博士論文

Coding model extension by non-linear signal shaping for low-delay speech-and-audio coding (低遅延音声音響符号化のための非線形信号整形法による符号化モデル拡張についての研究)

杉浦 亮介

和文概要 (邦題：「音声音響符号化のための非線形信号整形法による符号化モデル拡張についての研究」)

近年, ネットワークや通信機器の発展に支えられ, 我々は日々大量の情報を生み出し, 発信・受信することが可能となった．中でも音声通信の発展は目覚ましく, 今では場所や時間によらず音声音響信号のやりとりが可能である．このように音声通信が当たり前となった時代においては, より品質の高い通信に対する需要が益々増加していくことが予想される．この品質は, 扱える情報量や音質は勿論のこと, 通信のリアルタイム性やコストなども含めて包括的に考慮する必要がある．特にリアルタイム性は, 刻一刻生み出される音声音響信号の情報を最大限に活用するには必要不可欠な要素である．

長年この音声通信を下支えしてきたのが音声音響符号化の技術であり, この技術はこれらの需要に応えるにあたって重要な役割を果たす．音声音響符号化の枠組みにおいて, 音声通信の品質に大きく影響を与える情報量や, 復号音質, 原理遅延, 計算量, 誤り耐性の各要素はお互いに複雑に依存し合い, 様々なトレードオフが存在する．通信の用途によって変わる制約の中で, 高い音質を保ちながらリアルタイム性を重視した低遅延な符号化を実現することは困難な課題として残っており, 本研究はこの課題に焦点を当てる．

本論文では, 既存の多くの符号化方式が, 制約の違い毎に実装は大きく異なるがその本質は変わらないことに着目し, 低遅延条件下の符号化方式において幅広く活用できる基盤となる技術を提案する．その中で重視するのが, 計算量や既存の技術との親和性である．低遅延な条件下では計算量が増加しやすく, また符号化は様々な技術の複合体となるため, 簡潔で低演算量な処理が望ましい．そこで, 提案手法は全て信号や情報の整形, 情報を失わずに見方だけを変える単純な可逆な変換により, 既存の重要なモデルの拡張を実現している．

章の構成としては, まず第 2 章で, 音声音響符号化が大きく信号・感度・符号の三種類のモデル化によって構成されていると解釈し, 多くの方式が陰に陽に用いている共通の仮定をこの考え方に則って説明する．そして第 3, 4, 5 章でこれらモデル化における基盤となるモデルを拡張するための整形法の提案をし, 第 6, 7, 8 章で提案手法の応用例として実際の符号化方式を実装し評価を行う．

第 3 章は信号モデル, 音声音響信号の統計的な仮定を決めるモデルに着目している．ここでは信号の周波数スペクトルの包絡表現に重きを置き, 主要技術である線形予測を拡張するための, サンプル単位で効果を発揮する整形法を独立した観点から三つ提案する．一つ目はスペクトル包絡推定の周波数の解像度を伸縮するための非負値疎行列による整形である．二つ目はス

ペクトルの分布の仮定を, より広いクラスである一般化 Gauss 分布としたときでも最適にスペクトル包絡を推定するための累乗演算による整形である. 三つ目はスペクトル包絡による白色化処理を整数値の領域においても可逆に行えるようにするための, ビット平面上の置換による整形である.

第 4 章は感度モデル, 復号音声の聞き手の感度を表すモデルに着目している. ここでは聴覚心理に従った量子化雑音の聴感的な制御を信号の重み付け処理として捉え, 他のモデルとの親和性の高いものを実現するため, 信号のブロック単位で効果を発揮する整形法として, ブロック圧伸の拡張による整形を提案する.

第 5 章は符号モデル, 信号・感度モデルが与える目的関数や仮定に基づいて歪みと情報量の最適化を図るアルゴリズムをモデルに着目している. ここでは, 低演算量で Laplace 分布に属する信号の圧縮に最適なことから広く用いられている可変長符号である Golomb–Rice 符号に着目し, Golomb–Rice 符号の最適性の条件を拡張するため, 分布単位で効果を発揮する整形法を独立した観点から二つ提案する. 一つ目は Golomb–Rice 符号が最適性を失うような疎な Laplace 分布において最適な符号を構成できるようにするための, ビット反転による整形である. 二つ目は一般化 Gauss 分布に従う入力に対してでも Golomb–Rice 符号で最適に圧縮できるようにするための, 整数マッピングによる整形である.

第 6 章では, 移動通信のような低遅延且つ低ビットレートな符号化方式の応用例を示す. 音声信号が得意な時間領域の符号化方式と組み合わせて用いることを想定した, 最新の周波数領域の符号化方式において, 信号モデルにおける提案法が信号の仮定をより忠実なものとし, 圧縮の効率を向上させることを実証する. また, 符号モデルにおける提案法により, 一般化 Gauss 分布の仮定に拡張しても Golomb–Rice 符号を用いて効率的に圧縮可能であることを確かめる.

第 7 章では, 主に放送局で用いられることを想定したリアルタイム音源伝送用の低遅延で高音質な符号化方式の応用例を示す. 柔軟な実装を可能とする感度モデルにおける提案法により, 別の用途で開発された既存の国際標準方式を再利用する方法や, その性能について議論する.

第 8 章では, 無線マイクに代表されるリアルタイム機器間通信を想定した, 伝送路でビット誤りが起こっても復号結果への影響が小さい低地艶な符号化方式の応用例を示す. 信号モデルと感度モデルにおける提案法を駆使し, ビット誤りに対して頑健な符号化方式を提案し, その頑健性や音質について検証する.

これらの応用例は実装は異なるが, 基本的な構成・仮定は共通している. 提案する整形法も全て互いに組み合わせて使用することが可能であり, 既に一部は商用として採用されていることから, 低遅延音声音響符号化方式の設計において欠かせない技術となると期待できる.

キーワード　音声音響符号化, 低遅延, 線形予測分析, Golomb–Rice 符号, AIFV 符号, 周波数伸縮, 非負値疎行列, ブロック圧伸, 一般化 Gauss 分布

Abstract

In these days, owing to the development of networks and devices, we can every day produce, send, and receive many kinds of information. Among that, speech-and-audio communication has significantly been developed, which enables us to communicate with each other anytime, anywhere. Now, with the speech-and-audio communication taken for granted, it is expected that the demand for higher quality will further increase. The quality of communication, of course, includes the sound quality and information amount to be dealt with, but also its realtimeness and costs should be taken into account. Notably, the realtimeness is one of the essential factors for entirely making use of the speech and audio produced moment by moment.

Speech-and-audio coding techniques have been responsible for communication for years and are the key to meeting the rising demands. In the framework of coding, the factors related to the quality of the communication such as information amount, reproduction sound quality, algorithmic delay, computational complexity, and error robustness depend on each other in very complicated ways so that there are many trade-offs related to them. There remain challenges to realize low-delay coding schemes maintaining high sound quality under various conditions dependent on use cases, which is the primary goal of this work.

This thesis focuses on the fact that there are standard fundamentals of modeling speech-and-audio coding even though their implementations differ following the use cases. Basic techniques are proposed to extend the coding models, making them more useful in low-delay coding. Here, we place a high value on computational complexity and affinity to other technologies. Since computation costs tend to be significant in low-delay conditions, and the techniques have to harmonize to perform efficient coding, the methods are preferable to be simple and low complexity. Therefore, all the methods proposed here are realized by shaping, simple invertible transformations that change only the measurement of signals without losing any information.

This thesis begins in Chap. 2 with an interpretation of modeling speech-and-audio coding roughly as three types of modelings, signal, perception, and code modelings. It explains the common assumptions used in many coding schemes implicitly or explicitly.

Then in Chaps. 3, 4, and 5, the shaping methods are presented as basic techniques, which are implemented and evaluated in some codec applications in Chaps. 6, 7, and 8.

Chap. 3 stands on the signal modeling, the modeling determining the statistical assumptions for signals. Here, we focus on envelope representation of frequency spectra and present three shaping methods based on samples, from independent points of view, to extend the linear prediction, the principal model for the signal modeling. The first one is the shaping by non-negative sparse matrices for warping frequency resolution of envelope estimation. The second one is the shaping by powering operation for extending the assumption of frequency spectra in envelope estimation to generalized Gaussian distributions, a broad class of exponential family. The third one is the shaping by bit-plane rearrangement for realizing invertible whitening by spectral envelopes for integer-domain frequency spectra.

Chap. 4 stands on perception modeling, the modeling defining the sensitivity of the listeners. Here, we understand the psycho-acoustic control of quantization noise as weighting processing and present a shaping method based on blocks of signal, consisting of an extension of block companding, to realize a high-affinity model.

Chap. 5 stands on the code modeling, the modeling giving us algorithms to optimize the rate-distortion trade-offs subjecting to the sensitivity and assumptions derived from signal and perception modeling. Here, we focus on Golomb–Rice code, the widely-used low-complexity variable-length code known to be optimal for Laplacian-distributed signals, and present two independent shaping methods based on distributions to extend its optimality. The first one is the shaping by bit inverting for constructing codes optimal for sparse Laplacian sources for which the conventional Golomb–Rice code loses its optimality. The second one is the shaping by integer mappings for Golomb–Rice code to compress optimally the inputs belonging to generalized Gaussian distributions.

In Chap. 6, we show as an application a low-delay and low-bit-rate codec mainly for mobile communication. We evaluate a state-of-the-art frequency-domain codec implementing the proposed methods for the signal modeling, assuming its use in combination with time-domain codecs, which is known to be efficient for speech inputs. The methods are examined whether they give us more reasonable assumptions for frequency spectra and enhance compression efficiency. Besides, we discuss applying the proposed method for the code modeling to use generalized Gaussian assumptions for real codecs based on

Golomb–Rice coding.

In Chap. 7, we show as another application a low-delay and high-sound-quality codec mainly for real-time sound data transmission in broadcasting. We discuss reusing the international standards, developed for other use cases, by applying the proposed method for perception modeling enabling us more flexible codec design and evaluate its performance.

In Chap. 8, we show as the last application a low-delay codec having a small influence on the reconstruction quality even if bit errors occur in the communication line, aiming at its use in real-time inter-device communication such as wireless microphones. Taking advantage of the proposed methods for signal and perception modeling, we design a bit-error-robust codec and investigate its robustness and sound quality.

These applications differ in the implementation, but their underlying structures and assumptions are the same. Additionally, all the shaping methods proposed here can be combined, and some of them are already adopted for commercial use. Therefore, they are expected to be essential techniques for low-delay speech-and-audio coding.

