determining $h_{k,i}^{v}$ and $b_{k,i}$. For the state $\Upsilon_{k,i}$, $h_{k,i}^{v}$ is the index set of HCB that satisfies the proposed rule in (3.15); namely,

$$h_{k,i}^{v}=\left\{n \mid H_{n}(\boldsymbol{q}_{k,i}) \leq \min_{m}\{H_{m}(\boldsymbol{q}_{k,i})\}+\delta, n \in \{0,\cdots,11\}\right\} \tag{3.15}$$

The $\min_m\{H_m(\boldsymbol{q}_{k,i})\}$ term is the minimum number of bits for coding $\boldsymbol{q}_{k,i}$ without considering the coding bits for HCB indices and δ is an offset parameter. For example, if $H_1(\boldsymbol{q}_{k,i})$ and $H_3(\boldsymbol{q}_{k,i})$ are both smaller than or equal to $\min_m\{H_m(\boldsymbol{q}_{k,i})\}+\delta$, then $h_{k,i}^{v}$ equals to {1,3}. Although (3.5) and (3.8) do not include the bits number for coding HCB indices, it is found from experiments that including this term leads to a better estimate of SF. Therefore, we expand (3.5) to approximate (3.3) and expand (3.8) to approximate (3.4) with additional terms. Based on the above observation, $b_{k,i}$ is rewritten as:

$$b_{k,i} = \left(\left(\sum\nolimits_{n\in h_{k,i}^{v}} H_n(q_{k,i})\right)\Big/\left|h_{k,i}^{v}\right|\right) + \alpha \cdot R_v(h_{l,i-1}^{v}, h_{k,i}^{v}) \tag{3.16}$$

where α is a weight for including $R_v(h_{l,i-1}^{v}, h_{k,i}^{v})$ into $b_{k,i}$ and $|h_{k,i}^{v}|$ is the number of elements in $h_{k,i}^{v}$. The symbol R_v is the run-length coding function performed on the pseudo HCB and is defined below.

$$R_v(h_{l,i-1}^{v}, h_{k,i}^{v}) = \begin{cases} 0, & if\ (h_{l,i-1}^{v} \cap h_{k,i}^{v}) \neq \phi \\ 9, & otherwise \end{cases} \tag{3.17}$$

Note that the R_v() function is essentially the R() function in (3.11). However, because $h_{k,i}^{v}$ and $h_{l,i-1}^{v}$ are index sets of HCB, the intersection is used in (3.17).

After having derived (3.15) and (3.16), we still need to determine the proper values for δ and α. The values of δ and α can be determined by examining the difference between the JTB scheme and the CTB scheme at different values of δ and α and the results are shown in Fig. 3.3 and Fig. 3.4. Note that $C_{\mathrm{ANMR}}^{\mathrm{JTB}}$ and $C_{\mathrm{ANMR}}^{\mathrm{CTB}}$ are the C_{ANMR} (in (3.3)) derived using the JTB scheme and CTB scheme, respectively. $C_{\mathrm{MNMR}}^{\mathrm{JTB}}$ and $C_{\mathrm{MNMR}}^{\mathrm{CTB}}$ are the C_{MNMR} (in (3.4)) derived using the JTB scheme and CTB scheme, respectively. We find that for a wide range of δ values, we can achieve a pretty good performance when $R_v(h_{l,i-1}^{v}, h_{k,i}^{v})$ is included in $b_{k,i}$ ($\alpha > 0$). As Fig. 3.3 and Fig. 3.4 indicate, the case that δ=1 and α=0.5 gives the best results. Hence, we choose 1 for δ and 0.5 for α in our implementation.

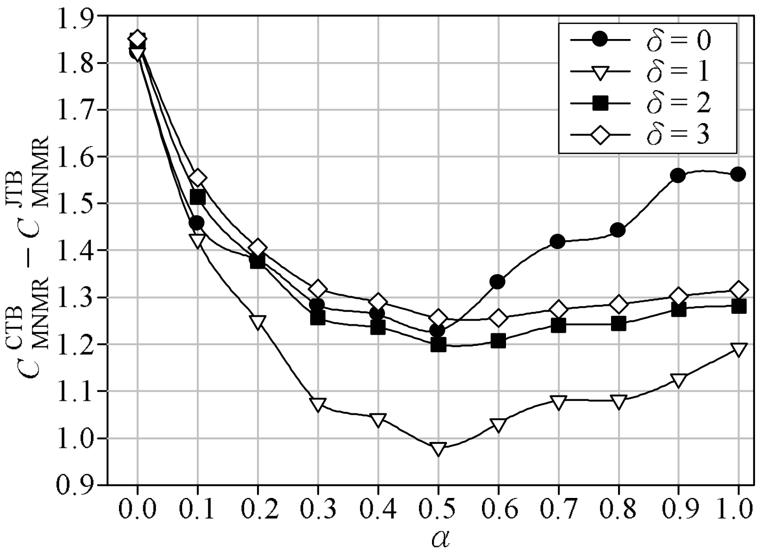


Fig. 3.3: ($C^{\text{CTB}}_{\text{MNMR}}$ - $C^{\text{JTB}}_{\text{MNMR}}$) vs. ($\delta$, α).

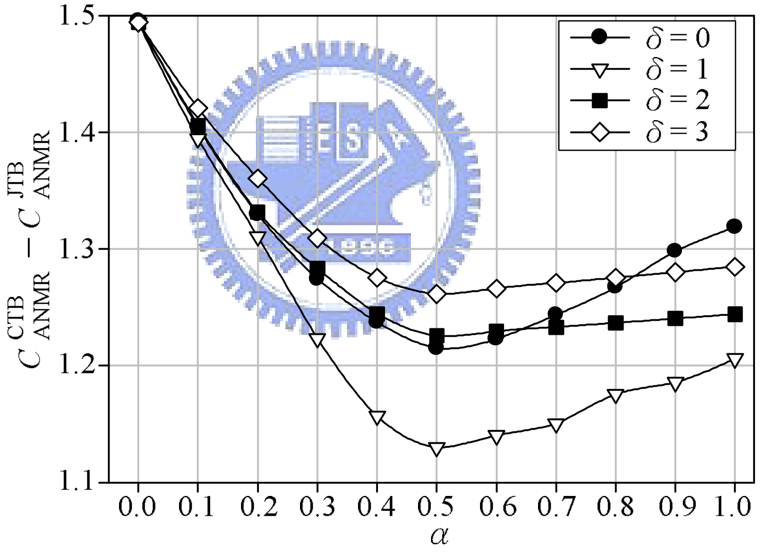


Fig. 3.4: ($C^{\text{CTB}}_{\text{ANMR}}$ - $C^{\text{JTB}}_{\text{ANMR}}$) vs. ($\delta$, α).

3.1.5 Cascaded Trellis-Based Optimization Procedure

The major steps in our CTB scheme have been described in detail in Sections 3.1.2 to 3.1.4. The flowchart of the complete CTB optimization scheme is summarized in Fig. 3.5. Passing through the first step (block), we obtain a set of optimal SF, $\boldsymbol{s}_{opt}$. Then, the second step produces $\boldsymbol{h}_{opt}$, a set of optimal HCB. Based on this set of $\boldsymbol{h}_{opt}$, $\boldsymbol{s}'_{opt}$ is a new set of optimal SF derived at the end of the third step. Note that the same trellis-based optimization on SF is used in steps 1 and 3, but they are derived using different HCB modes. In step 1, the pseudo-HCB mode is used and in step 3 the fixed-HCB mode is used. The pseudo-HCB mode has been described in Section 3.1.3. For the fixed-HCB mode, an index set of fixed HCBs, $\boldsymbol{h}^X = [h_1^X \quad h_2^X \ .... \ h^X_{N\_SFB}]$, is pre-chosen and used in the trellis-based SF optimization procedure. For all the states at the *i*th stage in the SF trellis in Section 3.1.2, the QSCs, $\boldsymbol{q}_{k,i}$, are entropy-coded using the h_i^Xth HCB; therefore, the value of $b_{k,i}$ in (3.5) and (3.8) is correctly calculated by $b_{k,i} = H_{h_i^X}(\boldsymbol{q}_{k,i}), \forall k$. In this flowchart, $\boldsymbol{h}^X$ is derived from step 2 and is the final $\boldsymbol{h}_{opt}$ in our CTB scheme.

The 4-step procedure in Fig. 3.5 is called *two-iteration mode*, because the optimization process on SF is done twice. The second optimization step on SF (step 3) can recover some inadequate SF values determined in step 1 owing to the incorrect HCB used in the pseudo HCB model. The CTB optimization can be further simplified, in which step 3 is omitted to save computation. This is called *one-iteration mode*. Clearly, there is a trade-off between complexity and coding performance.

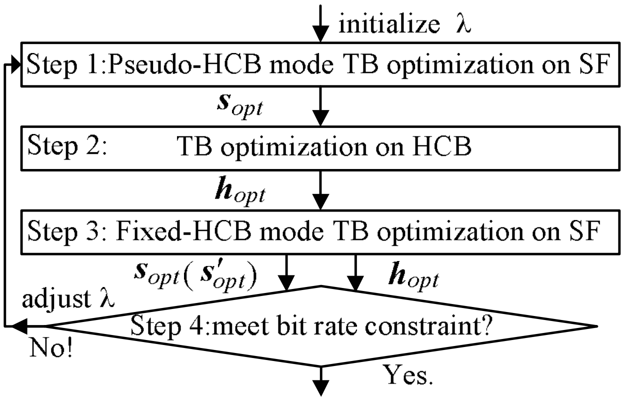

initialize λ
Step 1:Pseudo-HCB mode TB optimization on SF
$\boldsymbol{s}_{opt}$
Step 2: TB optimization on HCB
$\boldsymbol{h}_{opt}$
Step 3: Fixed-HCB mode TB optimization on SF
$\boldsymbol{s}_{opt}(\boldsymbol{s}'_{opt})$
$\boldsymbol{h}_{opt}$
adjust λ
Step 4:meet bit rate constraint?
No!
Yes.

Fig. 3.5: Flowchart of the cascaded trellis-based optimization scheme.

3.2 Fast Trellis Search Algorithm

As described in the previous section, the basic structure of our CTB algorithm (or JTB algorithm) is trellis search. If we can reduce the complexity of trellis search, we speed up the entire process. In this section, we propose fast algorithms aiming at reducing the trellis search complexity. The complexity of the trellis-based optimization scheme depends on the searching range (number of states) of each stage in the trellis. Hence, reducing the candidate states at each stage is an effective way in reducing the overall computational complexity.

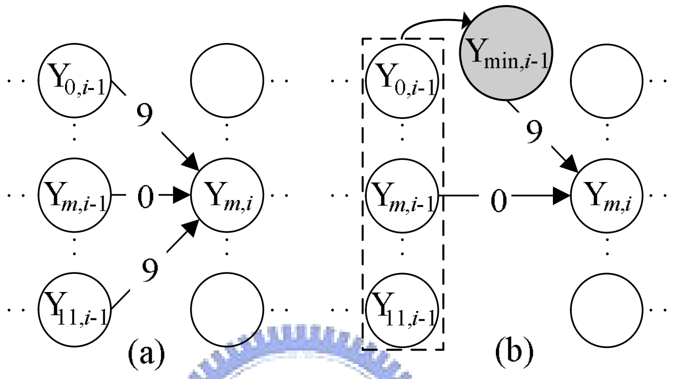
3.2.1 Fast Search Algorithm for HCB Optimization

In AAC, SFs are differentially coded and HCBs are coded by run-length coding. Run-length coding can be viewed as a special case of differential coding; therefore, the procedure of trellis-based optimization on HCB is similar to that on SF. However, the output of run-length coding has only two possible values, either 0 or 9 as shown in (3.11). As shown in Fig. 3.6(a), in order to find the optimal path ending at $\mathrm{Y}_{m,i}$, all the HCB candidates at the (i-1)th stage have to be taken into consideration. In AAC, there are 12 pre-designed HCBs; thus, the searching complexity for finding all the optimal paths ending at the ith stage is 12×12.

The number on the directional branch in Fig. 3.6 is the state-transition cost. Except for the path $\mathrm{Y}_{m,i\text{-}1}\rightarrow\mathrm{Y}_{m,i}$, the state-transition costs of the other 11 paths ending at $\mathrm{Y}_{m,i}$ are all identical (equal to 9). Therefore, in calculating $C_{m,i}$ in (3.12), among these 11 paths, the path with the least $C_{n,i\text{-}1}$ will result in the smallest $C_{m,i}$. Based on this property, a fast search algorithm is proposed, which is divided into two steps.

1) Among the 12 candidate states at the (i-1)th stage, the state with the minimum cost, $C_{\min,i\text{-}1}$, is chosen and treated as the pseudo thirteenth state, $\mathrm{Y}_{\min,i\text{-}1}$, and $C_{\min,i\text{-}1} = \min_n\{C_{n,i\text{-}1}\}$.
2) As shown in Fig. 3.6(b), when finding the optimal path ending at $\mathrm{Y}_{m,i}$, we only have to consider two paths, path ($\mathrm{Y}_{m,i\text{-}1}\rightarrow\mathrm{Y}_{m,i}$) and path ($\mathrm{Y}_{\min,i\text{-}1}\rightarrow\mathrm{Y}_{m,i}$). The rest of this revised searching procedure is the same as that in Section 3.1.3.

The searching complexity (in terms of branch metric calculation) of this fast algorithm is approximately $12+2\times12$, which is about 1/4 of the complexity of the full search algorithm. The first "12" term is needed for determining $\mathrm{Y}_{\mathrm{min},i\text{-}1}$. Note that the performance (accuracy) of this fast search algorithm is the same as that of the full search algorithm. Therefore, this fast search algorithm can be used by both CTB and JTB optimization schemes without any performance loss.



$\mathrm{Y}_{0,i\text{-}1}$
9
$\mathrm{Y}_{m,i\text{-}1}$
0
$\mathrm{Y}_{m,i}$
9
$\mathrm{Y}_{11,i\text{-}1}$
(a)
$\mathrm{Y}_{\mathrm{min},i\text{-}1}$
$\mathrm{Y}_{0,i\text{-}1}$
9
$\mathrm{Y}_{m,i\text{-}1}$
0
$\mathrm{Y}_{m,i}$
$\mathrm{Y}_{11,i\text{-}1}$
(b)

Fig. 3.6: Trellis representation of the HCB transition.
(a) Full search mode and (b) Fast search mode

3.2.2 Fast Search Algorithm for SF Optimization (MNMR)

In the trellis-based optimization on SF, each state in the trellis represents an SF candidate. Searching over a larger set of SF candidates can result in better performance, but at the cost of higher searching complexity. In general, the number of states (*sn*) for all the stages in the trellis are the same and the searching complexity for each stage transition in this uniform trellis is $sn \times sn$.