Keywords Audio compression, Low delay, Linear prediction, Golomb–Rice code, Frequency warping, Non-negative matrix, Block companding, Generalized Gaussian distributions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of speech-and-audio coding

In about half a century, the digitalization of speech-and-audio signals has made rapid changes in communication. Mainly, their compression, or coding, has been greatly contributed to prosperity. Commencing with ITU-T G. 711 [1], used for fixed-line telephones, the development of low-bit-rate codecs such as 3GPP Adaptive multi-rate (AMR) [2, 3], 3GPP AMR wideband (AMR-WB) [4, 5], and 3GPP Enhanced voice services (EVS) [6–8] now enables us, just with small mobile phones, to talk naturally to whoever we want, regardless of wherever we are and whenever it is. High-sound-quality codecs such as MPEG-4 Advanced audio coding (AAC) [9, 10] and MPEG-4 Audio lossless coding (ALS) [11, 12] have enhanced the flexibility of broadcasting and archiving of audio, allowing us to enjoy high-quality audio contents in various media not limited to television.

As described above, speech-and-audio coding has supported the realization of desired sound communication overcoming the limitations of space and devices. However, the recent circumstances of data and network dependence are requiring more and more peculiar and strict conditions for compression. From the perspective of data in general, the information amount of the data generated in a year exceeded the overall storage capacity in 2007 [13], of which the gap is getting larger every year. The sensors collecting those data are also increasing in amount as well as in variety, exceeding ten billion in 2015 [14]. Also, the networks are changing. For instance, the fifth-generation (5G) mobile communication networks are designed to use network slicing, a technique of flexibly costuming the network capability and priority, to meet minute demand on data communications [15].

This fact means we are expected to consume more and more amount of data coming from sensors under various kinds of conditions. Furthermore, speech-and-audio data are no exceptions: We are now communicating with people remotely by phones or maybe locally by amplifying our voice with microphones, which send our speech signals to loudspeakers in real-time; sometimes sound data are transmitted to audio processing systems in broadcasting; we may communicate even with machines owing to speech recognition and synthesis techniques, for example. Speech and audio coding technology have the potential to realize those demands, which usually have their specific conditions, more flexibly with higher quality.

Here, we mainly focus on lossy coding, which compresses input signals usually into some constant bit rate admitting distortion in the decoded ones and is often more useful than lossless when the bit rates have strict upper limits. Lossy compression of speech-and-audio signals has fundamental trade-offs between the following conditions:

- Reconstruction quality, the quality of the decoded signals;
- compression bit rates, the overall description length of the encoded signals;
- algorithmic delay, the unavoidable delay due to the coding processes even if the computation and delivery are done immediately;
- computational complexity, the computational costs affecting the required computational resources;
- error robustness, the effect on reconstruction quality of errors occurred in encoded bitstreams.

These conditions are complicatedly related to each other, and their required levels heavily depend on the use cases, which makes the codecs difficult to design. Also, the reconstruction quality depends on how we use the decoded signals. For example, if the signals are analyzed or processed after decoded, the fidelity of the waveforms such as signal-to-noise ratio (SNR) may be a critical criterion, while thinking of speech communications, the criterion should be a perceptual difference, which cannot always be estimated from SNR.

Let us briefly look at some conditions in the conventional international standards of lossy codecs described in Table 1.1: ITU-T G. 711. G. 726 [16], G. 729 [17], and 3GPP AMR-WB are speech codecs working with low delay; codecs such as 3GPP Extended adaptive multi-rate wideband (AMR-WB+) [18] and MPEG-D Unified speech and audio coding

(USAC) [19–21] aims also at music but have a longer delay; in music codecs, MPEG-1/2 Audio layer 3 (MP3) [22, 23] is very famous for audio archiving, and MPEG-4 AAC Enhanced low delay (AAC-ELD) [24] compresses with low delay but still requires higher bit rate compared to speech codecs. The assumed input type and the sampling rate are related to the requirements for the reconstruction quality. Typically, music signals contain more information than speech ones in the same sampling rate. The packet dependence shows that whether or not the codecs are designed for use in packet-based communication such as Internet protocol (IP). The conditions of the use cases correspond to the error robustness, whether the codecs are bit-error robust or packet-error robust, and in general, the former is the stricter. As we can see from the table, the conditions have many trade-offs, and the standardized codecs have covered many combinations of requirements by developing several techniques for each case.

According to the trends mentioned previously about data and networks, more demands are expected in the future on continuously consuming speech-and-audio data: Demands on mobile communications with higher sound quality and presence, demands on making use of captured sounds by such as sound detection or speech recognition, demands on live coverage. In that sense, the timeliness of speech-and-audio communications becomes more important, and for their coding, “low delay” will be one of the critical factors.

It is known that, in simple conversation, the overall transmission delay becomes detectable from 100 ms [25], and thus the algorithmic delay of a codec should be at most around 30 ms because there are other factors such as networks. In more severe use cases such as remote ensemble and local speech amplifying by microphones, for example, the limitations of the algorithmic delay will be much shorter as a few milliseconds. Under such conditions, it becomes more difficult to lower the bit rates or enhance the reproduction quality compared to the codecs used in archiving. Moreover, due to the shorter frame of coding processes, the computational complexity tends to be higher when adding some analysis. Therefore, low-delay coding of speech-and-audio signals remains a very challenging issue, requiring efficient modeling of speech and audio in both senses of compression efficiency and computational complexity.

1.2 Motivation of study

The goal of this study is to provide high-efficiency tools, mainly focusing on low-delay lossy speech-and-audio coding, making the codec design more flexible. As shown in Table 1.1, many standardized codecs work under various conditions. However, from a practical point of view, there are still many demands remaining for speech-and-audio coding: Low-delay and low-bit-rate codecs for mobile communications supporting music as well as speech; low-delay codecs for live broadcasting with very high reconstruction quality close to lossless; low-delay and high-quality speech and music codecs that can be used in packet-independent communication lines such as local radio transmission.

Of course, there may be some entirely new approach for each condition. However, deep understandings in the fundamental ideas often give us natural extensions of the conventional methods that provide us with simple but efficient solutions easily combined with other techniques. This study, based on theoretical investigations, proposes “shaping” techniques, techniques in some sense transforming signals without losing information to extend and enhance the conventional basic models.

Some proposed methods in this thesis are already adopted in EVS, the state-of-the-art

Table 1.1. Examples of the conditions of the standardized lossy speech/audio codecs

Codec name	Assumed input type(s)	Sampling rate [kHz]	Bit rate per channel [kbps]	Algorithmic delay [ms]	Packet dependence
G. 711	Speech/Faximile	8	64	< 1	Independent
G. 726	Speech	8	32	< 1	Independent
G. 729	Speech	8	8	15	Independent
AMR-WB	Speech	16	16	25	Independent
AMR-WB+	Speech/Music	16	16	> 100	Dependent
USAC	Speech/Music	48	24	> 100	Dependent
MP3	Music	48	96	> 100	Dependent
AAC-ELD	Music	48	64	15	Dependent

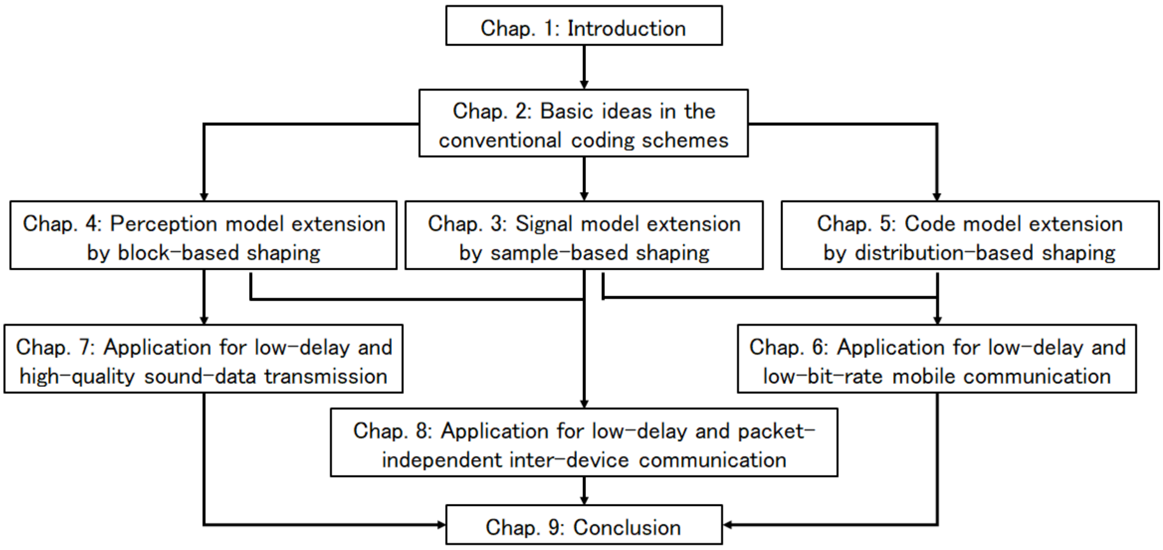


Fig. 1.1. Organization of the chapters.

standard of a speech-and-audio codec, and in codecs for commercial use. The methods may seem disconnected, but they are fundamentally capable of cooperating, depending on the required conditions. Of course, some methods are not yet adopted in the actual applications. However, they indeed show high efficiency and make us design codecs more freely, and thus there are still many chances to be used, sometimes combined with some other methods for the arising demands in the future.

1.3 Outline of this thesis

The organization of the chapters and their relationships are described in Fig. 1.1. The thesis first introduces in Chap. 2 the basic ideas used in the conventional coding schemes. It is explained that the speech-and-audio coding can be interpreted in three fundamental modelings on which every coding schemes are, whether implicitly or explicitly, based. The widely-used methods for them are reviewed, and in Chaps. 3, 4, and 5, the shaping methods are proposed for each of them.

After showing the basic shaping methods, the practical applications are discussed. In Chap. 6, the cooperation of the methods in Chaps. 3 and 5 is introduced focusing on mobile communications. Based on sound-data transmission in broadcasting, the method in Chap. 4 is applied to a codec in Chap. 7. Then in Chap. 8, the methods in Chaps. 3

and 4 are combined for the situations in local inter-device communication. Finally, the thesis is concluded in Chap. 9.

Chapter 2

Basic ideas in conventional coding schemes

2.1 Introduction of this chapter

As explained in the previous chapter, there are many codec standards, and they use different coding schemes to achieve the required conditions. However, most of them are designed based on the modeling categorized, of course, there are some overlaps, into the following three types.

The first one is the signal modeling, which represents the inter-sample relationships of the speech-and-audio signals based on the acoustic characteristics. This one is the modeling of the senders of the sounds or sounds themselves. One of the most straightforward characteristics in this context is the time continuity of the signals, which is often combined with the assumptions of periodicity and harmonics and modeled as filters [26, 27], prediction [28], and frequency spectra [29], for example. For multi-channel signals, their space continuity is one of the essential clues to represent the signals, which is also included in this category. From a statistical point of view, these modeling can be interpreted as regarding the signals as stochastic processes and assuming some distributions that generate their samples.

The second one is perception modeling, which represents the sensitivity of the listeners of the sounds. It concerns whether the difference in the signals is audible and is, in other words, the modeling of the receivers. This one is usually modeled for human listeners by

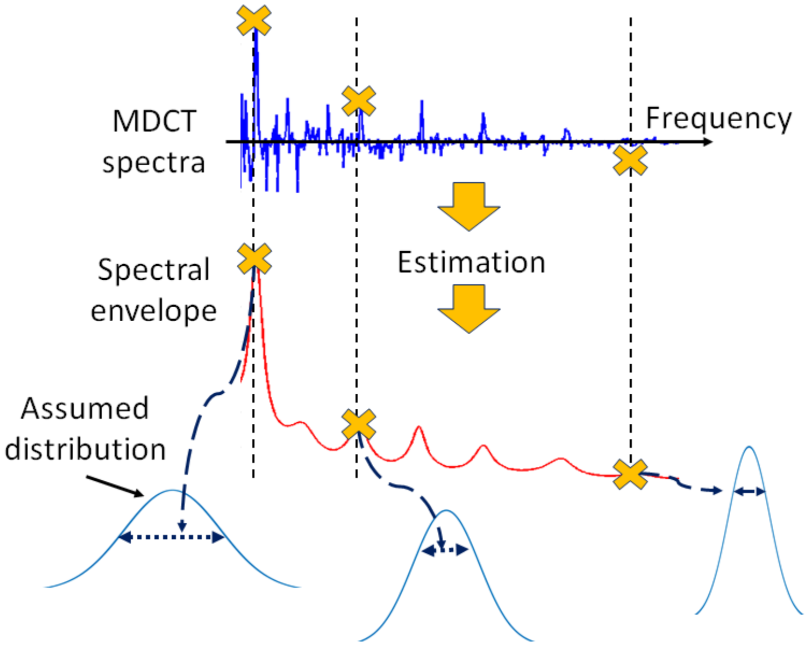


Fig. 2.1. Example of assumptions based on speech modeling.

using the psycho-acoustic effects such as frequency/temporal masking [30]. It is only after modeling the perception that we can numerically evaluate the coding schemes.

The last one is the code modeling, or the modeling of the transmission, which represents the signals by finite numbers or symbols. This one is responsible for finding an optimal trade-off between the bit rates and the reconstruction quality using quantization and data compression techniques based on the rate-distortion theory [31]. The optimization fully depends both on distributions assumed for signals and evaluations in the sense of sensitivity.

To summarize the above, speech-and-audio coding is modeled as an optimization defined by the code modeling with its objective given by the perception modeling and its target distribution by the signal modeling. Based on this interpretation, this chapter shows the major assumptions used in many codecs and widely-used techniques for each modeling, pointing out the main approach of this study.

2.2 Overview of major assumptions

From now on, we mainly focus on monaural signals for simplicity. Many speech-and-audio coding schemes implicitly or explicitly model the correlation of the distributions assumed for the signals, each regarded as a vector of samples in a frame. Furthermore, taking long enough frame length N, they are also assumed to be decorrelated by Fourier transformation. This means the frequency spectra $\{X_k\}_{k=0}^{N-1}$ of a frame, say they are complex numbers given by discrete Fourier transformation (DFT) or maybe real numbers given by modified discrete cosine transformation (MDCT) for example, are assumed independently for each frequency k as

$$X_k \sim p_X(X|\phi_k) \tag{2.1}$$

using some class of distributions $p_X(X|\phi)$ with their scales $\{\phi_k\}_{k=0}^{N-1}$.

For the class of distributions $p_X(X|\phi)$, it is known to be preferable to use ones that are zero-mean, symmetric, and monotonically non-increasing by the magnitude. Therefore, the distributions in the exponential family, especially Gaussian and Laplacian, are well-used in this context. It should be noted that the above assumption underlies not only the frequency-domain coding schemes, compressing strategies for the frequency spectra, such as in [32–34] but the time-domain ones, directly compressing the signal waveforms using some filters, such as in [12, 26, 35, 36]. In the encoding, the scales $\{\phi_k\}_{k=0}^{N-1}$ are directly or indirectly estimated, or parameterized, from the input frame $\{X_k\}_{k=0}^{N-1}$ by the signal modeling to construct a reasonable numerical settings for compression optimization. The compression always strictly concerns the bit rates, critically related to the log-likelihood of the signals by their assumptions. For this reason, the estimation of the model parameters for encoding is often based on a maximum-log-likelihood criterion, equivalent to minimum-description-length criterion when there is no quantization. An example of this assumption is depicted in Fig. 2.1, which shows the assumed distributions of corresponding spectra with their estimated scales. If the values of the spectra are more likely for the distributions, we can achieve lower bit rates by using optimal codes designed for each distribution.

The perception modeling works as some weights in compression optimization. There are various ways of implementation, but its fundamentals are briefly interpreted as follows:

- We are more sensitive to the difference in small-amplitude frequencies;
- we are less sensitive to the difference in large-amplitude frequencies;
- we are less sensitive to the difference in frequencies near the large-amplitude ones (masking effect [30]).

The same characteristics can also be seen in the time domain. Taking into account these characteristics, the reconstruction quality, the difference between the original and decoded signals, are somewhat weighted, for example, by perceptual weights $\{P_k\}_{k=0}^{N-1}$ for each frequency, which usually derived from the input spectra $\{X_k\}_{k=0}^{N-1}$.

Roughly speaking, the code modeling decides the way $Enc[\cdot] : \mathbb{R}^N \to \{0,1\}^{bit}$ and $Dec[\cdot] : \{0,1\}^{bit} \to \mathbb{R}^N$ to minimize the problem based on the above assumptions with a target bit rate B as

$$\sum_{k=0}^{N-1} \int (P_k X_k - P_k \hat{X}_k)^2 p(X_k|\phi_k)\, \mathrm{d}X_k \tag{2.2}$$

subjecting to

$$\{\hat{X}_k\}_{k=0}^{N-1} = Dec[Enc[\{X_k\}_{k=0}^{N-1}]],\ bit \leq B. \tag{2.3}$$

Of course, the distance in the objective function may not limited to L^2-norm, and it is sometimes difficult to distinguish to which modeling it belongs.

2.3 Signal modeling

One of the most basic methods for the signal modeling is linear prediction (LP), capable of representing the short-term correlation of the samples in a frame. Owing to its simple and efficient algorithm for estimating the frame-wise model parameters, it is widely used in speech-and-audio coding schemes [2, 4, 11, 12, 18, 28, 32, 35, 36]. Here, we briefly review how the LP works.

The LP assumes the signals to be predicted by the p-th-order all-pole filter that can be represented in the z-transformation domain as

$$H(z) = \frac{\sigma^2}{1 + \sum_{n=1}^{p} a_n z^{-n}} \tag{2.4}$$

with its model parameters, or LP coefficients $\{a_n\}_{n=1}^{p}$, and prediction gain σ^2, namely the variance of the prediction residuals. The model parameters that minimize the squared errors of the prediction residuals in time domain are estimated from the input signals by solving the Yule-Walker equation:

$$\begin{bmatrix} r_0 & r_1 & \cdots & r_{p-1} \\ r_1 & r_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_1 \\ r_{p-1} & \cdots & r_1 & r_0 \end{bmatrix} \begin{bmatrix} a_1 \\ \\ \vdots \\ \\ a_p \end{bmatrix} = - \begin{bmatrix} r_1 \\ \\ \vdots \\ \\ r_p \end{bmatrix} \tag{2.5}$$

where $\{r_n\}_{n=0}^{p}$ is the auto-correlation function of the input signal. The auto-correlation function $\{r_n\}_{n=0}^{p}$ can be obtained either by correlation operation in the time domain or by inverse cosine transforming the power spectra of the input signal. Eq. (2.5) is the particular case of the Toeplitz-type equations so that it can be solved efficiently by the Levinson–Durbin algorithm while easily checking the stability of the model parameters. Besides, the parameters can be equivalently represented as partial autocorrelation (PARCOR) coefficients and line spectrum pairs (LSPs) [37, 38]. These representations are robust for quantization, which is also the primary reason for adopting LP to the coding schemes.

In the frequency domain, the model of Eq. (2.4) represents the spectral envelope of the input signal. We can easily deduce that the minimization problem of the LP can be approximately written into another form in frequency domain when $p \ll N$ as

$$\min_{\{a_n\}} \sum_{k=0}^{N-1} D_{\mathrm{IS}}(H_k^2 \mid |X_k|^2) \tag{2.6}$$

(see appendix A for the proof) where $\{H_k\}_{k=0}^{N-1} \equiv \{|H(e^{\pi \mathrm{j} k/N})|\}_{k=0}^{N-1}$, and

$$D_{\mathrm{IS}}(x|y) = \frac{y}{x} - \ln \frac{y}{x} - 1 \tag{2.7}$$

is for the Itakura–Saito (IS) divergence of x from y, emphasizing that IS divergence does not satisfy the axiom of distance. j and $\ln(\cdot)$ are for the imaginary number and the natural logarithmic function, respectively. For each frequency k, the terms in the objective function of Eq. (2.6) relating to H_k correspond to the negative log-likelihood of Gaussian with scales H_k:

$$p_G(X_k \mid H_k) = \frac{1}{\sqrt{2\pi} H_k} \exp\left(-\frac{1}{2} \left| \frac{X_k}{H_k} \right|^2 \right). \tag{2.8}$$

This leads to the interpretation of the LP: From the signal-modeling point of view, it assumes Gaussian for the base distribution $p_X(X|\phi)$ explained in the previous section, models its non-uniform scales as $\phi_k = H_k$, and parameterizes the maximum-likelihood scales for all the frequencies, which forms the spectral envelope, with the constraint of the all-pole model in Eq. (2.4).