In this section, we propose two non-uniform (adaptive) trellis search algorithms for SF optimization under MNMR criterion, in which the number of state for each stage in the trellis can vary to reduce the overall searching complexity. The first one is called "**G**lobal **m**inimum (reference) SF-restricted **N**on-**u**niform *trellis*", or "GMNU" in short, and the second one is called "**L**ocal **m**inimum (reference) SF-restricted **N**on-**u**niform *trellis*", or "LMNU". In both cases, a reference SF is first identified and then the number of candidates is reduced against this reference. Note that all SFs at the *i*th stage in the trellis are sorted and indexed in ascending order.

First of all, we define the reference SF, s_i^{ref}, for the *i*th SFB as the largest SF among all the valid states at the *i*th stage. Note that s_i^{ref} is the upper bound of SF candidate at the *i*th stage, which means that the SF values of all the other valid states at the *i*th stage are less than s_i^{ref}. In the GMNU algorithm, we define the integer index of the “global minimum” (reference) SF, $s_{G\cdot Min}^{ref}$, as the minimum reference SF value of *all* the scale factor bands; that is, $s_{G\cdot Min}^{ref} = \min_j \{s_j^{ref}\}$. Then, we restrict the SF candidates at the *i*th stage in the range of $[s_{G\cdot Min}^{ref} - \varepsilon, s_i^{ref}]$. Namely, we only use the SF values between s_i^{ref} and $s_{G\cdot Min}^{ref} - \varepsilon$. Thus, the number of state at the *i*th stage, $sn_{Gm,i}$, equals to $(s_i^{ref} - s_{G\cdot Min}^{ref} + 1 + \varepsilon)$.

In the LMNU algorithm, we define the integer index of the ρth-order “local minimum” (reference) SF at the *i*th stage, $s_{L\cdot Min,i}^{ref}$, as $s_{L\cdot Min,i}^{ref} = \min_{i-\rho \le j \le i+\rho} \{s_j^{ref}\}$. Essentially, instead of all the bands, we only look at a *local neighborhood* of the current stage. Then, we restrict the SF candidates at the *i*th stage in the range of $[s_{L\cdot Min,i}^{ref} - \varepsilon, s_i^{ref}]$. Therefore, the number of states for the *i*th stage, $sn_{Lm,i}$, equals to $(s_i^{ref} - s_{L\cdot Min,i}^{ref} + 1 + \varepsilon)$. In both cases, ε is the parameter that controls the search range. In the simulations in Section 3.3, the value of ε is set to 1 and the *1*th-order (ρ=1) local minimum reference SF is used in the LMNU algorithm.

We first explain our motivation behind the fast GMNU algorithm. As shown in (3.8) in Section 3.1.2, The cost function $C_{\mathrm{SF\_MNMR}}$ can be divided into two parts, the differentially coded bits of SF values, $\Sigma_i D(s_i - s_{i-1})$, and the QSC coded bits, $\Sigma_i b$. In general, a larger value of SF will result in a smaller value of QSC and thus fewer coded QSC bits. If we set the SF value of the *i*th SFB to $s_i^{ref}, \forall i$, we achieve the globally minimal $\Sigma_i b$ but this rule leads to a larger $\Sigma_i D(s_i - s_{i-1})$. On the other hand, if we set the SF values of all SFBs to $s_{G \cdot Min}^{ref}$, we achieve the globally minimal $\Sigma_i D(s_i - s_{i-1})$ because the differential SF values are all zero, but this rule leads to a larger $\Sigma_i b$. Therefore, a good guess is that the optimal SF value that minimizes $C_{\text{SF\_MNMR}}$ likely falls in the range of $[s_{G \cdot Min}^{ref}, s_i^{ref}]$. Although exceptions could exist, our guess by far dominates. The statistics show that the probability of occurrence of exceptions is less than 0.5% and the average increased value on $C_{\text{SF\_MNMR}}$ due to exceptions is less than 1 bit.

The idea behind the LMNU algorithm is similar. However, we only look at the local $\Sigma_i b$ and $\Sigma_i D(s_i - s_{i-1})$ values in this case. Therefore, the LMNU algorithm requires an even lower computation but it leads to a higher distortion. Note that, GMNU and LMNU algorithms also can be used by both CTB and JTB optimization schemes.

As for the trellis-based ANMR optimization on SF, the cost function $C_{\text{SF\_ANMR}}$ depends not only on $\Sigma_i b$ and $\Sigma_i D(s_i - s_{i-1})$ but also on $\Sigma_i w_i d_i$. Therefore, the GMNU or LMNU fast search algorithm cannot be applied.

3.3 Simulation Results

In this section, we will evaluate the computational complexity and the coded audio quality of our proposed algorithms. Four types of R-D control algorithms are simulated and compared as described below.

1) The TLS algorithm in MPEG-4 AAC VM (VM-TLS).

2) The JTB optimization schemes aiming at minimizing ANMR and MNMR, abbreviated as JTB-ANMR and JTB-MNMR respectively, proposed in [7][8].

3) The CTB optimization schemes aiming at minimizing ANMR and MNMR, abbreviated as CTB-ANMR and CTB-MNMR respectively, described in Section 3.1.

4) The CTB-MNMR (or JTB-MNMR) incorporating GMNU, LMNU and the fast search algorithm for HCB optimization (FSHCB) described in Section 3.2.

To focus only on the R-D performance, all the optional tools in AAC, such as TNS and M/S stereo coding, are not used in our simulations. Ten two-channel audio sequences with a sampling rate at 44.1 kHz are tested. Two of them are extracted from MPEG SQAM [6], and the rest are from EBU [24].

3.3.1 Complexity Analysis

The complexity analysis for the aforementioned several R-D control algorithms is summarized in Table 3.1. The “Computation” column is the number of branch metrics in calculating one-stage transition in the trellis. For the convenience of comparison, the JTB-ANMR or JTB-MNMR is chosen to be the reference (ratio=1) and all the other schemes are rated based on this reference. Also shown in Table 3.1 is the storage requirement. Again, it is measured in the number of branch metrics.

We can find from Table 3.1 that the CTB-ANMR and CTB-MNMR schemes are approximately (142/*n*) times faster than the JTB-ANMR and JTB-MNMR schemes, in which *n* equals to 1 or 2. Moreover, the storage requirement for the trellis search in the CTB-ANMR and CTB-MNMR schemes is much smaller than that in the JTB-ANMR and JTB-MNMR scheme.

For the JTB scheme, the fast HCB search algorithm can be adopted to reduce the complexity down to 1/4. Note that sn_{Gm}^{ave} and sn_{Lm}^{ave} in Table 3.1 are the average number of states in the GMNU and LMNU algorithms and are calculated by using (3.18) and (3.19), respectively.

$$sn_{Gm}^{ave} = \left(\sum_{i=1}^{N\_SFB} (sn_{Gm,i-1} \cdot sn_{Gm,i}) \Big/ N\_SFB \right)^{1/2} \tag{3.18}$$

$$sn_{Lm}^{ave} = \left(\sum_{i=1}^{N\_SFB} (sn_{Lm,i-1} \cdot sn_{Lm,i}) \Big/ N\_SFB \right)^{1/2} \tag{3.19}$$

The simulation data show that a typical sn_{Gm}^{ave} is approximately 12 and sn_{Lm}^{ave} is around 5. Hence, the GMNU algorithm can reduce the complexity to $(12/60)^2$ =1/25 and the LMNU algorithm can reduce the complexity to $(5/60)^2$ =1/144.

Table 3.1: Complexity Analysis for JTB, CTB and Fast Search Algorithms.

Scheme	Computation	Ratio	Storage
JTB-ANMR (JTB-MNMR)	$(60\times2)^2\times12^2$	1	120×12
CTB-ANMR (CTB-MNMR)	$n\times(60\times2)^2+12^2$	*n*/142	120
JTB-MNMR + GMNU + FSHCB	$(sn_{Gm}^{ave}\times2)^2\times36$	~1/(25×4)	120×12
JTB-MNMR + LMNU + FSHCB	$(sn_{Lm}^{ave}\times2)^2\times36$	~1/(144×4)	120×12
CTB-MNMR + GMNU + FSHCB	$n\times(sn_{Gm}^{ave}\times2)^2+36$	*n*/3600	120
CTB-MNMR + LMNU + FSHCB	$n\times(sn_{Lm}^{ave}\times2)^2+36$	(*n*+0.4)/20736	120

※ *n* equals to 1 or 2. *n* = 1: one-iteration mode. *n* = 2: two-iteration mode.

3.3.2 Objective Quality

The rate-distortion curves of these bit allocation schemes are displayed in Fig. 3.7 and Fig. 3.8. Two major distortion metrics, ANMR and MNMR, are in use. We can find that the performance of the CTB scheme is similar to that of the JTB scheme. The ANMR performance loss is less than 0.2dB for the one-iteration CTB-ANMR and less than 0.1dB for the two-iteration CTB-ANMR (the lowest three curves in Fig. 3.7). The MNMR performance loss is less than 0.1dB for both one- and two-iteration CTB-MNMR (the lowest three curves in Fig. 3.8). All of them are much better than the VM-TLS (the top line).

The differences of performance between the fast SF search algorithms and the full search (original) algorithm for the CTB-MNMR scheme are shown in Fig. 3.9 and Fig. 3.10. Note that the original CTB-MNMR scheme uses the uniform trellis with the state number *sn*=60 in SF optimization. In addition to the two non-uniform trellis fast algorithms, GMNU and LMNU, for comparison purpose, we create two *uniform* trellis with smaller numbers of states, namely, sn=5 and sn=12, which approximately equal to the values of sn_{Gm}^{ave} and sn_{Lm}^{ave} in Section 3.3.1. There is nearly no performance loss for the GMNU algorithm (ANMR or MNMR Difference ≈ 0). The penalty on LMNU is small but exists. The advantage of the non-uniform algorithms over the uniform algorithms at about the same complexity is clearly shown in Fig. 3.9 and Fig. 3.10.

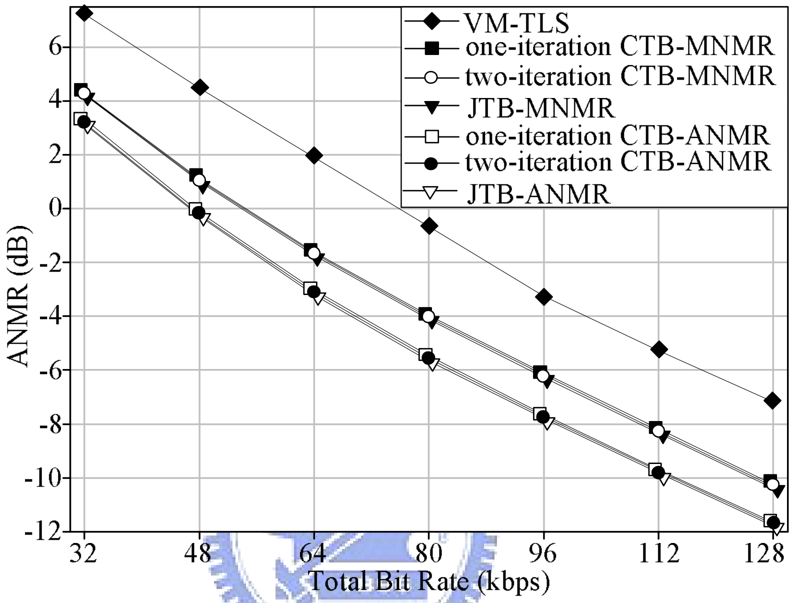


Fig. 3.7: ANMR rate-distortion comparison for VM-TLS, JTB and CTB.

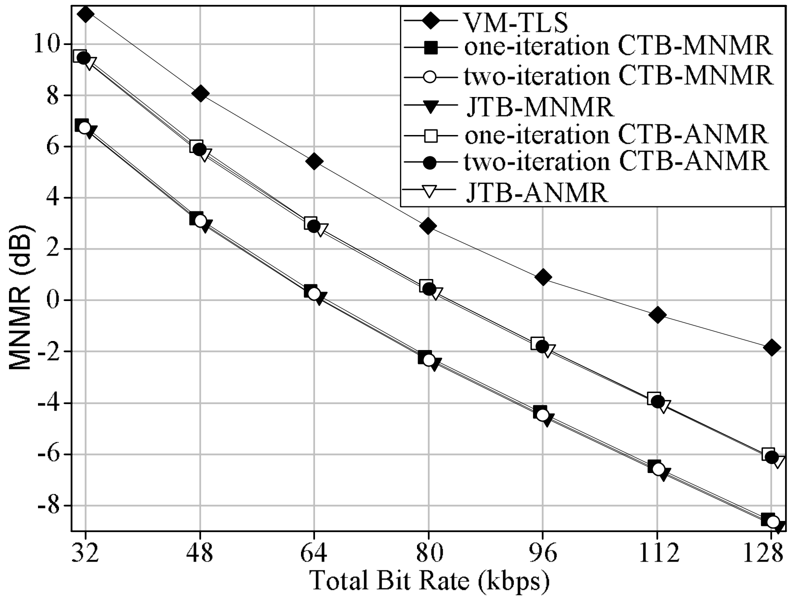


Fig. 3.8: MNMR rate-distortion comparison for VM-TLS, JTB and CTB.

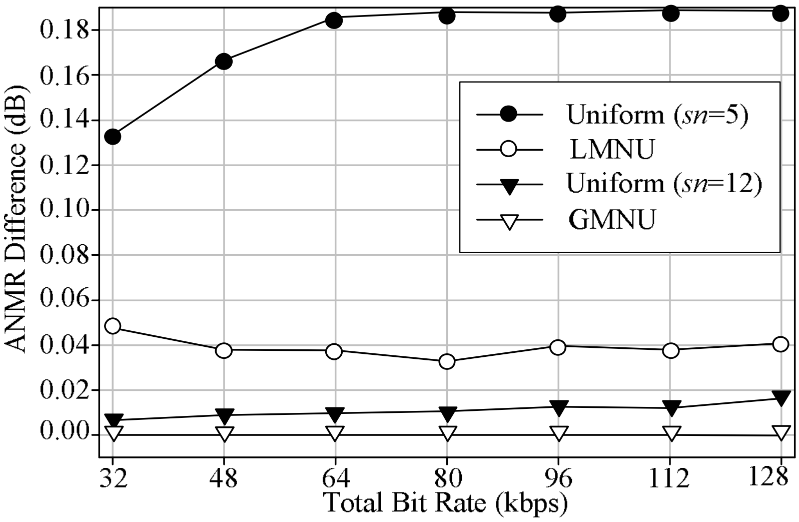


Fig. 3.9: ANMR differences between the full and fast SF search algorithms for CTB-MNMR.