2.4 Perception modeling

For high bit rates, it is sometimes enough for perception modeling to satisfy only the first two psycho-acoustic characteristics introduced in the previous section. In that case, μ law is used, which is one of the simplest methods adopted, for example, in ITU-T G. 711 [1]. It compands the signals by quantizing logarithmically each sample, which also can be interpreted as a uniform quantization after weighting each sample value by large weights for low values and small weights for high values, respectively.

Block companding [39] is a bit smarter, which uses block-wise representatives to calculate the weights to perform for the corresponding blocks. This process makes use of the values of neighboring samples, approximating the masking effects. By uniform quantizing both the weighted samples in the block and also quantizing its representatives, the block companding perceptually shapes the quantization noise.

The spectral envelopes are as well often used for this kind of modeling, especially when the coding scheme uses LP for the signal modeling. In this method the weights $\{P_k\}_{k=0}^{N-1}$ are approximated from the envelopes as

$$P_k = 1/\tilde{H}_k = \left|1 + \sum_{n=1}^{p} a_n \gamma^n e^{-\mathrm{j}\frac{\pi k}{N} n}\right| \tag{2.9}$$

where $0 < \gamma < 1$. Coefficients $\{\tilde{H}_k\}_{k=0}^{N-1}$ in Eq. (2.9) corresponds to the smoothed form of the envelope $\{H_k\}_{k=0}^{N-1}$. This weighting before uniform quantization shapes the ratio of quantization noise to the original value of the spectra in each frequency into approximately $\tilde{H}_k/H_k$, which is lower in the spectral peaks and higher in the spectral valleys around the peaks.

As the readers may notice from the above explanations, the perception modeling is highly dependent on the other modeling and coding strategies. Therefore, its design often has an impact on or is limited by the whole structure of the codec.

2.5 Code modeling

Here, we are referring to the code modeling as the modeling of compression processes, source coding, in other words, containing quantization and representation in binary codes. The source coding has theoretical limits on the bit rates for given allowed distortion of reconstructed signals, which can be ideally achieved by vector quantization mapping samples at once to fixed-length codes in the optimized codebooks.

Vector quantization shows high compression performance and is thus used in audio coding schemes as in [40]. However, in general, it requires high computational complexity, especially for higher bit rates, and its theoretical optimality holds only for frames with long enough length. Therefore, many codecs use fixed-to-variable-length (FV) coding, or entropy coding, with scaler quantization [7, 8, 10, 18, 20], which compresses the quantized signals based on their entropy with a restriction of decodability. Huffman coding and arithmetic coding are the FV codes capable of optimally compressing stationary non-memory random numbers by using codebooks or algorithm based on the distributions of input signals.

On the other hand, some FV codes focus on specific distributions to realize simpler algorithms. Golomb code [41] and Golomb–Rice (GR) code [42] are the well-known examples. After this, we refer to GR code as Rice code when we want to distinguish from Golomb code. These codes focuses on the geometric distribution, or the half-sided Laplacian distribution,

$$p_{\mathrm{Geo}}[X|\theta] = (1-\theta)\theta^X \tag{2.10}$$

enabling us to compress the absolutes x of the quantized signals with low computational complexity and small memory. For example, Rice code, the particular case of Golomb code, consists of a prefix code and an R-bit suffix code, with a tunable Rice parameter R (≥ 0). Rice encoding only requires division of input symbols by 2^R, or an R-bit-shift operation, and respectively represents the quotient and the remainder by unary and R-bit binary. Golomb code is almost as simple as Rice code, having a finer tunable Golomb parameter S ($= 2^R - s,\ R \geq 0,\ 0 \leq s < 2^{R-1}$) and identical to Rice code when S is a power of 2. Instead of the R-bit suffix code, Golomb code includes the phased-in binary code [43–46], one of the complete binary tree (CBT) codes [47] which switches R-bit and

$(R-1)$-bit binary codes depending on the input values. The example of the codes is described in Table 2.1.

Laplacian distribution is practical for use in the signal assumption $p_X(X|\phi)$ mentioned in the previous section. From this fact and the simplicity of the algorithms, Golomb and Rice codes are widely used for coding applications such as in [48–50].

2.6 Focus of this study

As shown in the previous sections, there are many useful techniques to model speech-and-audio coding, categorized into the three modelings. Although there are still many conditions they do not fully cover, it is preferable, in the sense of affinity to each other, to make more use of the conventional techniques. Therefore, our study aims at extending and enhancing the models through simple processing based on theoretical investigations. We propose in this thesis six “shaping” methods, transforming signals without losing information, to make the conventional methods solve their challenges.

As the readers may notice that the modeling introduced in the previous sections differs in the time-or-frequency scopes they tackle with: The signal modeling takes care of the statistics of each sample; perception modeling mainly concerns only the blocks of neighboring samples; the code modeling depends on the distributions for wider scopes. With the

Table 2.1. Code examples of Golomb and Rice codes

Input	$R=0$	$R=1$	-	$R=2$	-
x	$S=1$	$S=2$	$S=3$	$S=4$	$S=5$
0	‘1′	‘10′	‘10′	‘100′	‘100′
1	‘01′	‘11′	‘110′	‘101′	‘101′
2	‘001′	‘010′	‘111′	‘110′	‘110′
3	‘0001′	‘011′	‘010′	‘111′	‘1110′
4	‘00001′	‘0010′	‘0110′	‘0100′	‘1111′
5	‘000001′	‘0011′	‘0111′	‘0101′	‘0100′
6	‘0000001′	‘00010′	‘0010′	‘0110′	‘0101′
⋮	⋮	⋮	⋮	⋮	⋮

clues of these facts, this study shows sample-based, block-based, and distribution-based shaping methods for respective modeling.

2.7 Conclusion of Chapter 2

This chapter explained the assumptions which are well-used in speech-and-audio coding models. It is shown that, from a statistical point of view, the signal modeling can often be interpreted as parameterizing the scales of the frequency-wise distributions. On the other hand, the code modeling gives the optimization method of the trade-offs between bit rates and distortion defined by perception modeling based on psycho-acoustic effects.

The widely used method was introduced, such as linear prediction for the signal modeling, companding methods for perception modeling, and entropy coding for the code modeling. Based on those conventional ideas, we showed the aim of the overall thesis. The conventional coding models will be extended by proposing six shaping methods based on the three types of modeling. From the next chapter, the concepts of each proposed shaping method are explained first.

Chapter 3

Signal model extension by sample-based shaping

3.1 Introduction of this chapter

The signal modeling defines the assumed distribution of the signals and thus determines the theoretical limits of compression efficiency. Therefore, the model should be reasonable as possible for representing the real speech-and-audio signals. In the sense of compression, how reasonable the models are can be discussed based on maximum-likelihood criterion because ideal expected bit rates of the encoded signals depend on the likelihood of the models when allowing the same distortion. In other words, in the same freedom of models, enhancing their likelihood can lower the distortion in the same bit rates and is expected to enhance the sound quality of the decoded signals.

This chapter focuses on the LP, one of the most important signal models, as mentioned in the previous chapter, and aims at extending to a more efficient model. Since the LP has simple assumptions in the frequency domain from a statistical point of view, and, in general, frequency-domain coding schemes are practically efficient for a broad class of sounds, we here use the ideas of its model in the frequency domain.

As explained in the previous chapter, the scales of the distributions estimated by the LP from the frequency spectra are represented as spectral envelopes. In the following sections, we propose shaping methods, from three independent approaches, making the envelope representation more reasonable. The first approach is for the model resolution, or the fidelity of the model, in each frequency. It makes use of the characteristics of the natural sounds that often have more dynamics in the lower frequencies than the higher frequencies. The second one is for the model distribution, the basic class of distributions p_X. It gives more reasonable signal assumptions while requiring a small increase in computational costs for the estimation of the model parameters. The third approach is for the model discreteness, for the fact that the target signals take discrete values due to quantization. It enables us to whiten the spectra, without losing any information, even if they are quantized and integer-valued. In other words, it realizes an algorithm to make the scales uniform in the domain of discrete values.

The methods coming from the above approaches tackle different challenges and can be combined as necessary. Therefore, founded on the requirements of the codecs, appropriately adopting these methods will enhance the compression efficiency.

3.2 Shaping by frequency warping focusing on model resolution

3.2.1 Motivation

Generally, the envelope extracted by the conventional LP has a uniform frequency resolution over the frequencies, depending on its prediction order, for fitting on the spectra because the envelope is composed of the reciprocal of the linearly-combined sinusoids. This limitation on the resolution causes, in some cases, low accuracy of scale estimation, which ends up in unexpected compression results given by the code modeling. Of course, this resolution can be enhanced by increasing the order of LP, but this also leads to the increase of the bits to code the LP coefficients and computational complexity to estimate the LP coefficients from the signal.

Most natural sounds have a relatively higher power at lower frequencies so that the information content of quantized signals tends to concentrate in the lower band. Fig. 3.1 shows an example of this nature. We calculated the log-likelihood of MDCT spectra assuming their distribution as Eq. (2.8) with the scales given by prediction orders up to 32. The figure plots how much the log-likelihood of each frequency enhanced in ratio when the prediction order doubled. Here, the enhancement by the first-order is valued as 1. The ratio was much larger in the lower frequencies than the higher ones, which comes

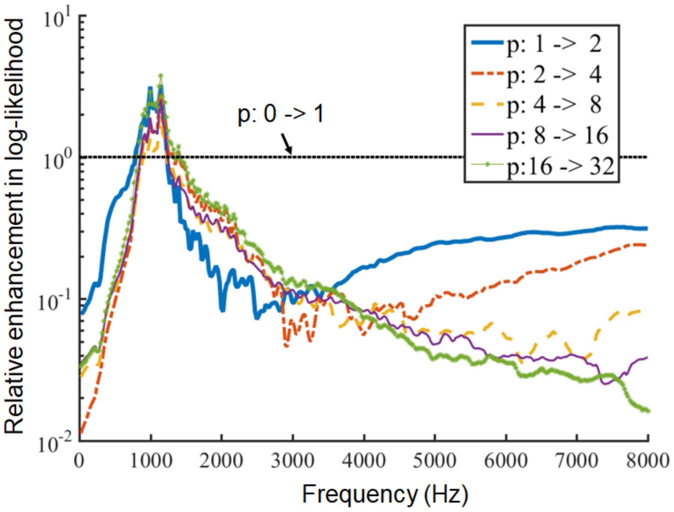


Fig. 3.1. Frequency-wise log-likelihood enhancement ratio by prediction order compared to the enhancement by the first order. Total 24720 frames of 320-sample MDCT coefficients were tested.

from the large dynamic range of low frequencies. Note that the frequencies near 0 and 8000 Hz have special characteristics derived from the nature of frame-wise processing, and it is difficult to enhance the likelihood. Taking these facts into account, we here propose a representation of the envelopes based on a model modified to warp the frequency resolution into a Mel-frequency scale [51], instead of using higher-order LP.