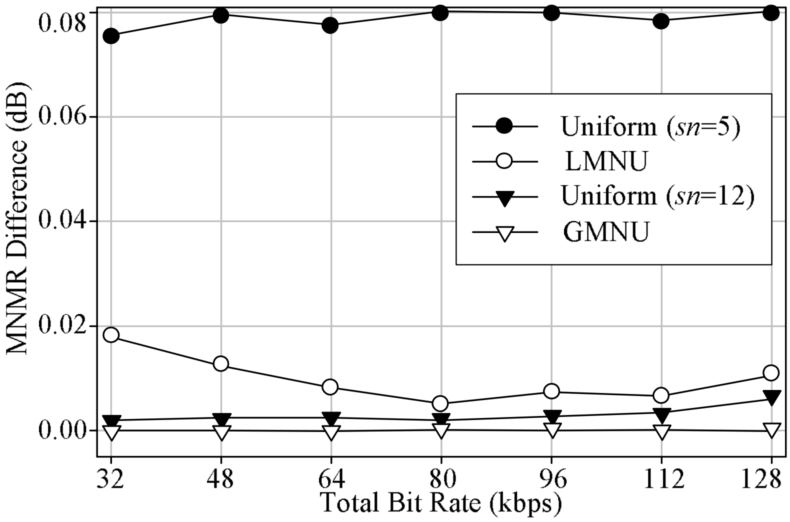


Fig. 3.10: MNMR differences between the full and fast SF search algorithms for CTB-MNMR.

3.3.3 Subjective Quality

Listening test by human ears is the traditional way to subjectively evaluate the audio quality and is also the most recognized subjective quality measure. However, such subjective test is expensive, time consuming, and difficult to reproduce. As described in Section 6.2 in [8], the subjective quality (mean opinion score, MOS) of the JTB-ANMR (or JTB-MNMR) scheme is significantly better than that of the VM-TLS. MOS is derived from the ITU (International Telecommunications Union) 5-grade absolute category rating (ACR) scheme [25]. Moreover, the informal listening tests on the aforementioned schemes show that it is hard to tell the difference between JTB and various CTB schemes. In addition, a "*simulated*" subjective measure, Objective Difference Grade (ODG), is used in audio quality evaluation. ODG is generated by a procedure designed to be comparable to the Subjective Difference Grade (SDG) judged by human ears. It is calculated based on the difference between the quality rating of the "reference" signal and the "test" signal. The ODG has a range of [-4, 0], in which –4 stands for very significant difference and 0 stands for imperceptible difference between the reference and the test signal [26][27].

The ODG results for various R-D control schemes discussed in this paper are shown in Fig. 3.11, in which the reference signal is the original audio sequence. According to the collected test data (Fig. 3.11), the difference between JTB and CTB schemes is quite small.

The ODG results, comparing against the full search CTB-MNMR scheme, are shown in Fig. 3.12. Note that the reference signal here is the coded audio sequence produced by the full search CTB-MNMR scheme. We can find that there is almost no difference between the GMNU algorithm and the full search CTB-MNMR particularly at mid to high bit rates. Again the performance of the non-uniform trellis algorithms is better than that of the uniform trellis algorithms with the same computational complexity.

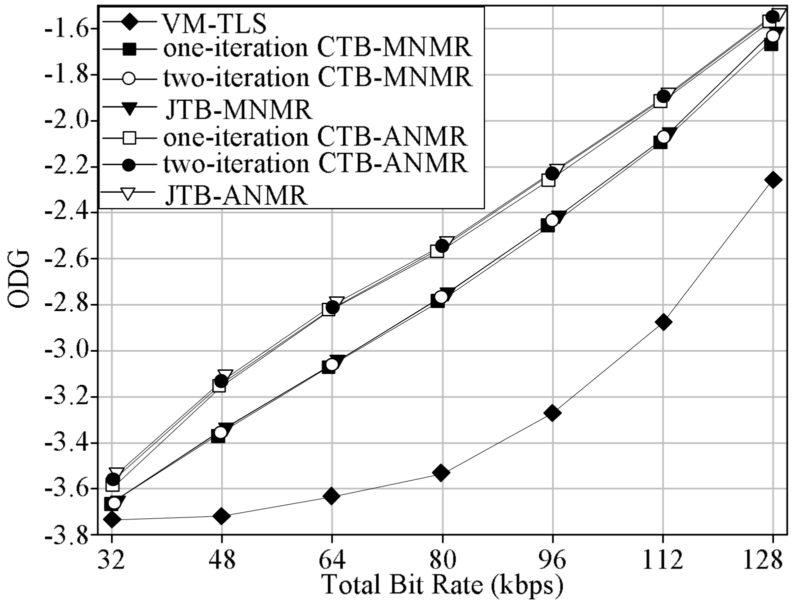


Fig. 3.11: ODG of VM-TLS, JTB and CTB.

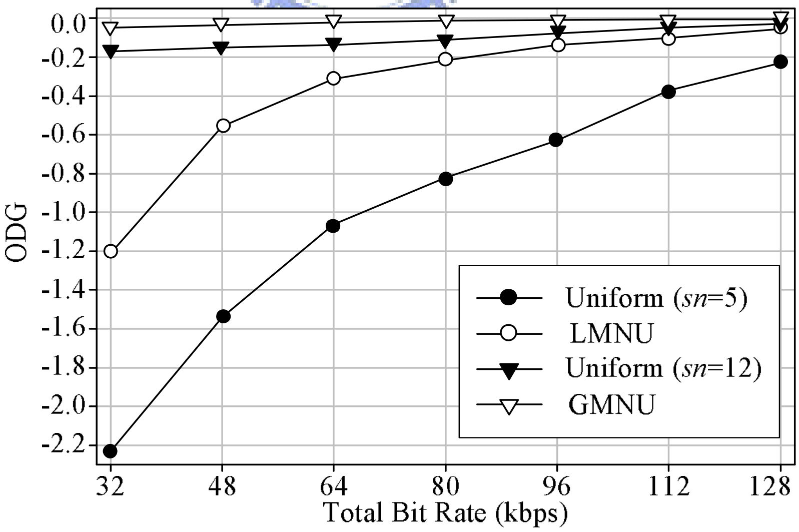


Fig. 3.12: ODG of various fast SF search algorithms for CTB-MNMR.

Chapter 4
Enhanced BFOS Bit Allocation Algorithm

How to make use of the bits more efficiently is always the key issue in perceptual audio coding. The traditional bit allocation strategies, “allocate bits to the band with the largest NMR” or “allocate bits to the bands of which the distortion is larger than the masking threshold” [23][28], do not necessarily provide the best bit-use efficiency. The “bit-use efficiency” here means the distortion improvement due to receiving bits.

The previously proposed generalized BFOS algorithm is an efficient bit allocation algorithm for classified vector quatization or subband coding [9][10]. The research in [11] also shows that the generalized BFOS algorithm is a near optimal bit allocation scheme for MPEG-1 Layer Ⅰ / Layer Ⅱ audio coding. However, our research shows that the generalized BFOS bit allocation algorithm becomes less efficient for MPEG-4 AAC in which the inter-band dependency of coding process exists.

In this chapter, we describe the proposed second type of R-D control algorithms, the Enhanced BFOS (EBFOS) bit allocation algorithm. We first describe briefly the generalized BFOS bit allocation algorithm in Section 4.1. The bit allocation procedure of our EBFOS scheme for AAC and its fast version are described in Section 4.2. For comparison, we also propose an approach to integrate the generalized BFOS bit allocation algorithm in AAC in Section 4.3. Finally, the complexity analysis and the simulation results are presented in Section 4.4.

4.1 Generalized BFOS Algorithm

As illustrated in [10], the generalized Breiman, Friedman, Olshen, and Stone (BFOS) algorithm is an extension of an algorithm for optimal pruning in tree-structured classification and regression to coding [29]. For a source coding application, it finds a sequence of nested subtrees of a given tree-structured coder. Bit allocation of the audio coder using the generalized BFOS algorithm is first suggested in [11].

As described in [10][11], the BFOS algorithm allocates the bits based on the "Marginal Returns" provide by each allocation. The marginal return, *MR*, is defined as (4.1) [11].

$$MR = \frac{\Delta D}{\Delta R} \tag{4.1}$$

ΔD and ΔR in (4.1) are the changes of distortion and bit rate, respectively. For a subband coding application, we restate the basic idea of the generalized BFOS bit allocation algorithm as "allocate bits to the band with the maximum distortion decrease per bit" or "de-allocate bits from the band with the minimum distortion increase per bit". However, as described in [10][11], the analytic procedure of "distortion decrease per bit" or "distortion increase per bit" is inter-band independent in the generalized BFOS bit allocation algorithm. This may result in less efficient bit allocation for the coder in which the inter-band dependence of coding process exists, such as AAC.

4.2 Enhanced BFOS Bit Allocation Algorithm for AAC (EBFOS)

In the proposed second type of R-D control algorithms, instead of performing the trellis search through entire frame, we allocate the bits to the proper band step by step. If the inter-band dependency of coding process does not exist, the bit allocation problem can be efficiently solved by the generalized BFOS algorithm. However, as described in Section 2.9, the differential and run-length coding induce the inter-band dependency in coding process of AAC**.**

Similar to the generalized BFOS bit allocation algorithm, the basic idea behind our bit allocation scheme is "allocate bits to the band with the maximum NMR gain per bit" or “retrieve bits from the band with the maximum bits per NMR loss”. However, in our approach, we also consider the inter-band dependency of coding process. Therefore, our bit allocation approach is called Enhanced BFOS algorithm. “NMR gain per bit” (NGPB) means the gain in NMR by allocating one bit and is formulated by (4.2). “bits per NMR loss” (BPNL) is the number bits we save if we give away one unit of NMR and is formulated by (4.3). In (4.2) and (4.3), NMR_{ref} and $bits_{ref}$ are the original NMR value and bit numbers, respectively. NMR_{new} is the new NMR value after allocating new bit numbers, $bits_{new}$. In principle, our proposed scheme tries to reduce the total NMR of all bands. Hence, it has a performance close to the algorithm that minimizes the averaged NMR criterion.

$$NMR\ \ Gain\ /\ bit = (NMR_{ref} - NMR_{new})/(bits_{new} - bits_{ref}) \tag{4.2}$$

$$bits\ /\ NMR\ \ Loss = (bits_{ref} - bits_{new})/(NMR_{new} - NMR_{ref}) \tag{4.3}$$

4.2.1 Bit Allocation Procedure of EBFOS Scheme

As illustrated in Section 2.9, in AAC, NMR in each band is controlled by the SF value. In general, larger SF value (referring to larger step size of the quantizer) will result in larger NMR value in each band. After been quantized, the quantized spectral coefficients (in each band) are entropy-coded by a proper choice of HCB. In addition, the indices of SFs and HCBs for all the bands are coded using differential and run-length codes respectively. The total coding bits, *TB*, for a frame can be expressed as (4.4). The values of SF and HCB for the *i*th SFB are denoted by s_i and h_i, respectively. Symbol *D*() and *R*() represent the number of bits produced by differential coding and run-length coding, respectively. Parameter b_i is the number of bits for coding the quantized spectral coefficients and parameter B_i is the total coding bits for the *i*th SFB.

$$TB = \sum_i B_i = \sum_i \left(b_i + D(s_i - s_{i-1}) + R(h_{i-1}, h_i)\right) \tag{4.4}$$

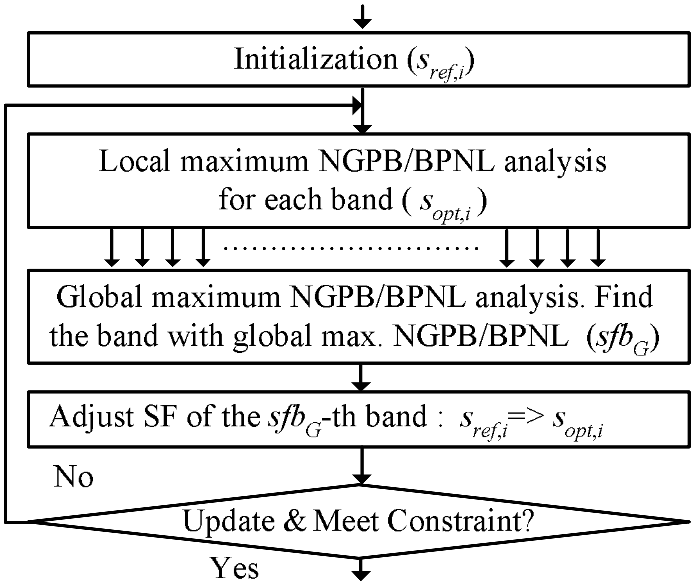

Initialization ($s_{ref,i}$)
Local maximum NGPB/BPNL analysis for each band ($s_{opt,i}$)
Global maximum NGPB/BPNL analysis. Find the band with global max. NGPB/BPNL (sfb_G)
Adjust SF of the sfb_G-th band : $s_{ref,i}$=> $s_{opt,i}$
No
Update & Meet Constraint?
Yes

Fig. 4.1: EBFOS bit allocation scheme.

The block diagram of the EBFOS bit allocation scheme is shown in Fig. 4.1. Each step in Fig. 4.1 is elaborated below.

1. *Initialization*. This step is to initialize the reference NMR for each band, $NMR_{ref,i}$, at the start-up of Maximum NGPB/ BPNL analysis. Then, we can determine the reference SF for each band, $s_{ref,i}$,and calculate the values of reference total coding bits for a frame, TB_{ref}, based on the adopted reference NMR value. In general, larger $NMR_{ref,i}$ value (at the start-up) will result in smaller TB_{ref} value at the start-up. There seems to be no theoretically optimal choice for these values. In our implementation, we set the reference NMR to 1 (0 dB) for all the bands, $\mathrm{NMR}_{ref,i}=1$, $\forall i$. In other words, we are targeting at perceptually lossless coding at the beginning of processing a frame.