3.2.2 Resolution warping by non-negative sparse matrices

For warping the resolution of the envelopes, the most straightforward idea is to divide the spectra by several frequency bands and apply a different order of LP to each band. However, this method has too many tunable and non-intuitive factors, combinations of frequency bands, and their corresponding LP orders. Moreover, the envelopes significantly lose their continuity, especially when the target spectra have peaks at the boundary of the divided frequency bands, which leaves the risk of artifacts in the decoded signal.

The model in [52] is the well-known representation of continuous envelopes with warped resolution. This model is written as a modification of Eq. (2.4) with an all-pass filter approximating the frequency warping:

$$H(z) = \left(\sum_n a_n \left(\frac{z^{-1} - \delta}{1 - \delta z^{-1}} \right)^n \right)^{-1} \tag{3.1}$$

where δ stands for an extent of the warping, that approximates the Mel-frequency scale when $\delta = 0.41$ at 16-kHz sampling rate. This method of representation is also known as a special case of Mel generalized cepstrum analysis [53], called the Mel linear predictive coding (Mel-LPC) method, and its integration in several types of codecs has been considered before [54–57].

Based on the same sense of warping as Mel-LPC, the method proposed in this section provides a much simpler scheme with lower complexity owing to its limited use only in the frequency domain. The method uses the shaping of spectra by two $N \times N$ sparse non-negative matrices prepared in advance: one for approximating frequency warping of the power spectra; the other for approximating the inverse warping. Here, we simply call them the warping matrix and the inverse warping matrix, respectively.

The extraction of the envelopes is done as follows. First, the power spectra of the signal are shaped by the warping matrix $W \equiv \{W_{ij}\}$, having their frequency resolution warped, and inverse Fourier transformed. Then, as in the conventional LP, LP coefficients are estimated by performing the Levinson–Durbin algorithm with the transformed spectra regarded as a pseudo-auto-correlation function. Substituting the LP coefficients for Eq. (2.4) gives a frequency-warped envelope with its resolution uniform over the warped frequencies. Therefore, resetting the frequency by the inverse warping matrix results in an envelope with its frequency resolution warped. In other words, this shaping by frequency warping changes the model of the envelope as

$$H_k = \left(\sum_{i=0}^{N-1} U_{ki} \left| 1 + \sum_{n=1}^{p} a_n e^{-\mathrm{j}\frac{\pi i}{N} n} \right|^{-2} \right)^{\frac{1}{2}}, \tag{3.2}$$

with the $(i,\ j)$th element (counted from $(0,\ 0)$ for convenience) of the inverse warping matrix $U \equiv \{U_{ij}\}$. The square root of the prediction gain σ^2 given by the algorithm represents the gain of the envelope mentioned in the last chapter. This method differs from the conventional LP only in the additional matrix operations, which costs minor complexity when the matrices are sparsely designed. In addition, the Levinson–Durbin algorithm enables us to check the stability of the LP coefficients easily during the estimation, just as in the conventional LP.

3.2.3 Optimization of warping matrix

Here, we provide one way to design the warping and inverse warping matrices. The envelope extracted from the frequency-warped spectra, as mentioned above, gets inversely warped to reset the frequency axis. However, these warping and inverse warping operations make errors in the envelope, in the sense of *how high the accuracy of the envelope should be in the original frequency axis*, because of the inherent irreversibility of the warping operations. For that reason, the warping and inverse warping matrices must be designed according to not only the accuracy of the warping but also to the consistency of the warping and inverse warping operations. Therefore, the matrices are prepared here by an optimization using a training data set of spectra.

The training data set contains the power spectra of audio signals $X \equiv \{X_{kj}\}$ where X_{kj} is the spectrum of the kth linear (unwarped) frequency in the jth frame of the signals. The warped power spectra $Y \equiv \{Y_{k'j}\}$ are also given by sinc-interpolating the spectrum of the same signals in accordance with the function $f(k')$ based on the function in [51], which maps from the index k' $(0 \leq k' \leq N-1)$ of frequency sampled uniformly on the Mel-frequency scale to the corresponding index on the linear frequency scale, the same domain as k in the previous discussions:

$$f(k') = \frac{2f_0}{f_s} N \left(\exp\left(\frac{\ln(f_s/(2f_0)+1)}{N} k' \right) - 1 \right) \tag{3.3}$$

with $f_0 = 1000$ (Hz) and sampling rate $f_s = 16000$ (Hz). $\ln(\cdot)$ is for the natural logarithmic function. Note that $f(0) = 0$, $f(N) = N$. With these spectra, the warping and the inverse warping matrices $W \equiv \{W_{ij}\}$ and $U \equiv \{U_{ij}\}$ are optimized to satisfy

$$Y \approx WX \quad \text{and} \quad X \approx UWX, \tag{3.4}$$

respectively. Since the conventional LP is based on IS divergence between the spectra and their envelopes in the frequency domain, as explained in the last chapter, these approximations should also be measured by IS divergence.

These requirements lead to the objective function of the optimization for the warping matrix X:

$$\sum_{i,j} \left(\frac{Y_{ij}}{\sum_k W_{ik} X_{kj}} - \ln \frac{Y_{ij}}{\sum_k W_{ik} X_{kj}} - 1 \right). \tag{3.5}$$

This function stands for IS divergence of WX from Y. Here, this objective function has to be minimized by W, with all of its elements restricted to non-negative values. This minimization problem cannot be solved explicitly, making it necessary to take an implicit approach, such as the auxiliary function method introduced in [58].

First, only the terms in the objective function Eq. (3.5) related to W are considered:

$$\sum_{i,j} \frac{Y_{ij}}{\sum_k W_{ik} X_{kj}} + \sum_{i,j} \ln \left(\sum_k W_{ik} X_{kj} \right) \equiv L(W). \tag{3.6}$$

Since the reciprocal function $\frac{1}{x}$ is convex in $x > 0$, Jensen's inequality holds as

$$\frac{1}{\sum_k W_{ik} X_{kj}} = \frac{1}{\sum_k \lambda_{ijk} (W_{ik} X_{kj} / \lambda_{ijk})} \leq \sum_k \frac{\lambda_{ijk}}{W_{ik} X_{kj} / \lambda_{ijk}} \tag{3.7}$$

with a set of auxiliary variables $\{\lambda_{ijk}\ (\geq 0)\}$ that satisfies $\sum_k \lambda_{ijk} = 1$ for all i and j. In addition, the concavity of the logarithmic function leads to an inequality between its tangential line and itself as

$$\ln \left(\sum_k W_{ik} X_{kj} \right) \leq \ln \xi_{ij} + \frac{\sum_k W_{ik} X_{kj}}{\xi_{ij}} \tag{3.8}$$

with another set of auxiliary variables $\{\xi_{ij}\ (> 0)\}$. Applying both inequalities (3.7) and (3.8) to Eq. (3.6), the upper bound can be set to the function $L(W)$ by an auxiliary function as

$$L(W) \leq \sum_{i,j} Y_{ij} \sum_k \frac{\lambda_{ijk}^2}{W_{ik} X_{kj}} + \sum_{i,j} \left(\ln \xi_{ij} + \frac{\sum_k W_{ik} X_{kj}}{\xi_{ij}} \right) \equiv G(W) \tag{3.9}$$

where the equality is attained if and only if the auxiliary variables hold:

$$\lambda_{ijk} = \frac{W_{ik} X_{kj}}{\sum_k W_{ik} X_{kj}}, \quad \xi_{ij} = \sum_k W_{ik} X_{kj} \tag{3.10}$$

for all i, j and k. Because of the convexity, the auxiliary function $G(W)$ in Eq. (3.9), with $\{\lambda_{ijk}\}$ and $\{\xi_{ij}\}$ all fixed, can be uniquely minimized at the stationary point of W found as

$$\left. \frac{\partial G(W)}{\partial W_{mn}} \right|_{W=\tilde{W}} = \left(\sum_j Y_{mj} \lambda_{mjn}^2 / X_{nj} \right) \cdot \left(-\frac{1}{\tilde{W}_{mn}^2} \right) + \sum_j X_{nj} / \xi_{mj} = 0$$

$$\iff \tilde{W}_{mn} = \sqrt{\frac{\sum_j Y_{mj} \lambda_{mjn}^2 / X_{nj}}{\sum_j X_{nj} / \xi_{mj}}}. \tag{3.11}$$

At last, by Eqs. (3.10) and (3.11), $\{\lambda_{ijk}\}$, $\{\xi_{ij}\}$ and $\tilde{W}$ are iteratively updated, making the objective function decrease monotonically until it ends up in a local optimum. This iteration can be summarized as follows.

Both $\{\lambda_{ijk}\}$ and $\{\xi_{ij}\}$ are updated by $W^{(l)}$, which indicates $\tilde{W}$ in the lth iteration, in accordance with Eq. (3.10) as

$$\lambda_{ijk} = \frac{W_{ik}^{(l)} X_{kj}}{\sum_k W_{ik}^{(l)} X_{kj}}, \ \xi_{ij} = \sum_k W_{ik}^{(l)} X_{kj}. \tag{3.12}$$

Substituting them into Eq. (3.11) leads to $\tilde{W}$ in the $(l+1)$th iteration:

$$W_{mn}^{(l+1)} = \sqrt{\frac{\sum_j Y_{mj} W_{mn}^{(l)2} X_{nj} / \left(\sum_k W_{mk}^{(l)} X_{kj}\right)^2}{\sum_j X_{nj} / \sum_k W_{mk}^{(l)} X_{kj}}}. \tag{3.13}$$

This results in the following updating rule:

$$W_{mn}^{(l+1)} = W_{mn}^{(l)} \sqrt{\frac{\sum_j Y_{mk} X_{nj} / \hat{Y}_{mj}^2}{\sum_j X_{nj} / \hat{Y}_{mj}}}, \tag{3.14}$$

$$\hat{Y} = W^{(l)} X.$$

After the iteration for W, the inverse matrix U is also optimized as is done for W, to minimize IS divergence of UWX from X with the optimized W. The iteration for U can be written in the same form as in Eq. (3.14):

$$U_{mn}^{(l+1)} = U_{mn}^{(l)} \sqrt{\frac{\sum_j X_{mk} \hat{Y}_{nj} / \hat{X}_{mj}^2}{\sum_j \hat{Y}_{nj} / \hat{X}_{mj}}}, \tag{3.15}$$

$$\hat{Y} = W^{(l)} X, \ \hat{X} = U^{(l)} W^{(l)} X.$$

Both iterations introduced above are written in products of each element and positive values. Hence the objective function can be minimized in the sparse non-negative condition of W and U, as shown in Fig. 3.2, by choosing sparse non-negative matrices for the initial values of W and U. Note that by changing the mapping function $f(k')$ for calculating Y, it is possible to design different kinds of warping matrices flexibly.

3.2.4 Smoothing in LSP domain for perceptual noise control

As mentioned in Chap. 2, the envelopes are also used for perceptual control of quantization noise by smoothing them. However, simply applying the estimated LP coefficients $\{a_n\}$ to Eq. (2.9) means smoothing the envelopes with the warped-frequency scale regarded as the linear-frequency scale. Mismatch in scales causes changes in the properties of the

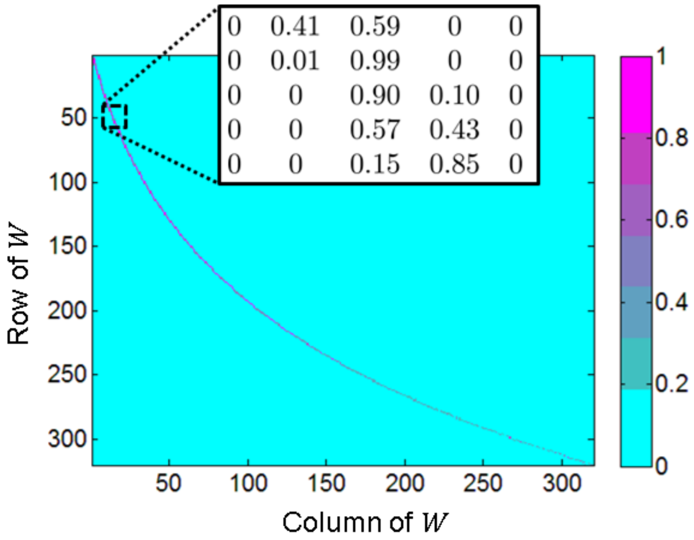


Fig. 3.2. Color mapping of the optimized warping matrix W and an example of its elements.

smoothing from those of the conventional one, resulting in unexpected shaping of the noise. Therefore, to approximate the conventional smoothing, we modify the parameter γ used in smoothing envelopes as

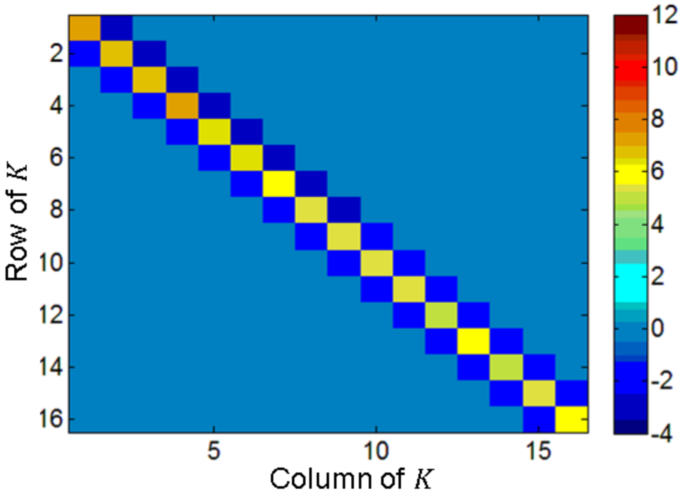
$$\tilde{H}_k = \begin{cases} \left(\sum_{i=0}^{N-1} U_{0i}\left|1+\sum_{n=1}^{p} a_n\gamma^n\right|^{-2}\right)^{\frac{1}{2}} & (\text{if } k=0) \\ \\ \left(\sum_{i=0}^{N-1} U_{ki}\left|1+\sum_{n=1}^{p} a_n\gamma^{\frac{g(k)}{k}n} e^{-\mathrm{j}\frac{\pi k}{N}n}\right|^{-2}\right)^{\frac{1}{2}} & (\text{otherwise}) \end{cases}, \tag{3.16}$$

lowering γ when the frequencies are stretched by the warping and heightening γ when the frequencies are squeezed. Note that we assumed here $g(k) = k$ when $k = 0$. $g(k)$ could either be the one in Eq. (6.5) or its approximation:

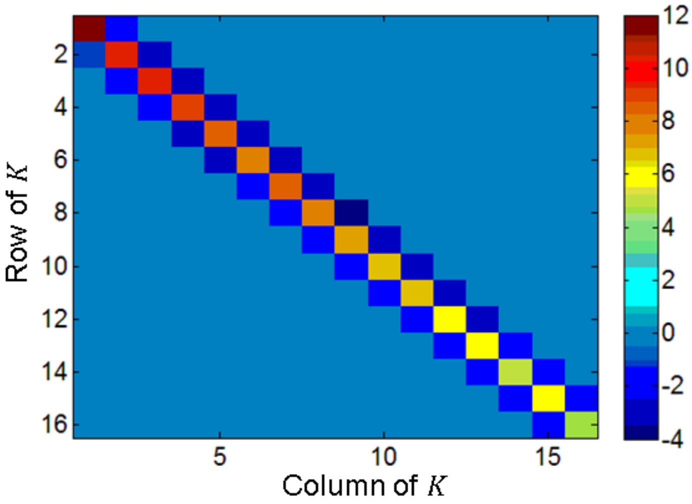
$$\begin{pmatrix} g(0) \\ \vdots \\ g(N-1) \end{pmatrix} \approx U \begin{pmatrix} 0 \\ \vdots \\ N-1 \end{pmatrix}. \tag{3.17}$$

This method smooths the spectra more strongly in the frequencies stretched by the warping (here, in the lower band) than in the frequencies squeezed (here, in the higher band).

However, the modification makes the smoothed envelopes unable to be calculated by Fourier transformation, which approximately doubles the complexity for the calculation. Thus, to save those computational costs, the envelopes should be smoothed in the LSP



(a)



(b)

Fig. 3.3. Color mappings of matrices K designed respectively for approximating the smoothing (a) for the conventional envelope and (b) for the resolution-warped envelope. Both matrices have positive values for the main diagonal elements, negative values for the elements of the first diagonal below and above the main diagonal, and zeros for the other elements.

domain, since the LP coefficients are often quantized and compressed in that domain, by the following low-cost and straightforward approximative conversion scheme. The smoothing is described with the LSP representation of the LP coefficients estimated by the proposed method $\{\omega_n\}_{n=1}^p$ and of the coefficients $\{\tilde{\omega}_n\}_{n=1}^p$ corresponding to the smoothed envelope as

$$\begin{pmatrix} \tilde{\omega}_1 \\ \vdots \\ \tilde{\omega}_p \end{pmatrix} = K^{\mathrm{LSP}} \begin{pmatrix} \omega_1 - \frac{\pi}{p+1} \\ \vdots \\ \omega_p - \frac{\pi p}{p+1} \end{pmatrix} (\gamma - 1) + \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_p \end{pmatrix} \tag{3.18}$$

where

$$K^{\mathrm{LSP}} = \begin{pmatrix} x_1 & y_1 & & & 0 \\ z_2 & x_2 & y_2 & & \\ & \ddots & \ddots & \ddots & \\ 0 & & & z_p & x_p \end{pmatrix}. \tag{3.19}$$

With the LP coefficients converted from LSP parameters $\{\tilde{\omega}_n\}_{n=1}^p$, the smoothed envelope is simply given by Eq. (3.2), which has no additional costs for smoothing. This LSP conversion of smoothing makes the values of the LSP parameters farther from each other linearly, with the conversion depending only on the values of the neighboring parameters. The main idea of this method is to utilize the relation between envelopes and their corresponding LSP parameter values. LSPs correspond to the positions of peaks and valleys in envelopes, for this case, in the *warped* frequency axis, and are set in even intervals over 0 to π when the envelope is completely flat. The parameters in each row of the conversion matrix K^{LSP} respectively indicate the extent to which this conversion makes the LSP parameter of each order closer to even intervals. By properly setting the matrix K^{LSP}, the operation of Eq. (3.18) can smooth more strongly the LSP parameters corresponding to the lower frequencies than the parameters corresponding to the higher frequencies, resulting in effects similar to those in Eq. (3.16).

The conversion matrix K^{LSP} is designed by optimization as in the case of the warping matrix, using a training data set of LSP parameters estimated from the audio signals. The optimization proceeds as follows.

First, the LSP parameters

$$\omega_j = (\omega_{1j}, ..., \omega_{pj})^{\mathsf{T}} \text{ and } \tilde{\omega}_j = (\tilde{\omega}_{1j}, ..., \tilde{\omega}_{pj})^{\mathsf{T}} \tag{3.20}$$

respectively corresponding to the envelope and to the smoothed envelope of the jth frame of the signals are prepared. Here, we use T for transpose of vectors and matrices. Since the parameters corresponding to the smoothed envelops cannot be explicitly represented, we approximate them by LSP parameters converted from the LP coefficients estimated from LP analysis of the envelopes smoothed by Eq. (3.16). Then, the K^{LSP} in Eq. (3.19) is chosen by minimizing the Euclidean distance between the LSP parameters:

$$\sum_j ||\,\omega_j - \hat{\omega}_j||_2^2. \tag{3.21}$$

To minimize Eq. (3.21) the conversion Eq. (3.18) is rewritten as

$$\tilde{\omega}_j = D_j\mathbf{x} + \omega_j, \tag{3.22}$$

where

$$D_j = \begin{bmatrix} d_{1j} & d_{2j} & & & & & & & 0 \\ & & d_{1j} & d_{2j} & d_{3j} & & & & \\ & & & & & d_{2j} & d_{3j} & d_{4j} & \\ 0 & & & & & & & & \ddots \end{bmatrix}, \tag{3.23}$$

$$d_{nj} = \left(\omega_{nj} - \frac{\pi n}{p+1}\right)(\gamma - 1), \tag{3.24}$$

$$\mathbf{x} = (x_1, y_1, z_2, x_2, y_2, ..., z_p, x_p)^{\mathsf{T}}, \tag{3.25}$$

and substituted for Eq. (3.21) resulting in

$$\sum_j \left\{\mathbf{x}^{\mathsf{T}} D_j^{\mathsf{T}} D_j \mathbf{x} - 2\Delta_j^{\mathsf{T}} D_j \mathbf{x} + \Delta_j^{\mathsf{T}} \Delta_j\right\} \equiv J(\mathbf{x}) \tag{3.26}$$

with $\Delta_j \equiv \omega_j - \tilde{\omega}_j$. Since this function $J(\mathbf{x})$ is convex, it can be minimized at the stationary point of $\mathbf{x}$ found as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}} J(\mathbf{x}) &= \sum_j \left\{2D_j^{\mathsf{T}} D_j \mathbf{x} - 2D_j^{\mathsf{T}} \Delta_j\right\} = 0 \\ \Longleftrightarrow \mathbf{x} &= \left(\sum_j D_j^{\mathsf{T}} D_j\right)^{-1} \cdot \left(\sum_j D_j^{\mathsf{T}} \Delta_j\right). \end{aligned} \tag{3.27}$$

Finally, K^{LSP} is defined by the parameters in $\mathbf{x}$.