2. *Local Maximum NGPB/BPNL analysis*. This step is to find the local maximum NGPB and BPNL values for all bands. We can determine the local maximum NGPB and BPNL of the ith SFB, denoted by $NGPB_{L,i}$ and $BPNL_{L,i}$, by computing:

$$NGPB_{L,i} = \max_{s_{new,i}}\left\{(NMR_{ref,i} - NMR_{new,i})/(TB_{new} - TB_{ref})\right\} \quad \forall s_{new,i}\ ,\ (s_{ref,i} - n_i) \le s_{new,i} < s_{ref,i} \tag{4.5}$$

and

$$BPNL_{L,i} = \max_{s_{new,i}} \left\{ (TB_{ref} - TB_{new})/(NMR_{new,i} - NMR_{ref,i}) \right\} \quad \forall s_{new,i} \text{ , } s_{ref,i} < s_{new,i} \le (s_{ref,i} + n_i) \tag{4.6}$$

TB_{new} and $NMR_{new,i}$ are the new value of total coding bits for a frame and new value of distortion for the *i*th SFB respectively, when the corresponding SF value of the *i*th SFB is changed from $s_{ref,i}$ to $s_{new,i}$. The local optimal SF value of the *i*th SFB, $s_{opt,i}$, is the SF value associated with the local maximum NGPB or BPNL. The n_i in (4.5) (or (4.6)) determines the candidate number of $s_{new,i}$, which is approximate to 12 on the average from the statistics of coded data.

Note that, in performing the local maximum NGPB or BPNL analysis for the *i*th SFB, only the SF value of the *i*th SFB is changed from $s_{ref,i}$ to $s_{new,i}$. The SF values of the other SFBs are kept unchanged ($s_j = s_{ref,j}$, $\forall j$, $j \neq i$).

3. *Global Maximum NGPB/BPNL analysis.* We first find the global maximum NGPB and BPNL value, $NGPB_G$ and $BPNL_G$, for a frame by computing:

$$NGPB_G = \max_i \left\{ NGPB_{L,i} \right\} \quad \forall i \text{ , } 1 \le i \le 49 \tag{4.7}$$

$$BPNL_G = \max_i \left\{ BPNL_{L,i} \right\} \quad \forall i \text{ , } 1 \le i \le 49 \tag{4.8}$$

The global optimal SFB, sfb_G, is the SFB that has $NGPB_G$ (or $BPNL_G$). Then we set the SF value only of the sfb_G*-th* SFB to the local optimal SF value of the sfb_G*-th* SFB.

4. Update $NMR_{ref,i}$ (as well as $s_{ref,i}$) of the sfb_G*-th* SFB and TB_{ref}. Go to step 2 if the bit budget constraint is not met.

In order to handle the inter-band dependency of coding process, we use *TB* instead of B_i for NGPB/BPNL analysis. Otherwise, the SF value change of the sfb_G*-th* SFB in step 3 will influences the local maximum NGPB/BPNL analyses of the other SFBs in step 2. Therefore, we have to performing the local maximum NGPB/BPNL analyses for all the bands for each iteration.

As mentioned in step 2 of the preceding procedure, after changing the SF value from $s_{ref,i}$ to $s_{new,i}$, we need to calculate TB_{new} and $NMR_{new,i}$. The value of $NMR_{new,i}$ depends only on the value of $s_{new,i}$. However, (4.4) indicates that the value of TB_{new} depends not only on the value of $s_{new,i}$; it also depends on the choice of HCB. In our bit allocation scheme, we adopt the trellis-based optimization algorithm for HCB decision proposed in Section 3.1.3.

In general, either the Maximum NGPB analysis or the Maximum BPNL analysis (but not both) has to be performed for each iteration. The Maximum NGPB analysis is used for spending the bit budget (when the bit budget is positive) and the Maximum BPNL analysis is used for recovering the bit budget (when the bit budget is negative).

4.2.2 Fast Algorithm for EBFOS Scheme

The complexity of our EBFOS scheme highly depends on the times that the NGPB/BPNL calculation in step 2 of the bit allocation procedure (in Section 4.2.1) is performed. Taking the local maximum NGPB analysis as example, we need to perform n_i times NGPB calculation for locating the local maximum NGPB of the ith SFB. Hence, the total number of calculations for finding the global maximum NGPB is $\sum_i n_i$.

It is obvious that the most effective way for reducing computations is to reduce the number of NGPB/BPNL calculations. From the statistics of the local optimal parameters, $s_{opt,i}$ and $NGPB_{L,i}$ (or $BPNL_{L,i}$) collected from the coded data, we find some interesting properties whish are summarized in Table 4.1.

In Table 4.1, i is the SFB index and m is the index of SF adjustment iteration. sfb_G^m is the global optimal SFB of the mth SF adjustment iteration and $S = \{sfb_G^m - 1, sfb_G^m + 1\}$, the set of two SFBs. The first statistic in Table 4.1 is the probability that $s_{opt,i}^{m+1}$ differs from $s_{opt,i}^m$

and is denoted as $P_C\,(s^{m+1}_{opt,i} \neq s^m_{opt,i})$, where $s^m_{opt,i}$ is the local optimal SF value of *i*th SFB for the *m*th SF adjustment iteration. We can find that $P_{i\in S}\,(s^{m+1}_{opt,i} \neq s^m_{opt,i})$ reaches to 28.72% which is approximate 18 times of $P_{i\notin(S\cup sfb^m_G)}\,(s^{m+1}_{opt,i} \neq s^m_{opt,i})$. The other statistic, taking the Maximum NGPB analysis as example, is the average value of normalized differences between $NGPB^m_{L,i}$ and $NGPB^{m+1}_{L,i}$, $AD_C\,(NGPB^m_{L,i}, NGPB^{m+1}_{L,i})$, where $NGPB^m_{L,i}$ is the local maximum NGPB value of *i*th SFB for the *m*th SF adjustment iteration. $AD_C\,(NGPB^m_{L,i}, NGPB^{m+1}_{L,i})$ is formulated in (4.9), where *M* is 2 for $C = (i \in S)$ and is (49-3) for $C = (i \notin (S \cup sfb^m_G))$. We can find that, $AD_{i\notin(S\cup sfb^m_G)}(NGPB^m_{L,i}, NGPB^{m+1}_{L,i})$ is typically quite small, but $AD_{i\in S}(NGPB^m_{L,i}, NGPB^{m+1}_{L,i})$ is much larger.

$$AD_C\,(NGPB^m_{L,i}, NGPB^{m+1}_{L,i}) = \frac{1}{M} \times \sum_C \left(\frac{abs(NGPB^{m+1}_{L,i} - NGPB^m_{L,i})}{NGPB^{m+1}_{L,i}} \right) \tag{4.9}$$

It is clearly that the differences of local maximum NGPB/BPNL analyses between each SF adjustment iteration mainly centralize at the SFB belong to *S*. Using these properties, we can drastically reduce the number of iterations in determining *sfb*$_G$ and *NGPB*$_G$ (or *BPNL*$_G$). We only need to perform the local maximum NGPB/BPNL analysis on three SFBs ($SFB = \{sfb^m_G - 1, sfb^m_G, sfb^m_G + 1\}$) after the first SF adjustment iteration. This is the *fast version* of our EBFOS algorithm.

Table 4.1: Statistics of the local optimal parameters in maximum NGPB/BPNL analysis.

Condition (C)	$i \in S$	$i \notin (S \cup sfb^m_G)$
$P_C\,(s^{m+1}_{opt,i} \neq s^m_{opt,i})$	28.72%	1.60%
$AD_C\,(NGPB^m_{L,i}, NGPB^{m+1}_{L,i})$ or $AD_C\,(BPNL^m_{L,i}, BPNL^{m+1}_{L,i})$	0.7428	0.0211

4.3 Generalized BFOS Bit Allocation Algorithm for AAC

The generalized BFOS algorithm is an efficient bit allocation algorithm for subband coding. For the purpose of analyses and comparisons, we propose an approach to integrate the generalized BFOS bit allocation algorithm in AAC in this section based on the concepts described in [10] and [11]. The bit allocation procedure of the generalized BFOS scheme for AAC is similar to that of the EBFOS scheme (see Fig. 4.1). Each step in the generalized BFOS scheme for AAC is elaborated below.

1. *Initialization.* The same to the initialization step in Section 4.2.1, we set the reference NMR to 1 (0 dB) for all the bands. Then, we determine the $s_{ref,i}$ value and calculate the value of reference total coding bits for each band, $B_{ref,i}$ based on the adopted reference NMR value, NMR$_{ref,i}$=1, $\forall i$.
2. *Local Maximum NGPB/BPNL analysis.* Differing from the EBFOS scheme, the local maximum NGPB and BPNL of the ith SFB for the BFOS scheme are determine by the formula (4.10) and (4.11) respectively.

$$NGPB_{L,i} = \max_{s_{new,i}}\left\{(NMR_{ref,i} - NMR_{new,i})/(B_{new,i} - B_{ref,i})\right\} \quad \forall s_{new,i}\ ,\ (s_{ref,i} - n_i) \le s_{new,i} < s_{ref,i} \tag{4.10}$$

and

$$BPNL_{L,i} = \max_{s_{new,i}}\left\{(B_{ref,i} - B_{new,i})/(NMR_{new,i} - NMR_{ref,i})\right\} \quad \forall s_{new,i}\ ,\ s_{ref,i} < s_{new,i} \le (s_{ref,i} + n_i) \tag{4.11}$$

$B_{new,i}$ and $NMR_{new,i}$ are the new values of total coding bit and distortion for the *i*th SFB respectively, when the corresponding SF value of the *i*th SFB is changed from $s_{ref,i}$ to $s_{new,i}$. The local optimal SF value of the *i*th SFB, $s_{opt,i}$, is the SF value associated with the local maximum NGPB or BPNL.

3. *Global Maximum NGPB/BPNL analysis.* The same to the step 3 in Section 4.2.1, we first find the $NGPB_G$ (or $BPNL_G$) for a frame by the formula (4.7) (or (4.8)) and determine sfb_G . Then we set the SF value only of the sfb_G*-th* SFB to the local optimal SF value of the sfb_G*-th* SFB.
4. Update $NMR_{ref,i}$ (as well as $s_{ref,i}$) and $B_{ref,i}$ of the sfb_G*-th* SFB. Go to step 2 if the bit budget constraint is not met.

In the generalized BFOS bit allocation scheme here, we also adopt the trellis-based optimization algorithm for HCB decision. However, differing from the EBFOS scheme, we only perform the local maximum NGPB/BPNL analysis for the sfb_G*-th* SFB.

As described in [10], the generalized BFOS bit allocation scheme can be performed with and without convexity assumption. When the generalized BFOS scheme is performed with convexity assumption, n_i in (4.10) (or (4.11)) is equal to 1. When the generalized BFOS scheme is performed without convexity assumption, n_i is approximate to 14 on the average from the statistics of coded data.

4.4 Simulation Results

In this section, we evaluate the computational complexity and the coded audio quality in our experiments. Four types of bit allocation algorithms are simulated and compared as described below using the MPEG-4 AAC Verification Model (VM) as the test platform.

(1) The TLS algorithm in MPEG-4 AAC VM (VM-TLS).

(2) The BFOS algorithm for AAC with convexity assumption, BFOS-C, and without convexity assumption, BFOS-NC, which are described in Section 4.3.

(3) The trellis-based algorithm aiming at minimizing average NMR, JTB-ANMR, and aiming at minimizing maximum NMR, JTB-MNMR, which are described in [7] and [8].

(4) The EBFOS scheme and its fast version, which are described in Section 4.2.

In order to focus only on the bit allocation performance, all the optional tools in AAC, such as TNS and M/S stereo coding, are not used in our simulations. Ten two-channel audio sequences with a sampling rate at 44.1 kHz are tested. Two of them are extracted from MPEG SQAM [6], and the rest are from EBU [24].

4.4.1 Complexity Analysis

The complexity analysis for the aforementioned several bit allocation algorithms is summarized in Table 4.2. The "Computation" column is the average number of NGPB (or BPNL) calculation for a frame. The values in "Computation" column are derived from the statistics collected from the simulations on audio sequences. For the convenience of comparison, the BFOS-NC scheme is chosen to be the reference (ratio=1) and all the other schemes are rated based on this reference.

Table 4.2: Complexity Analysis of EBFOS scheme and generalized BFOS scheme

Scheme	Computation	Ratio
BFOS-C	119	0.27
BFOS-NC	444	1
Fast EBFOS	1145	2.58
EBFOS	11848	26.68

The experimental data indicate that the computation of fast EBFOS scheme is approximately 2.6 times higher than that of the BFOS-NC scheme. Moreover, the fast EBFOS scheme is approximately 10 times faster than that of the EBFOS scheme.

4.4.2 Objective Quality

The rate-distortion curves of the aforementioned bit allocation schemes are shown in Fig. 4.2 and Fig. 4.3. Two common objective quality measurements, average NMR (ANMR) and maximum NMR (MNMR) are adopted in the objective performance comparison.

The research in [11] shows that the BFOS-C scheme is a near optimal bit allocation scheme for MPEG-1 LayerⅠ/ LayerⅡ audio coding, but the simulation results show that the BFOS-C scheme becomes less efficiency for AAC. The performance of the BFOS-NC scheme is much better than that of the BFOS-C scheme which means that the convex assumption is not suitable for AAC. Otherwise, both the ANMR and MNMR performances of the BFOS-NC scheme are approximately 1dB worse than that of the JTB-ANMR scheme.

Clearly, the performances of the EBFOS scheme are much better than that of VM-TLS and better than that of the BFOS-NC scheme. If we look at the ANMR plot (Fig. 4.2), the performance of the EBFOS scheme is slightly worse than that of JTB-ANMR but they are very close. It is somewhat better than the JTB-MNMR scheme since the latter is not optimized for the ANMR criterion. If we look at the MNMR plot (Fig. 4.3), the EBFOS scheme is somewhat worse than JTB-MNMR but it is slightly better than the JTB-ANMR scheme. As stated earlier, the EBFOS scheme is aiming at reducing the overall NMR, which pretty much leads to minimizing ANMR. As for the fast version, there is almost no loss of performance (less than 0.06dB loss) in adopting the fast algorithm for EBFOS.

4.4.3 Subjective Quality

The informal listening tests on the aforementioned schemes show that it is hard to tell the difference between JTB-ANMR and the EBFOS scheme. In addition, a "*simulated*" subjective measure, Objective Difference Grade (ODG), is used in audio quality evaluation.

The ODG results of the aforementioned bit allocation schemes are shown in Fig. 4.4, in which the reference signal is the original audio sequence. Interestingly, JTB-ANMR is the best algorithm judged by ODG. According to the collected test data (Fig. 4.4), the EBFOS scheme is better than that of the BFOS-NC and BFOS-C schemes. Moreover, the difference between the EBFOS and the JTB-ANMR schemes is rather small.

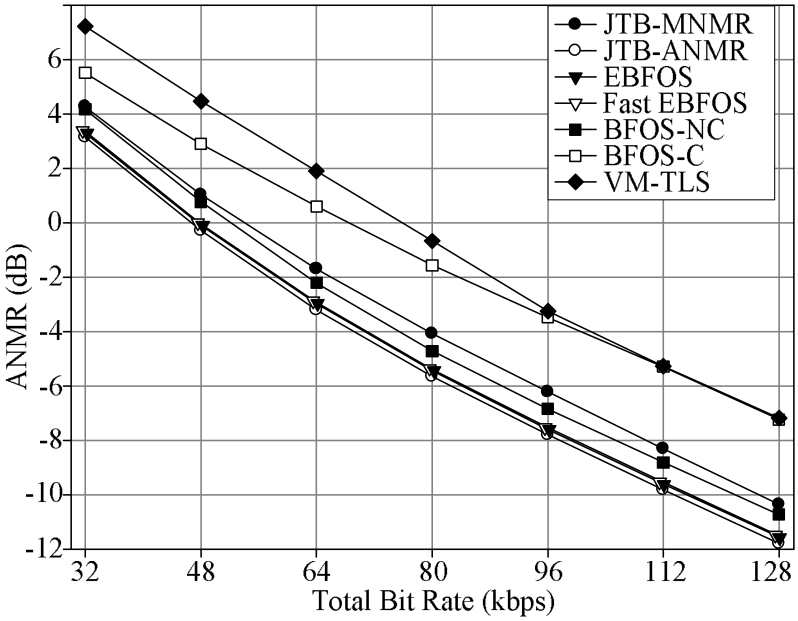


Fig. 4.2: ANMR rate-distortion comparison for various bit allocation schemes

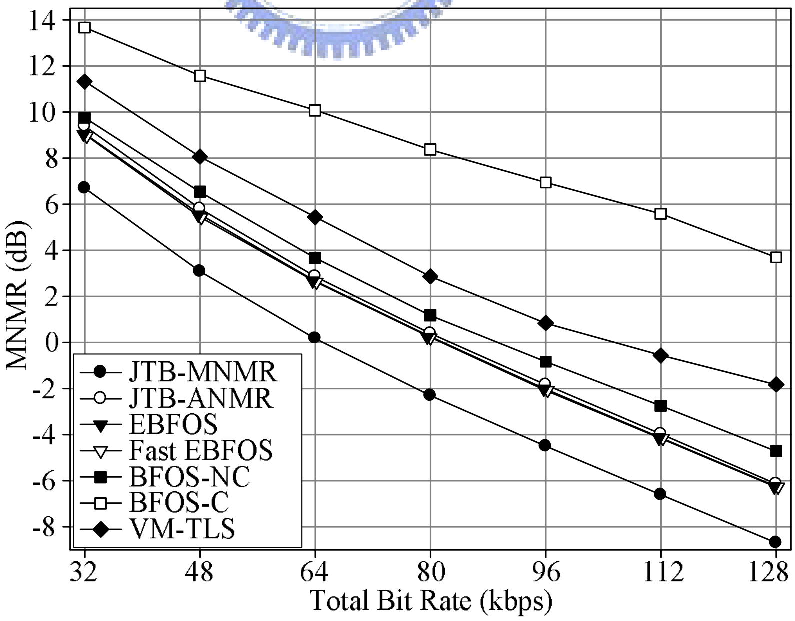


Fig. 4.3: MNMR rate-distortion comparison for various bit allocation schemes

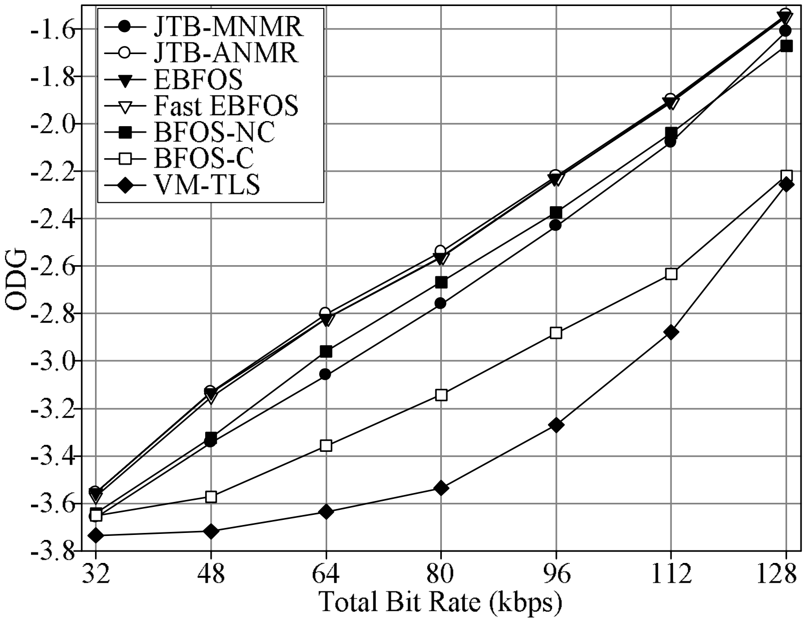


Fig. 4.4: ODG performance of various bit allocation schemes

Chapter 5 Perceptually Weighted Inter-Channel Prediction

Despite the success of current audio coding techniques, not much effort has been made to reduce the inter-channel redundancy inherent in multichannel audio. In order to achieve a higher efficiency in removing the inter-channel redundancy and, in the meanwhile, to maintain good audio quality, an efficient inter-channel prediction algorithm, called *perceptually weighted inter-channel prediction*, is described in this chapter.

We first give a brief review of linear prediction technique in Section 5.1. The proposed perceptual-weight inter-channel prediction scheme is described in Sections 5.2. The experiments and simulation results are summarized in Section 5.3.

5.1 Linear Prediction

Linear prediction technique is an effective tool in speech coding and lossless audio coding and thus is commonly used in those coders [30]. However, as mentioned in [31], there has been a long debate whether or not the inter-channel linear prediction in time domain can further increase the compression rate of the multichannel audio coder. Theoretically, the coder should achieve a higher compression rate by coding the prediction residual signals rather than the original signals. However, the research in [31] shows that the whitening effect of the prediction filter would increase the bit count in high frequency regions. Thus, the inter-channel prediction in time domain is often not applicable to general multichannel perceptual audio coder.

Although “time domain” linear prediction may not be effective in removing inter-channel redundancy in perceptual audio coder, the “transform domain” linear prediction techniques may offer some coding advantage. An efficient inter-channel redundancy removal algorithm based on the “transform domain” linear prediction is developed in this paper and is described in the following sections. Note that the "time domain" linear prediction here refers to performing linear prediction in the time domain (as shown in Fig. 5.1(a)) and the "transform domain" linear prediction here refers to producing prediction residuals in the transform domain (as shown in Fig. 5.1(b)).

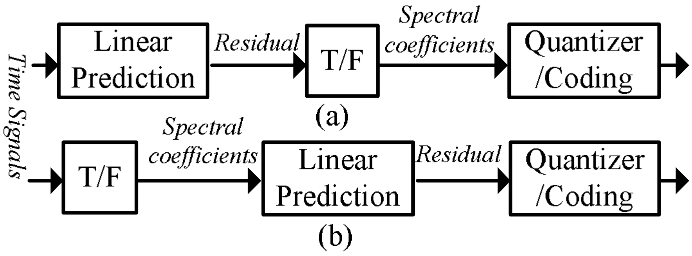


Fig. 5.1: Block diagram of (a) "time domain" linear prediction, and (b) "transform domain" linear prediction

5.2 Perceptually Weighted Inter-Channel Prediction (PW-ICP)

Our inter-channel redundancy removal algorithm uses the linear prediction in the transform domain. It is in a way similar to the operation of the Long Term Prediction (LTP) tool in MPEG-4 AAC [2]. However, different from LTP, the predicted signals are constructed from the quantized data of the other channel rather than the intra-channel data. The general structure of a perceptual audio coder with inter-channel prediction is shown in Fig. 5.2.

The "T/F" and "F/T" modules in Fig. 5.2 are the "Time to Frequency Transform" and the "Frequency to Time Transform" operations, respectively. The time-domain audio signals, $x(i)$, are first converted to their frequency-domain representation (spectral coefficients), $X(k)$, by the "T/F" operation. Motivated by the human auditory system, these spectral coefficients are grouped into a number of bands and then they are fed into the "Inter-Channel Prediction" module. The "Psychoacoustic Model" module calculates the perceptual masking threshold, which serves as the base for controlling the "Quantize/Coding" module. The highlighted (enclosed) region in Fig. 5.2 is the virtual decoder, which reconstructs all the necessary quantized data for prediction purpose, such as quantized spectral coefficients, $\tilde{X}(k)$, and quantized time signals, $\tilde{x}(i)$. In the encoding process, the optimal inter-channel predictor is first derived from the data. Then, we calculate the prediction signals, $\hat{X}(k)$, and the associated prediction residuals, $R(k)$. Finally, the prediction residuals are quantized and transmitted with side information, which may contain the predictor control (on/off) information and predictor parameters.

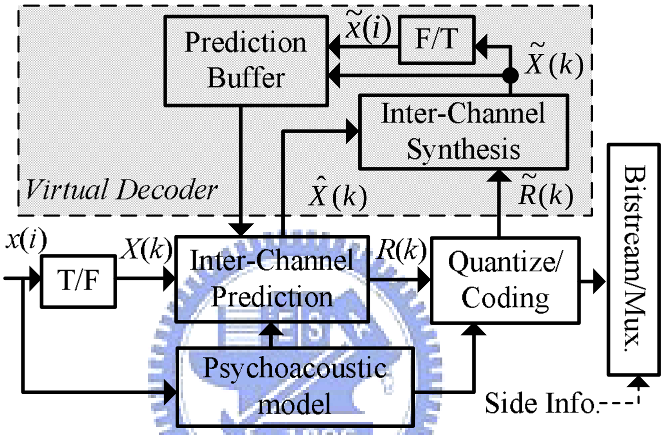


Fig. 5.2: Block diagram of perceptual audio coder with inter-channel prediction

5.2.1 Inter-Channel Predictor

One major step in any linear prediction based coding scheme is calculating the prediction signals. In this paper, two types of inter-channel predictor are proposed, the Time-Signal based predictor (TSP) and the Spectral-Coefficient based predictor (SCP). These two types of predictors are incorporated into the MPEG AAC system to perform inter-channel prediction and they are described in the following sub-sections.

1) Time-Signal Based Predictor (TSP):

In the Time-Signal based predictor, the time-domain estimates are first constructed and then transferred to their spectral representation. The block diagram of the Time-Signal based predictor is shown in Fig. 5.3.

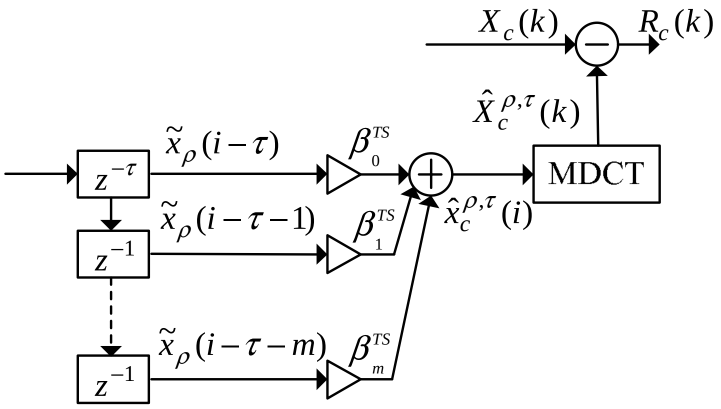


Fig. 5.3: Block diagram of Time-Signal based predictor

The time-domain estimates, $\hat{x}_c^{\rho,\tau}(i)$, and their spectral coefficient representations, $\hat{X}_c^{\rho,\tau}(k)$, for the c-th channel are constructed by an (m+1)-tap predictor as shown in formula (5.1).

$$\begin{aligned}&\hat{x}_c^{\rho,\tau}(i)=\sum_{l=0}^{m}\beta_l^{TS}\cdot\tilde{x}_\rho(i-\tau-l),\ i=0,\ldots,2N-1\\&\hat{\boldsymbol{X}}_c^{\rho,\tau}=\mathrm{MDCT}(\hat{\boldsymbol{x}}_c^{\rho,\tau}),\ \hat{\boldsymbol{x}}_c^{\rho,\tau}=[\hat{x}_c^{\rho,\tau}(0)\ldots\hat{x}_c^{\rho,\tau}(2N-1)],\ \hat{\boldsymbol{X}}_c^{\rho,\tau}=[\hat{X}_c^{\rho,\tau}(0)\ldots\hat{X}_c^{\rho,\tau}(N-1)]\end{aligned}\qquad(5.1)$$

The notation “MDCT( )” represents the Modified Discrete Cosine Transform in AAC and N is 1024. The predictor coefficients β_l^{TS} are determined by minimizing the mean square error, $E\left[e_c^2(i)\right]$.

$$E\left[e_c^2(i)\right]=E\left[\left(x_c(i)-\hat{x}_c^{\rho,\tau}(i)\right)^2\right],\ i=0,\ldots,2N-1\qquad(5.2)$$

2) Spectral-Coefficient Based Predictor (SCP):

In the Spectral-Coefficient based predictor, the prediction operation is performed on the spectral coefficients. The block diagram of the Spectral-Coefficient based predictor is shown in Fig. 5.4.