To make more clear the meanings of the conversion matrix K^{LSP}, Fig. 3.3 compares K^{LSP} designed respectively for approximating the conventional smoothing described in

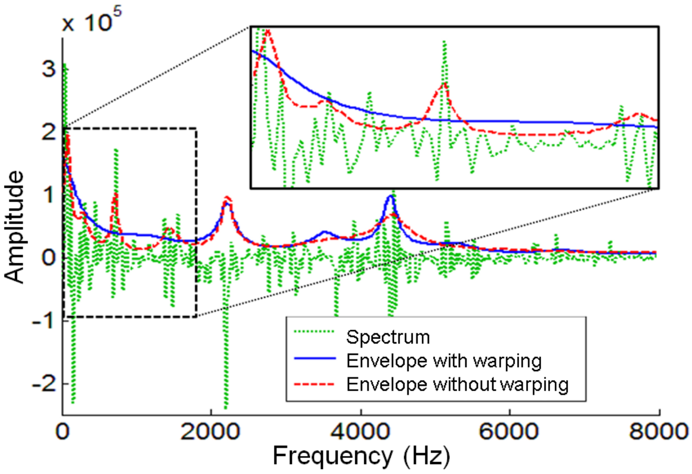


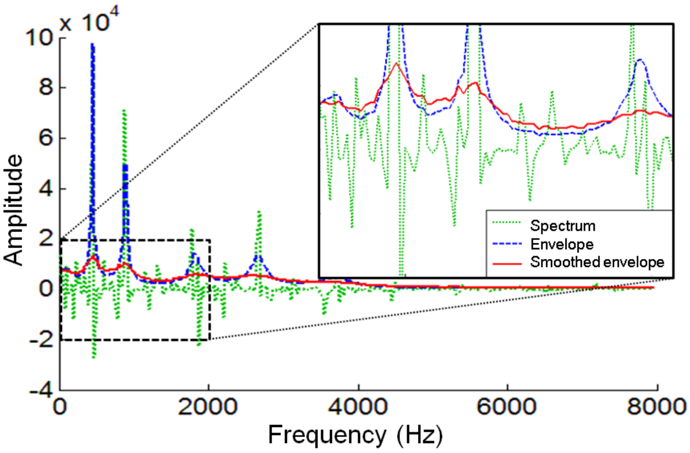
Fig. 3.4. Comparison of the envelopes in one frame. The green dotted line shows the MDCT spectrum of the input. The blue solid line and the red dashed line respectively indicate its envelopes extracted using linear prediction and the proposed method.

Eq. (2.9) and for approximating the smoothing for the resolution-warped envelope described in Eq. (3.16). It can be seen that the values in Fig. 3(b) are non-uniform over the diagonals with the upper-left elements being larger in absolute where the values in Fig. 3(a) are almost uniform over the diagonals, which means that the conversion using the matrix K^{LSP} in Fig. 3(b) smooths the envelope non-uniformly over the *warped* frequencies, to make the smoothing almost uniform over the *linear* frequencies, while the one using the matrix K^{LSP} in Fig. 3(a) smooths the envelope uniformly as the conventional smoothing does.

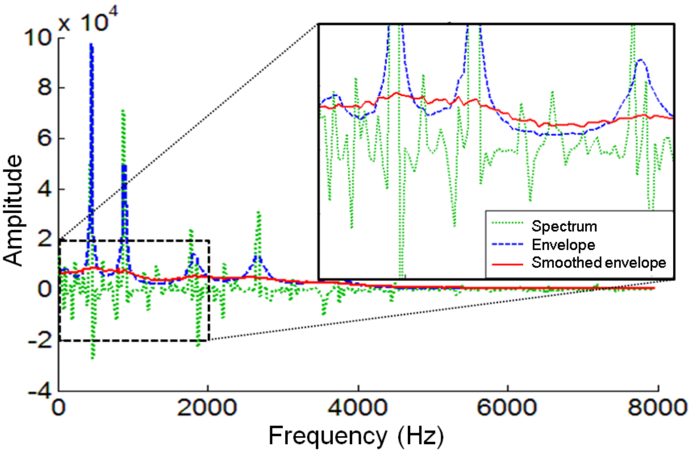
3.2.5 Examples of resolution-warped envelope and its smoothing

Here, we show some examples of the envelopes given by the method presented above.

Fig. 3.4 is a clear example of the comparison of the envelopes extracted from one frame by linear prediction and the method using the optimized matrix. The spectrum is presented in a linear-amplitude domain because it is quantized in this domain when coded. It is evident that the resolution of the envelope was warped: the accuracy of the envelope improved in the lower band, which is more important for the coding, at the cost of the higher band.



(a) Smoothed by perceptual weighting Eq. (2.9).



(b) Smoothed by the modified weighting Eq. (3.16).

Fig. 3.5. Resolution-warped envelope smoothed by each method. The blue dashed line and the red solid line respectively indicate the envelope before and after smoothing.

A comparison of the smoothing is shown in Fig. 3.5. Generally, smoothing by the weighting Eq. (2.9) has more influence on the steep peaks than on the gentle slopes of the envelopes. However, as Fig. 5(a), for example, shows, the weighting failed to smooth the resolution-warped envelopes, with peaks remaining in the lower band. This problem arose because the weighting Eq. (2.9) smoothed the envelope in the warped frequency domain instead of the linear domain. By modifying the weighting following Eq. (3.16), as shown

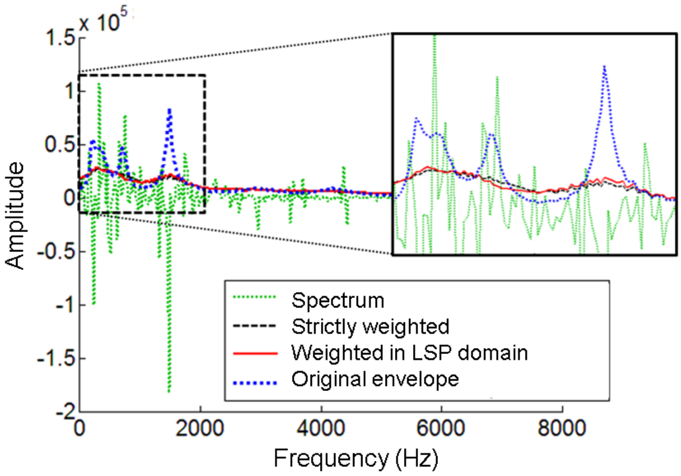


Fig. 3.6. Resolution-warped envelope smoothed in the LSP domain.

in Fig. 5(b), we were able to smooth the envelope appropriately by the warping. The local roughness in the smoothed envelopes was caused by the approximation errors in the warping and inverse warping.

Fig. 3.6 shows an example of applying the optimized conversion in the LSP domain. The red solid line is the envelope weighted in the LSP domain, while the black dashed line is the envelope strictly weighted by Eq. (3.16). The weighting model succeeded in approximating the weighting in the resolution-warped envelope. This method of weighting enables us to calculate the weighted envelopes by fast Fourier transform as in the case of regular envelopes and costs only additional $3p$ operations of multiplication.

3.3 Shaping by amplitude warping focusing on model distribution

3.3.1 Motivation

In designing signal models, we have an option to assume distributions for the target of the coding, and the more likely the actual target belongs to the assumed distribution, the more efficiently we can compress it by entropy coding designed based on the code modeling, which optimally allocates the bit length under the distribution. As mentioned in Chap.2, the conventional LP assumes Gaussian, which makes the model parameter optimally estimated by a low-complexity algorithm. Of course, there are some merits in assuming Gaussian because of its simplicity, but the distribution of the target may vary

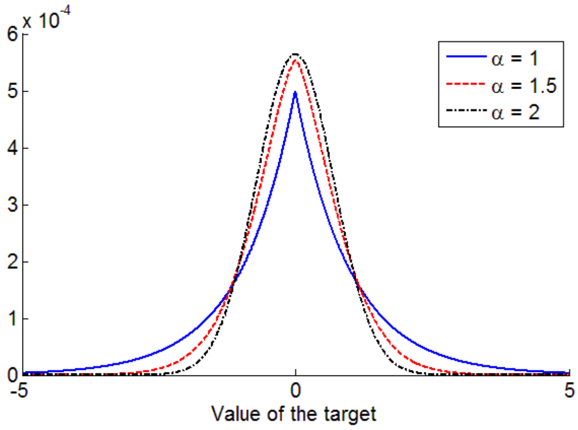


Fig. 3.7. Generalized Gaussian distribution of each shape parameter.

frame-by-frame depending both on acoustic properties and on coding conditions such as frame length. That is why many codecs, for example, use GR coding, which is optimal for Laplacian-distributed inputs and is not fit with LP in the sense of the assumptions. To fully use the statistical properties of the target and to further enhance the compression efficiency, we should be able to deal with a wider variety of distributions.

As a class of such distributions, generalized Gaussian distributions (GGDs) is well known in image and video analysis [59–63]. The zero-mean GGD for random variable X with a scale ϕ is represented as follows:

$$\tilde{p}_\alpha(X \mid \phi) = \frac{A(\alpha)}{\phi} \exp\left(-\left|\frac{B(\alpha)X}{\phi}\right|^\alpha\right) \tag{3.28}$$

where α is the shape parameter of this distribution, and

$$A(\alpha) = \frac{\alpha B(\alpha)}{2\Gamma(1/\alpha)}, \quad B(\alpha) = \sqrt{\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}} \tag{3.29}$$

with the gamma function:

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} \mathrm{d}t. \tag{3.30}$$

By changing the shape parameter α, we can represent respectively the Laplacian and the Gaussian at $\alpha = 1$ and $\alpha = 2$ as displayed in Fig. 3.7. There are some previous works for audio [59, 60] applied the model to the MDCT spectra. However, these works for audio had an assumption that the scales of the GGD are uniform over frequencies, which usually does not match the natural signals because their energy tends to concentrate on some frequencies.