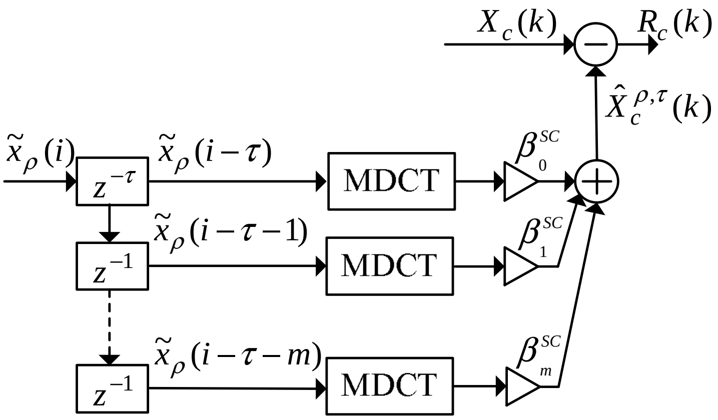


Fig. 5.4: Block diagram of Spectral-Coefficient based predictor

The spectral coefficient estimates, $\hat{X}_c^{\rho,\tau}(k)$, for the c-th channel are constructed by an (m+1)-tap predictor as shown in formula (5.3).

$$\begin{aligned}&\hat{X}_c^{\rho,\tau}(k)=\sum_{l=0}^{m}\beta_l^{SC}\cdot\tilde{X}_\rho^{\tau+l}(k),\,k=0,\ldots,N-1 \quad , \quad \tilde{\boldsymbol{X}}_\rho^{\tau+l}=\mathrm{MDCT}\left(\tilde{\boldsymbol{x}}_\rho^{\tau+l}\right),\\ &\tilde{\boldsymbol{x}}_\rho^{\tau+l}=[\tilde{x}_\rho(0-\tau-l)\ldots\tilde{x}_\rho(2N-1-\tau-l)] \quad , \quad \tilde{\boldsymbol{X}}_\rho^{\tau+l}=[\tilde{X}_\rho^{\tau+l}(0)\ldots\ldots\tilde{X}_\rho^{\tau+l}(N-1)]\end{aligned} \tag{5.3}$$

Different from β_l^{TS}, the predictor coefficients of the Spectral-Coefficient based predictor of the c-th channel, β_l^{SC}, are determined by minimizing the perceptually weighted mean square error, $E\left[e_c^2(sb)\cdot PW_c(sb)\right]$.

$$E\left[e_c^2(sb)\cdot PW_c(sb)\right]=E\left[\left(\sum_{k=w_{sb}}^{w_{sb+1}-1}\left(X_c(k)-\hat{X}_c^{\rho,\tau}(k)\right)^2\right)\cdot PW_c(sb)\right] \tag{5.4}$$

, where *sb* is the band index and w_{sb} is the first coefficient index of the *sb*-th band. $PW_c(sb)$ in (5.4) is the perceptual weight for the *sb*-th band (of the *c*-th channel) and is equal to the inverse of the masking threshold derived from the psychoacoustic model.

In (5.1) and (5.3), the ρ-th channel quantized signals, $\tilde{x}_\rho(i-\tau-l)$ or $\tilde{X}_\rho^{\tau+l}(k)$, include a time delay, τ, when used in estimation. The selections of parameters, τ and ρ, are to be discussed in Section 5.2.2.

5.2.2 Minimum Perceptually Weighted Prediction Error Analysis

The maximum correlation analysis between the original time signals, $x_c(i)$, and the prediction, $\hat{x}_c(i)$, over the entire frame as shown in formula (5.5) is a commonly used method for determining τ and ρ. This method may be suitable for encoding the time signals, such as speech and lossless audio coders. But we found a better index in transform domain as described below, which would result in a higher coding efficiency for the perceptual audio coders.

$$(\rho,\tau) = \arg\max_{\rho',\tau'} \left(\sum_i \left(\hat{x}_c^{\rho',\tau'}(i) \cdot x_c(i) \right) \Big/ \sum_i \hat{x}_c^{\rho',\tau'}(i)^2 \right) \tag{5.5}$$

We are motivated by the following observations. First, the critical operations in perceptual audio coding, such as Quantization/Entropy-Coding, are performed on the spectral coefficients. To establish a direct link between these operations and the optimization index, it seems to be more appropriate to conduct the correlation analysis on the spectral coefficients. Second, in a typical perceptual audio coder (such as MPEG audio), the spectral coefficients are grouped into a number of bands. The spectral coefficients belonging to the same band are quantized and entropy-coded using the same parameter set. Different bands can have different quantizers and/or entropy coding codebooks. Therefore, we expect a higher de-correlation performance by conducting the correlation analysis for each band separately. Third, in general, the number of bits for coding the spectral coefficients in a band depends not only on the magnitude of spectral coefficients; it also depends on the perceptual masking threshold of that band. Therefore, the perceptual masking threshold should be incorporated in the analysis. Base on the preceding observations, we design a new optimization index, called *minimum perceptually weighted prediction error*, to determine τ and ρ as shown in (5.6). The prediction error in each band is first calculated and then summed up with perceptual weights.

$$\left(\rho,\tau\right)=\underset{\rho',\tau'}{\arg\min}\left(\sum_{sb}\left(\left(\sum_{k=w_{sb}}^{w_{sb+1}-1}\left(X_c\left(k\right)-\hat{X}_c^{\rho',\tau'}\left(k\right)\right)^2\right)\cdot PW_c\left(sb\right)\right)\right) \tag{5.6}$$

, where *sb*, w_{sb}, and $PW_c(sb)$ are defined in (5.4).

Simulations and comparison between the traditional maximum correlation method and our proposed minimum perceptually weighted prediction error method will be given in Section 5.3.1.

5.2.3 Predictor Order

In the previous sub-sections, we have described some of the important parameters of the predictors (in (5.1) and (5.3)), such as β_l^{TS}, β_l^{SC}, τ and ρ. In this section, another critical parameter, the predictor order, will be discussed. In analyzing the impact of the predictor order on the aforementioned two types of predictors, two test audio sequences, TAS1 and TAS2, are tested. As shown in the following discussions in this section, these two sequences show clearly different characteristics when the predictor order varies.

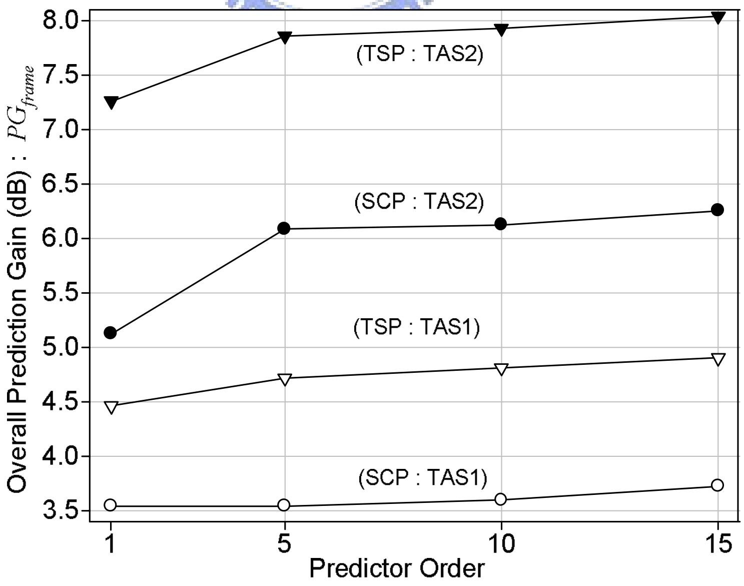


Fig. 5.5: Overall prediction gain vs. predictor order

In general, increasing the predictor order would decrease the overall energy of the prediction residuals, and thus leads to an increase in prediction gain as shown by the simulation results in Fig. 5.5. The prediction gain here is defined as the ratio of the original signal energy (power) to the prediction residual energy [31]. The overall prediction gain, PG_{frame}, is thus defined by (5.7).

$$PG_{frame} = \sum_{k=0}^{N-1} \left(X_c(k)\right)^2 \Big/ \sum_{k=0}^{N-1} \left(X_c(k) - \hat{X}_c^{\rho,\tau}(k)\right)^2 \tag{5.7}$$

, where N is the length of a frame and is equal to 1024 in AAC.

It is often believed that a smaller residual energy or, equivalently, a larger prediction gain would often leads to a smaller bit rate in coding the prediction residuals. Therefore, a high-order predictor (order≥10) is commonly adopted in speech and lossless audio coders. As shown in Fig. 5.5, the high-order predictor is expected to have higher bit rate reduction and the performance of TSP is expected to be better than that of SCP. However, the following analyses show that the coding bit rate is also affected by the spectral distribution of residuals, especially for the subband audio coders. Therefore, we re-examine the prediction gain band by band. The prediction gain of a band, PG_{band}, is defined by (5.8) and the simulation results for different predictor order are shown in Fig. 5.6 and Fig. 5.8.

$$PG_{band}(sb) = \sum_{k=w_{sb}}^{w_{sb+1}-1} \left(X_c(k)\right)^2 \Big/ \sum_{k=w_{sb}}^{w_{sb+1}-1} \left(X_c(k) - \hat{X}_c^{\rho,\tau}(k)\right)^2 \tag{5.8}$$

Fig. 5.6 shows that a high-order TSP often increases the energy in the higher frequency bands although the low frequency energy and the total energy are reduced. Because the energy of high-frequency bands is much lower, if we look at the coding gain, the prediction gain increase at low frequency of the high-order predictor is small but the prediction gain loss at high frequency is rather large. Therefore, as shown in Fig. 5.7, the bit rate reduction of a high-order TSP is less than that of a low-order TSP in the higher frequency bands. As a consequence, the 1-Tap TSP performs better for sequence TAS1. For TAS2, Fig. 5.8 and Fig.

5.9 show that the performance (bit rate reduction) is obviously improved due to the order of TSP increase from 1 to 5. The experiments in the above show that (a) PG_{band} is better correlated to the coding bit rates, and (b) low-order TSPs can at times perform better than the high-order TSPs. Thus, an adaptive-order TSP may be more appropriate for coding purpose.

We also show the coding gain and bit rate reduction of SCP in Fig. 5.6 to Fig. 5.9. We find that although PG_{frame} of SCP is less than that of TSP, the SCP has better efficiency in terms of PG_{band} at higher frequency bands. Therefore, SCP overall produces a better performance. In addition, different prediction orders of SCP seem to produce similar bit rate reduction performance as long as the order is not too small.

From the analyses described above, we find that, the choice of predictor order has a strong impact on the bit rate reduction performance. More simulations and analyses on the predictor order for the TSP and SCP will be further discussed in Section 5.3.2.

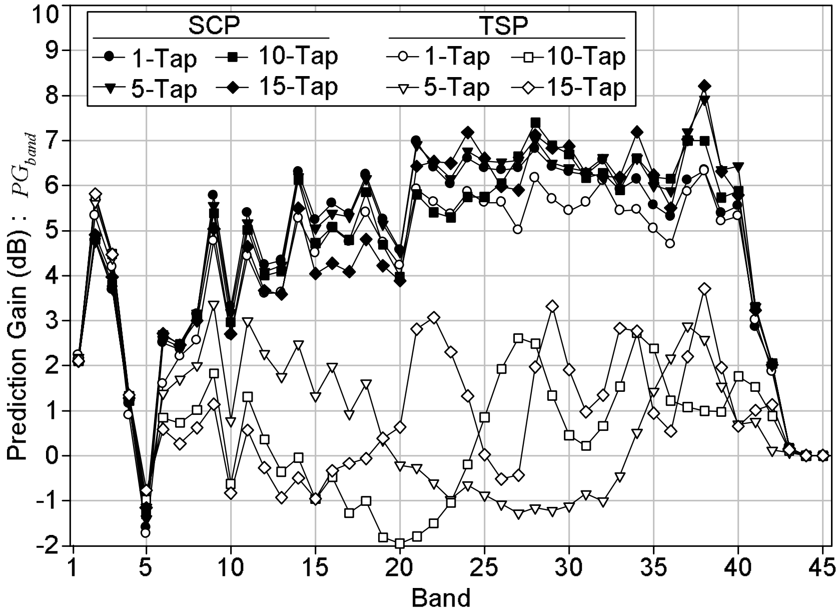


Fig. 5.6: Prediction gain (PG_{band}) vs. predictor order for individual band for sequence TAS1

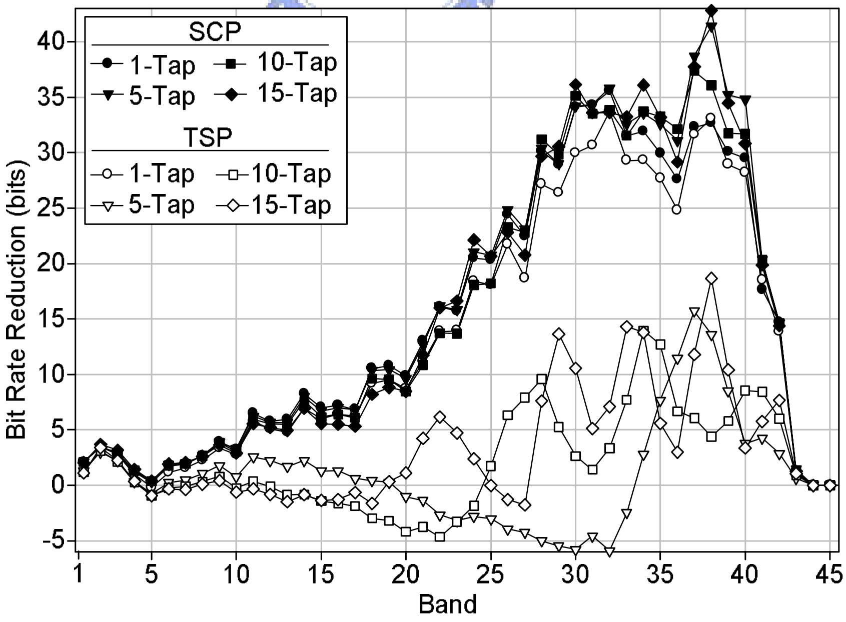


Fig. 5.7: Bit rate reduction vs. predictor order for individual band for sequence TAS1

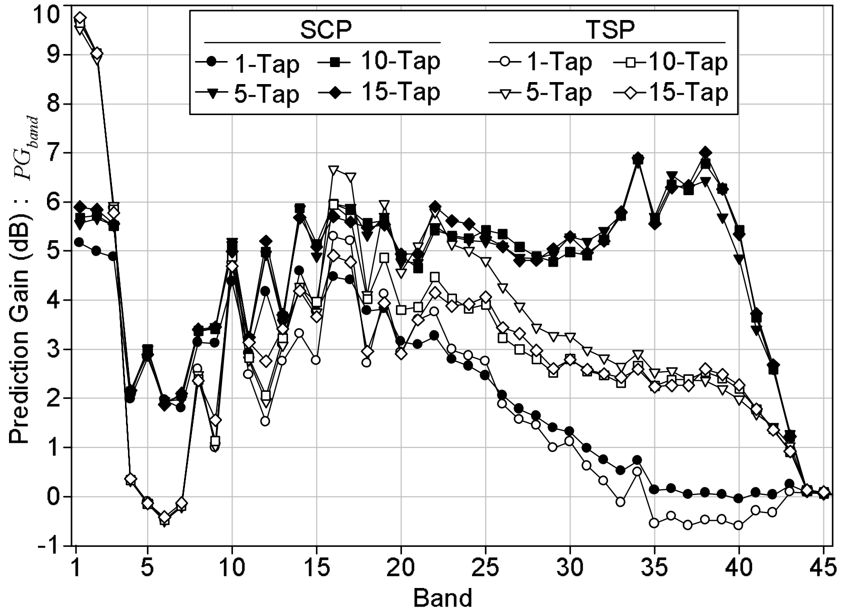


Fig. 5.8: Prediction gain (PG_{band}) vs. predictor order for individual band for sequence TAS2

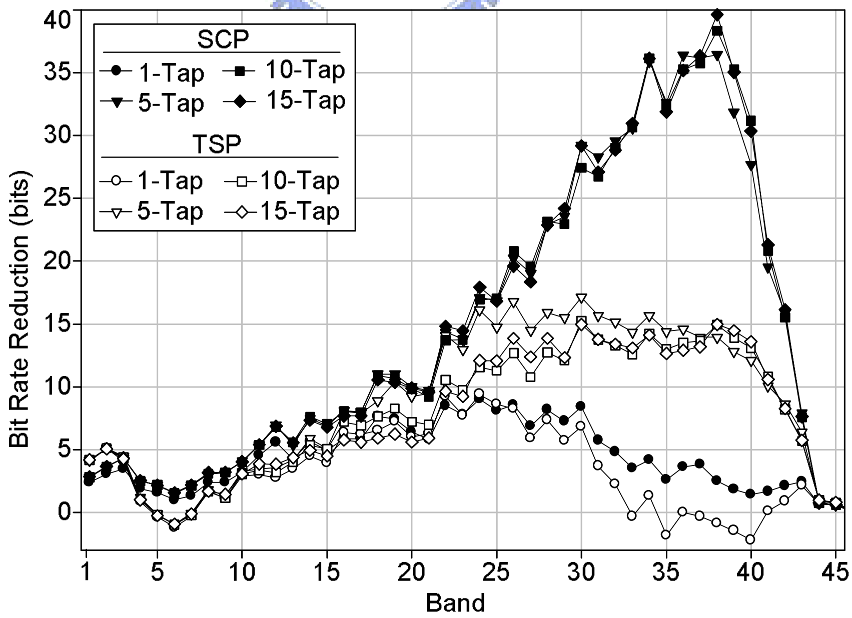


Fig. 5.9: Bit rate reduction vs. predictor order for individual band for sequence TAS2

5.2.4 Perceptual Masking Control

As mentioned earlier in the Introduction section, some inter-channel redundancy removal algorithms, such as M/S coding and KLT based approach, require extra perceptual masking checking on the (M/S or KLT) “transformed” coefficients. However, there is no simple perceptual model is these transformed domains. In this section, we will show that our PW-ICP scheme does not require extra perceptual masking checking.

After determining the optimal time delay, τ, and the optimal predictive channel index, ρ, we can compute the prediction of spectral coefficient, $\hat{X}_c^{\rho,\tau}(k)$. The residual spectral coefficients, $R_c(k)$, are calculated by (5.9), and the quantized spectral coefficient, $\tilde{X}_c(k)$, can then be constructed using (5.10).

$$R_c(k) = X_c(k) - \hat{X}_c^{\rho,\tau}(k) \tag{5.9}$$

$$\tilde{X}_c(k) = \hat{X}_c^{\rho,\tau}(k) + \tilde{R}_c(k) \tag{5.10}$$

, where $\tilde{R}_c(k)$ is the quantized residual spectral coefficient. Using these notations, the total quantization noise of the original spectral coefficients for the *sb*-th band in the *c*-th channel, $eX_{sb,c}$, can be derived as shown in (5.11).

$$\begin{aligned} eX_{sb,c} &= \sum_{k=w_{sb}}^{w_{sb+1}-1} \left(X_c(k) - \tilde{X}_c(k)\right)^2 = \sum_{k=w_{sb}}^{w_{sb+1}-1} \left(X_c(k) - \hat{X}_c^{\rho,\tau}(k) - \tilde{R}_c(k)\right)^2 \\ &= \sum_{k=w_{sb}}^{w_{sb+1}-1} \left(R_c(k) - \tilde{R}_c(k)\right)^2 = eR_{sb,c} \end{aligned} \tag{5.11}$$

, where $eR_{sb,c}$ is the total quantization noise of the residual spectral coefficients for the *sb*-th band on the *c*-th channel. Therefore, if we can control the quality of $eR_{sb,c}$, it is equivalent to control $eX_{sb,c}$. In other words, the perceptual masking thresholds derived for the original spectral coefficients, $X_c(k)$, can be directly used on the residual spectral coefficients, $R_c(k)$.

Consequently, no extra perceptual masking checking is needed.

5.2.5 Coding of Predictor Parameters

In order to reconstruct the coded signals at the decoder, the predictor parameters are coded and transmitted as side information sent to the decoder. The numbers of bit for coding the predictor parameters are shown in Table 5.1. The time delay, τ, is restricted to the range 0 to 1023 and is represented by a 10-bit index. Each predictor coefficient, β^{TS} or β^{SC}, is quantized by a 4-bit non-uniform quantizer. In addition, a 6-bit switch flag is needed to indicate the numbers of band that the PW-ICP is in use. For example, if switch flag equal to 40, the PW-ICP is in use for the first 40 bands. If the adaptive predictor order scheme is in use, an extra 2~3-bits index is needed. The value of ρ represents the reference channel used in prediction. When ρ equals to zero, PW-ICP is not in use and thus the other predictor parameters are not transmitted. Note that (m+1) is the predictor taps.

Table 5.1: Side information bits for inter-channel predictor

Parameter	Bits/Channel
reference channel (ρ)	1~2
switch flag	6
time delay (τ)	10
coefficient (β^{TS}, β^{SC})	4 × (m+1)
Order	2~3

5.3 Simulation Results

To analyze the characteristics of the proposed inter-channel predictors and to evaluate the performance of our inter-channel redundancy removal algorithms, we implement and compare several algorithms using the MPEG AAC as the test platform. A number of two-channel audio sequences with sampling rate 44.1kHz and five-channel surround audio sequences with sampling rate 48kHz are used in simulations. The two-channel audio sequences are extracted from MPEG SQAM [6], EBU [24] and downloaded from a few audio quality testing websites. The five-channel surround audio sequences are extracted from the 5.1-channel DTS audio streams of several DVDs.

5.3.1 Comparison on Methods of Determining τ and ρ

1) Bit Rate Reduction Performance:

In Section 5.2.2, we propose a new error index, called *minimum perceptually weighted prediction error* (PWPE_{min}), for determining τ and ρ. In this section, we conduct a few simulations on the PWPE_{min} method and the traditional correlation method, Corr_{max}, defined in (5.5). The 1-Tap SCP described in Section 5.2.1 is adopted. The bit rate reduction performance of 15 two-channel audio sequences is shown in Table 5.2. The notation "Fixed τ" in Table 5.2 means that the time delay, τ, of the predictor is fixed to a specific value and in the following experiments, the value is typically zero. We find that the performance (bit rate reduction) of PWPE_{min} method on the average is around 5% better than that of Corr_{max} method. In general, the performance of Corr_{max} method is worse than that of the fixed τ mode (τ=0). When examining the PWPE_{min} method with fixed or adaptive τ modes, we can find that, a good estimate of τ can provide significant improvement on certain audio sequences, such as the last three audio sequences (shaded) in Table 5.2. Another interesting phenomenon is that, for a lot of audio sequences, the fixed τ mode (τ=0) produces a performance similar to the adaptive τ mode (PWPE_{min}). But for the last 3 sequences, the bit reduction of using PWPE_{min} is significantly higher than the other two schemes. The computation complexity of the fixed τ mode is much lower than that of the adaptive τ mode. Therefore, there is a trade-off between complexity and performance.

Table 5.2: Bit rate reduction (%) of Fixed τ(τ=0), Corr$_{max}$ and PWPE$_{min}$

Audio Sequence	Fixed τ(τ= 0)	Adaptive τ(Corr$_{max}$)	Adaptive τ(PWPE$_{min}$)
1. vocal	1.8	0.69	4.42
2. speech	18.15	11.11	18.22
3. pop_1	6.89	3.34	7.5
4. pop_2	6.39	0.72	6.44
5. guitar	7.52	4.23	7.53
6. tune	10.98	8.46	11.58
7. pop_3	6.38	2.68	6.39
8. rock_1	6.09	8.35	9.06
9. pop_4	11.60	9.29	11.95
10. castanet	6.7	6.5	7.62
11. pop_5	23.6	3.89	23.6
12. pop_6	16.54	11.97	16.53
13. pop_7	9.08	8.54	15.98
14. rock_2	2.55	11.19	15.26
15. pop_8	12.31	10.34	20.92

2) Fast Algorithm for PWPE$_{\mathrm{min}}$ Analysis:

As described in Section 5.2.2, the proposed PWPE$_{min}$ method needs to test all possible delay parameters on the spectral coefficients, $\hat{X}_c^{\rho',\tau'}(k)$. For each candidate pair (ρ',τ'), a transform operation is thus required and the number of all candidates of τ' is 1024 in our PW-ICP algorithm. It results in a huge number of calculations. In this section, a fast algorithm for the PWPE$_{min}$ analysis in AAC is proposed by reducing the number of transform operations needed. The transform operation in AAC is the Modified Discrete Cosine Transform (MDCT) as defined in (5.12). In addition, for calculation purpose, we introduce another transform, called Modified Discrete Sine Transform (MDST), as in (5.13).

$$X[k] = 2 \cdot \sum_{i=0}^{2N-1} x[i] \cdot \cos(n_0 \cdot (k+0.5) \cdot (i+n_1)), \quad k = 0,\ldots,N-1 \tag{5.12}$$

$$\overline{X}[k] = 2\cdot\sum_{i=0}^{2N-1} x[i]\cdot\sin(n_0\cdot(k+0.5)\cdot(i+n_1)),\quad k=0,\ldots,N-1 \tag{5.13}$$

where N is the length of a frame, $x[i]$ is the time sample, $n_0 = (2\pi/2N)$ and $n_1 = (N+1)/2$.

The MDCT and MDST coefficients with delay τ' are denoted as $X^{\tau'}[k]$ and $\overline{X}^{\tau'}[k]$ and are defined by (5.14) and (5.15).

$$X^{\tau'}[k] = 2\cdot\sum_{i=0}^{2N-1} x[i-\tau']\cdot\cos(n_0\cdot(k+0.5)\cdot(i+n_1)) \tag{5.14}$$

$$\overline{X}^{\tau'}[k] = 2\cdot\sum_{i=0}^{2N-1} x[i-\tau']\cdot\sin(n_0\cdot(k+0.5)\cdot(i+n_1)) \tag{5.15}$$

Then, the MDCT coefficient with delay (τ'+1) is obtained from mostly combinations of $X^{\tau'}[k]$ and $\overline{X}^{\tau'}[k]$ as shown in (5.16).

$$\begin{aligned}
X^{\tau'+1}[k] &= 2\cdot\sum_{i=0}^{2N-1} x[i-(\tau'+1)]\cdot\cos(n_0\cdot(k+0.5)\cdot(i+n_1)) \\
&= 2\cdot x[-\tau'-1]\cdot\cos(n_0\cdot(k+0.5)\cdot n_1) + 2\cdot\sum_{i'=0}^{2N-2} x[i'-\tau']\cdot\cos(n_0\cdot(k+0.5)\cdot(i'+n_1+1)) \\
&= A_1[k] + \cos(n_0\cdot(k+0.5))\times 2\cdot\sum_{i'=0}^{2N-2} x[i'-\tau']\cdot\cos(n_0\cdot(k+0.5)\cdot(i'+n_1)) \\
&\qquad -\sin(n_0\cdot(k+0.5))\times 2\cdot\sum_{i'=0}^{2N-2} x[i'-\tau']\cdot\sin(n_0\cdot(k+0.5)\cdot(i'+n_1)) \\
&= \underline{A_1[k] + \cos(n_0\cdot(k+0.5))\times\left(X^{\tau'}[k]-B_1[k]\right) - \sin(n_0\cdot(k+0.5))\times\left(\overline{X}^{\tau'}[k]-B_2[k]\right)}
\end{aligned} \tag{5.16}$$

, where $i' = i-1$,

$A_1[k] = 2\cdot x[-\tau'-1]\cdot\cos(n_0\cdot(k+0.5)\cdot n_1)$,

$B_1[k] = -2\cdot x[2N-1-\tau']\cdot\cos(n_0\cdot(k+0.5)\cdot(2N-1+n_1))$,

$B_2[k] = -2\cdot x[2N-1-\tau']\cdot\sin(n_0\cdot(k+0.5)\cdot(2N-1+n_1))$.

Similarly, the MDST coefficient with delay (τ'+1) can be obtained from mostly combinations of $X^{\tau'}[k]$ and $\overline{X}^{\tau'}[k]$ as shown in (5.17).

$$\begin{aligned}\overline{X}^{\tau'+1}[k] &= 2\cdot\sum_{i=0}^{2N-1} x[i-(\tau'+1)]\cdot\sin(n_0\cdot(k+0.5)\cdot(i+n_1)) \\ &= 2\cdot x[-\tau'-1]\cdot\sin(n_0\cdot(k+0.5)\cdot n_1)+2\cdot\sum_{i'=0}^{2N-2} x[i'-\tau']\cdot\sin(n_0\cdot(k+0.5)\cdot(i'+n_1+1)) \\ &= \underline{A_2[k]+\sin(n_0\cdot(k+0.5))\times\left(X^{\tau'}[k]-B_1[k]\right)+\cos(n_0\cdot(k+0.5))\times\left(\overline{X}^{\tau'}[k]-B_2[k]\right)}\end{aligned}\tag{5.17}$$

, where $A_2[k]=2\cdot x[-\tau'-1]\cdot\sin(n_0\cdot(k+0.5)\cdot n_1)$.

As shown from (5.14) to (5.17), we only need to perform one full MDCT transform (in (5.14)) and one full MDST transform (in (5.15)) at the beginning of PWPE$_{min}$ analysis for a given ρ'. Then, $X^{\tau'+1}[k]$ and $\overline{X}^{\tau'+1}[k]$ can be simply derived from $X^{\tau'}[k]$ and $\overline{X}^{\tau'}[k]$ with a few additional multiplication and addition. Therefore, by using this fast algorithm, the complexity of PWPE$_{min}$ analysis can be drastically reduced.

5.3.2 Analysis on Predictor Order of Inter-Channel Predictor

In Section 5.2.3, we have discussed some characteristics of predictor order for the TSP and SCP. In this section, more simulations are conducted. The average bit rate reduction difference between the tested predictor (testee) and the 1-Tap TSP (reference) of 45 two-channel and 20 five-channel audio sequences are shown in Fig. 5.10. Note that, the side information bits for coding the predictor coefficients, β^{TS} or β^{SC}, are included. The first data point on the horizontal axis in Fig. 5.10 is marked as the "Adaptive Case", in which the predictor order can vary from one frame to the other. This may be most sophisticated case we can imagine.

We can find from Fig. 5.10 that, the Adaptive-Tap predictor has the best performance (bit rate reduction) for both TSP and SCP schemes. Taking two-channel audio sequences for example, the performance of the Adaptive-Tap SCP is around 4.5% better than that of 1-Tap TSP. For designing a Fixed-Tap predictor, we can find that, a low-order (around 1-Tap) TSP works better for the subband audio coders. For the SCP, the order that is less than or equal to 5 is adequate. The performance downfall of the SCP at high orders (≥ 10-Tap) is mostly due to the increase of side information bits.

Overall, the performance of SCP is better than that of TSP. However, the complexity of SCP is also higher than that of TSP. As shown in Fig. 5.3 and Fig. 5.4, an (*m*+1)-Tap SCP need (*m*+1)-times MDCT operations, but an (*m*+1)-Tap TSP needs only one MDCT operation. Moreover, the complexity of an Adaptive-Tap predictor is higher than that of a Fixed-Tap predictor. So, when we considering the predictor order factor, there is a trade-off between complexity and performance too.

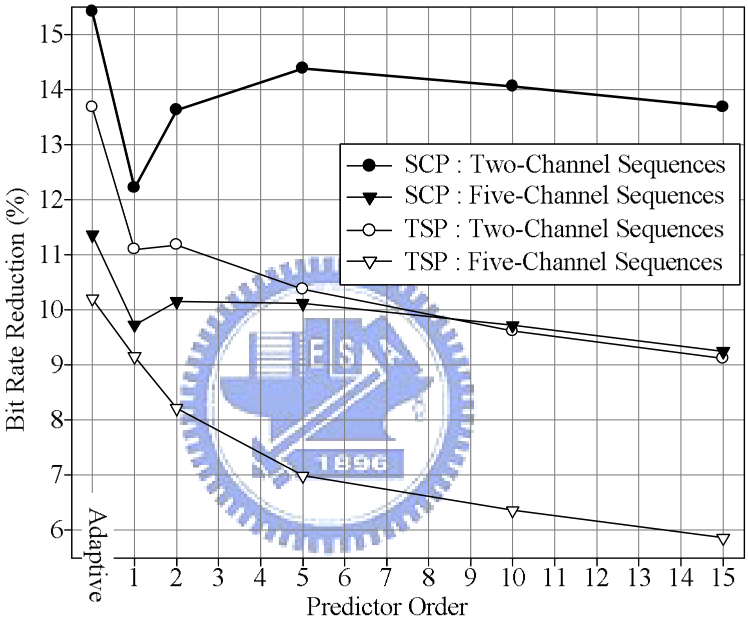


Fig. 5.10: Average bit rate reduction vs. predictor order

5.3.3 Analysis on Various Inter-Channel Redundancy Removal Algorithm

Finally, we come to compare the performance of the overall inter-channel algorithms. Two types of inter-channel redundancy removal algorithms are tested and compared as described below.

(1) The perceptually weighted inter-channel prediction (PW-ICP) algorithm described in Section 5.2.

(2) The INT-DCT based approach described in [15].

Fifteen two-channel audio sequences and fifteen five-channel sequences are tested. The bit rate reduction performance of our PW-ICP algorithm and the INT-DCT based algorithm are shown in Table 5.3 and Table 5.4. We find that the performance (bit rate reduction) of our PW-ICP algorithm on the average is around 10% better than that of INT-DCT based algorithm for the audio sequences that show 5% or more bit rate reduction than the separate-channel coding. In the other cases (lower than 5% bit rate reduction), our algorithm is as good as (if not better than) the INT-DCT based approach. Furthermore, the performance of SCP on the average is around 2% better than that of TSP.

Table 5.3: Bit rate reduction (%) of PW-ICP and INT-DCT for two-channel audio sequences.

Audio Sequence	INT-DCT	PW-ICP (1-Tap TSP)	PW-ICP (5-Tap SCP)
1.vocal	0.83	3.35	3.7
2.orchestra	1.31	2.86	2.57
3.rock_1	1.54	12.42	14.81
4.castanet	3.0	7.17	11.67
5.pop_1	3.35	8.59	8.72
6.rock_2	3.80	4.78	20.42
7.pop_2	5.44	15.08	16.54
8.pop_3	5.67	6.79	13.33
9.pop_4	7.78	14.37	14.9
10.pop_5	8.99	15.42	18.32
11.Speech	9.30	16.42	26.16
12.rock_3	11.85	17.82	20.93
13.guitar	15.37	26.99	28.23
14.pop_6	15.93	17.34	23.04
15.pop_7	25.48	42.97	39.8

Table 5.4: Bit rate reduction (%) of PW-ICP and INT-DCT for five-channel audio sequences.

Audio Sequence	INT-DCT	PW-ICP (1-Tap TSP)	PW-ICP (5-Tap SCP)
1.pop_1	0.29	4.10	6.59
2.Jazz	0.45	2.75	2.92
3.sym.	0.44	4.17	4.88
4.sym.	1.06	5.47	5.94
5.movie_1	1.47	4.87	7.93
6.movie_2	1.71	6.11	8.59
7.V&O*_1	1.8	7.84	7.75
8.orchestra	1.8	6.02	6.60
9.V&O_2	2.2	9.40	10.25
10.pop_2	3.35	11.77	13.20
11.game	4.54	14.28	14.69
12.V&O_3	6.05	15.18	16.15
13.movie_3	7.98	19.07	18.81
14.speech	10.71	31.45	29.93
15.pop_3	12.24	32.36	32.87
V&O* : Vocal & Orchestra			

Chapter 6 Conclusions

The main contributions of this dissertation are summarized as follows:

● Cascaded Trellis-Based Rate-Distortion Control Algorithm (CTB)

The cascaded trellis-based (CTB) optimization scheme is a low complexity and high performance R-D control algorithm for the MPEG-4 AAC coder. It is basically a fast algorithm of the previous joint trellis-based (JTB) scheme. The optimization procedure for finding coding parameters, SF and HCB, in the CTB scheme is partitioned into two sequential steps with carefully inserted steps. It thus has the advantage of a much reduced computation. The proposed CTB scheme is approximately 71 to 142 times faster than the JTB scheme. Simulation results show that both the objective and subjective quality of the proposed CTB scheme is close to that of the JTB scheme.

In addition, we also propose a lossless fast search algorithm for the trellis-based optimization on HCB, which provides roughly a 4-times speed-up. Furthermore, two non-uniform search algorithms for trellis-based MNMR optimization on SF, so-called GMNU and LMNU, are proposed for reducing the candidates in trellis search. Simulation results indicate that another factor of 25 speed-up can be achieved using GMNU with negligible audio quality loss. These two fast search algorithms can be applied to both the CTB scheme and the JTB scheme.

● Enhanced BFOS Bit Allocation Algorithm for AAC (EBFOS)

EBFOS is an efficient bit allocation algorithm for MPEG-4 AAC. Instead of performing the heavy trellis search through entire frame, the bits are allocated to the most needed band step by step in the EBFOS scheme. It thus has the advantages of low complexity and higher flexibility. The performance of the EBFOS scheme is better than that of VM-TLS and the generalized BFOS algorithms. Moreover, the EBFOS scheme has a performance close to the trellis-search based algorithm (optimized for the average NMR, JTB-ANMR). For reducing calculations, a fast algorithm is also introduced for the EBFOS scheme. The fast version can reduce the complexity to 1/10. Simulation result shows that there is almost no loss of performance (less than 0.06dB) in adopting the fast algorithm for the EBFOS scheme.

● Perceptually Weighted Inter-Channel Prediction (PW-ICP)

PW-ICP is an efficient inter-channel redundancy removal algorithm. Different from the M/S stereo coding or the KLT-based approach, the PW-ICP scheme does not propagate the quantization noise from one channel to other channel. Therefore, no extra perceptual masking control is needed. Moreover, similar to the INT-DCT based approach, no audio quality degradation is induced by our method. In our PW-ICP algorithm, two types of predictors, TSP and SCP, are introduced. Also presented in this dissertation are simulations and detailed discussions on how to determine the parameters of the predictors. We find that the performance of our new index, PWPE$_{min}$, is better than that of traditional correlation method, Corr$_{max}$. For a chosen predictor order, the predictor with adaptive order can achieve the best performance for all kinds of audio signals. (Larger order predictors are often not preferred.) As for the predictor with fixed order, in general, the order around 1 is appropriate for TSP and the order less than or equal to 5 works best for SCP.

To evaluate the performance of our PW-ICP algorithms, the INT-DCT based approach is also implemented and compared. We have tested this scheme on a number of two-channel and five-channel audio sequences. Based on the simulation results, we find that the bit rate reduction performance of our new method on the average is about 10% better than that of the well-known INT-DCT based approach for the audio sequences that show 5% or more bit rate reduction than the separate-channel coding.

Bibliography

[1] T. Painter and A. Spanias, “Perceptual coding of digital audio,” *Proc. of IEEE*, vol. 88, pp. 451- 515, Apr. 2000.

[2] ISO/IEC JTC1/SC29, “Information technology – Coding of audio-visual objects,” *ISO/IEC IS-14496 (Part 3, Audio)*, 1999.

[3] J. Herre, B. Grill, “Overview of MPEG-4 audio and its applications in mobile communications,” *Proc. of WCCC-ICSP*, vol. 1, pp. 11-20, Aug. 2000.

[4] H. Purnhagen, "An Overview of MPEG-4 Audio Version 2," *AES 17th International Conference on High-Quality Audio Coding*, Firenze, Sept. 1999.

[5] M. Bosi, *et al.*, “ISO/IEC MPEG-2 advanced audio coding,” Journal *of Audio Engineering Society*, vol. 45, pp. 789-812, Oct. 1997.

[6] “The MPEG audio web page,” [Online]. Available: http://www.tnt.uni-hannover.de/project/mpeg/audio/

[7] A. Aggarwal, *et al.*, “Trellis-based optimization of MPEG-4 advanced audio coding,” *Proc. IEEE Workshop on Speech Coding*, pp. 142-144, Sept. 2000.

[8] A. Aggarwal, *et al.*, “Near-optimal selection of encoding parameters for audio coding,” *Proc. of ICASSP*, vol. 5, pp. 3269-3272, May 2001.

[9] P. H. Westerink, *et al.*, “An optimal bit allocation algorithm for sub-band coding,” *Proc. of ICASSP*, pp. 757-760, 1988.

[10] E. A. Riskin, “Optimal bit allocation via the generalized BFOS algorithm,” *IEEE Trans. On Information Theory*, vol. 37, No. 2, Mar. 1991.

[11] G. Diego and R. Sudhakar: “Optimal Bit Allocation for MPEG Audio Standard Using the Generalized BFOS Algorithm", *98th AES Convention*, Paris, Feb. 1995

[12] G.D. Forney. “The Viterbi Algorithm,” *Proc. IEEE*, vol.1 61, pp. 268-278, Mar. 1973.

[13] K. Sayood, *Introduction to Data Compression*, 2nd ed., Morgan Kaufmann Publishers, San Francisco, 2000.

[14] D. Yang, *et al.*, “An inter-channel redundancy removal approach for high-quality multichannel audio Compression”, AES 109th Convention, Sept. 2000, Los Angeles, USA.

[15] Y. Wang, *et al.*, “A multichannel audio coding algorithm for inter-channel redundancy removal”, AES 110th Convention, May 2001, Amsterdam, Netherlands.

[16] ISO/IEC 11172-3 "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s, Part 3: Audio," 1992.

[17] ISO/IEC 13818-3, "Information Technology – Generic Coding of Moving Pictures and Associated Audio, Part 3: Audio," 1994-1997.

[18] J. H. Rothweiler, "Polyphase Quadrature Filters - A new Subband Coding Technique," International Conference IEEE ASSP 1983, Boston, pp. 1280-1283.

[19] J. Princen, A. Johnson, and A. Bradley, "Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation," Proc. of the ICASSP 1987, pp. 2161-2164.

[20] ISO/IEC JTC1/SC29/WG11 MPEG, International Standard ISO/IEC 13818-7 "Generic Coding of Moving Pictures and Associated Audio: Advanced Audio Coding", 1997

[21] ISO/IEC JTC1/SC29/WG11 (MPEG), International Standard 14496-3 Amd 1: "Coding of Audio-Visual Objects: Audio", 2000

[22] S. Golomb, “Run-length encodings,” *IEEE Transactions on Information Theory*, vol. 12, pp. 399-401, July 1966.

[23] H. Najafzadeh and P. Kabal, “Perceptual bit allocation for low rate coding of narrowband audio,” *Proc. of ICASSP*, vol. 2, pp. 893-896, June 2000.

[24] European Broadcasting Union, *Sound Quality Assessment Material: Recordings for Subjective Tests* Brussels, Belgium, Apr. 1988.

[25] ITU-R BS.1116, “Method for subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems,” 1994.

[26] Draft ITU-T Recommendation BS.1387: “Method for objective measurements of perceived audio quality,” July 2001.

[27] A. Lerchs, “EAQUAL software”, Version 0.1.3 alpha, [Online]. Available: http://mitiok.free.fr/c.htm

[28] K. Brandenburg and G. Stoll, “ISO-MPEG-1 audio: a generic standard for coding of high-quality digital audio,” *J. Audio Eng. Soc.*, vol. 42, pp.780-792, Oct. 1994.

[29] L. Breiman, J. H. Freidman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees. The Wadsworth Statistics/Probility Series*. Belmont, California: Wadsworth, 1984.

[30] T. Liebchen, “Lossless audio coding using adaptive multichannel prediction”, AES 113th Convention, Oct. 2002, Los Angeles, USA.

[31] J. D. Johnston and S. S. Kuo, “A study of why cross channel prediction is not applicable to perceptual audio coding”, IEEE Signal Processing Letters, Vol. 8, Issue. 9, pp. 245 - 247, Sept. 2001.