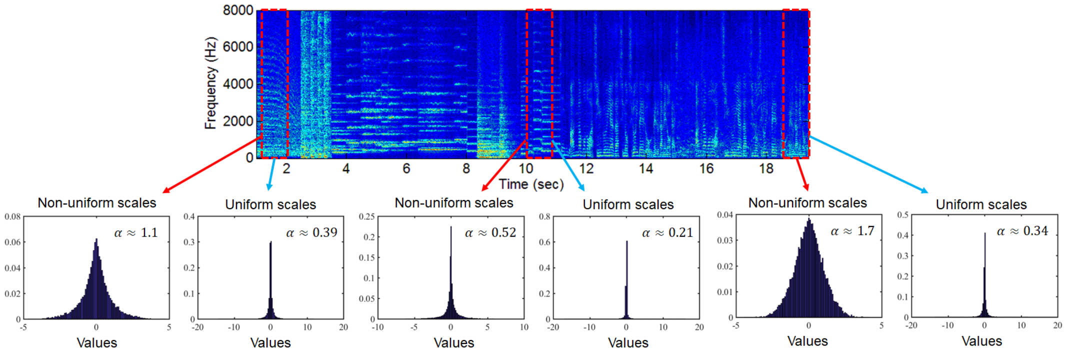


Fig. 3.8. Histograms of MDCT spectra normalized respectively by non-uniform and uniform scales estimated by LP.

Let us see some examples of the spectra. Fig. 3.8 shows a spectrogram of a music and speech signal. We picked up some frames of the MDCT spectra in the spectrogram, normalized each frame respectively by 16-th-order LP envelope (non-uniform scales) and ℓ_2 norm (uniform scales), and made their histograms. The shape parameter α was estimated by the moment-based method in [64]. The histograms for the non-uniform scales showed some variance, which seems to depend on acoustic features, while the ones for the uniform scales had a slight difference. The normalized spectra are equivalent to the prediction residuals, which corresponds to the quantization target of the coding, and thus modeling the distribution of the spectra by GGD is reasonable for encoding speech and audio signals, which have various statistical characteristics not limited to Gaussian.

Motivated by the above facts, this section presents a signal modeling based on the generalized-Gaussian-distributed spectra, which estimates the scale for each frequency from the spectral envelope by extending the LP model and presenting an optimal and simple method to extract the envelope, in other words, to parameterize the scales. Furthermore, we show that this method can be realized by a shaping based on an amplitude warping.

In the following discussion, we consider the situation of non-uniform scales $\{\phi_k\}_{k=0}^{N-1}$ for frequency spectra $\{X_k\}_{k=0}^{N-1}$ with a fixed α and parameterizing the maximum-likelihood $\{\phi_k\}_{k=0}^{N-1}$ for the given spectra.

3.3.2 Amplitude warping by powering operation

Here we consider the spectra $\{X_k\}_{k=0}^{N-1}$ as real values, supposing such as discrete cosine transform (DCT) or MDCT spectra, and introduce the optimization of the likelihood of their scales. There is an option to decide some of the following factors for this optimization:

- What kind of model to represent the spectral envelope;
- how to associate the envelope to the scales;
- how to extract the envelope.

For example, the conventional LP uses the all-pole model Eq. (2.4) for the envelope and extracts the envelope by the Levinson–Durbin algorithm, assuming the scale for each frequency to be the same as the corresponding value in the envelope. Nevertheless, this is not optimal for the Laplacian assumption often used in the code modeling. The previous works [65, 66] can be another example, which proposed LP optimizing the bit length for GR encoding of prediction residuals in the time domain based on minimizing the ℓ_1 norm of the residuals. If we interpret in the frequency domain, this idea uses the all-pole model Eq. (2.4) for the envelope and extracts the envelope by a numerical method called the auxiliary function method to optimize for the Laplacian assumed in the time domain. Here, it equivalently assumes the scale for each frequency to be the same as the corresponding value in the envelope. In other words, the algorithm to extract envelopes depends on the model of envelopes: If we fix the algorithm, we have to change the model of envelopes or the relation between envelopes and scales to optimize for the assumed distribution.

From the perspective of saving computational complexity, we design the model of the envelope that enables us to extract the envelope by the Levinson–Durbin algorithm. To design such a model, we have to transform the optimization problem of the likelihood into the form of Eq. (2.6). The negative log-likelihood for $\{X_k\}_{k=0}^{N-1}$, which also relates to the ideal description length, assuming the GGD with scale $\{\phi_k\}_{k=0}^{N-1}$ for each frequency is written as

$$\mathcal{L}_\alpha^{\mathrm{L}H}(\{X_k\}|\{\phi_k\}) = \sum_{k=0}^{N-1} -\log_2 \tilde{p}_\alpha(X_k \mid \phi_k). \tag{3.31}$$

This likelihood can be transformed as

$$\begin{aligned}
&\mathcal{L}_\alpha^{\mathrm{L}H}(\{X_k\}|\{\phi_k\}) \\
&= \sum_k (\log_2 e) \left[\left| \frac{B(\alpha) X_k}{\phi_k} \right|^\alpha + \ln \phi_k - \ln A(\alpha) \right] \\
&= \frac{1}{\alpha \ln 2} \sum_k \left[\frac{|X_k|^\alpha}{\phi_k^\alpha / (\alpha B(\alpha)^\alpha)} - \ln \frac{|X_k|^\alpha}{\phi_k^\alpha / (\alpha B(\alpha)^\alpha)} - 1 + 1 + \ln \alpha (|X_k| B(\alpha)/A(\alpha))^\alpha \right] \\
&= \frac{1}{\alpha \ln 2} \sum_k D_{\mathrm{IS}} \left(\frac{\phi_k^\alpha}{\alpha B(\alpha)^\alpha} \,\middle|\, |X_k|^\alpha \right) + C(\{X_k\}) \qquad (3.32)
\end{aligned}$$

where $C(\{X_k\})$ is a constant for $\{\phi_k\}_{k=0}^{N-1}$. Thus, by modeling the scale $\{\phi_k\}_{k=0}^{N-1}$ with a powered all-pole representation as

$$\phi_k \equiv \left(\frac{\alpha B(\alpha)^\alpha \sigma^2}{|1 + \sum_{n=1}^p a_n e^{-\mathrm{j} \frac{\pi n k}{N}}|^2} \right)^{1/\alpha}, \qquad (3.33)$$

the model parameters σ^2 and $\{a_n\}_{n=1}^p$ that minimize the negative log-likelihood $\mathcal{L}_\alpha^{\mathrm{LH}}$ are represented as

$$\begin{aligned}
&\operatorname*{arg\,min}_{\sigma^2, \{a_n\}} \mathcal{L}_\alpha^{\mathrm{L}H}(\{X_k\}|\{\phi_k\}) \qquad (3.34) \\
&= \operatorname*{arg\,min}_{\sigma^2, \{a_i\}} \sum_k D_{\mathrm{IS}} \left(\frac{\sigma^2}{|1 + \sum_{n=1}^p a_n e^{-\mathrm{j} \frac{\pi n k}{N}}|^2} \,\middle|\, |X_k|^\alpha \right).
\end{aligned}$$

This can be given by the Levinson–Durbin algorithm, as mentioned in the previous chapter, regarding $\{|X_k|^\alpha\}_{k=0}^{N-1}$ as *power* spectra, in other words, using for $\{r_n\}_{n=0}^p$ in Eq. (2.5) the inverse cosine transform of the spectra $\{|X_k|^\alpha\}_{k=0}^{N-1}$, shaped by amplitude warping, as

$$r_n = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^\alpha \cos\left(\frac{\pi n k}{N} \right). \qquad (3.35)$$

Moreover, the model parameters obtained above make $\{\phi_k^\alpha / (\alpha B(\alpha)^\alpha)\}_{k=0}^{N-1}$ match with $\{|X_k|^\alpha\}_{k=0}^{N-1}$ so that

$$H_{\alpha,k} = \left(\frac{\sigma^2}{|1 + \sum_{n=1}^p a_n e^{-\mathrm{j} \frac{\pi n k}{N}}|^2} \right)^{1/\alpha} \qquad (3.36)$$

represents the spectral envelope of $\{|X_k|\}_{k=0}^{N-1}$. This results in the relationship between the envelope and the scales:

$$\phi_k = (\alpha^{1/\alpha} B(\alpha)) H_{\alpha,k}. \qquad (3.37)$$

From now on, taking into account that the alternative model of scales is represented by a powered version of the conventional all-pole filter, we call this parameterization method of the scales, or the extraction of spectral envelope, *Powered all-pole spectrum estimation* (PAPSE). When the shape parameter of the GGD takes $\alpha = 2$, meaning the Gaussian, the maximum-likelihood scales $\{\phi_k\}_{k=0}^{N-1}$ is calculated from Eq. (3.33) and

$$B(2) = \sqrt{\frac{\Gamma(3/2)}{\Gamma(1/2)}} = \sqrt{\frac{\sqrt{\pi}/2}{\sqrt{\pi}}} = \frac{1}{\sqrt{2}} \tag{3.38}$$

as

$$\phi_k = \frac{\sigma}{|1 + \sum_{n=1}^{p} a_n e^{-\mathrm{j}\frac{\pi n k}{N}}|} = H_{\alpha=2,k}, \tag{3.39}$$

which matches to the conventional LP. Additionally, when the shape parameter takes $\alpha = 1$, meaning the Laplacian, the extraction of the envelope becomes the same as the one of the last section.

The inverse cosine transform of $\{|X_k|^\alpha\}_{k=0}^{N-1}$ in Eq. (3.35), which appears in the algorithm presented, is known as the zero phase signal representation [67–69] in the noise reduction context, and its application to LP is studied in other situations for the sake of its robustness against noise when a small α is used [70].

3.3.3 Shape parameter estimation with PAPSE

As stated above, we can obtain, for a given shape parameter α, the maximum-likelihood scales of GGD. However, the appropriate shape parameter may change momentarily depending on some sparseness of the spectra and is expected to reflect the acoustic features of the observation. Here, our concern is how to find smartly, among various shape parameters in the PAPSE scheme, the best parameter to represent the observation for each frame, namely the frame-by-frame maximum-likelihood α.

Although there are some previous works on estimating the shape parameter, merely applying them to the PAPSE scheme leads to inaccurate results: Methods such as moment-based method and maximum-likelihood estimation in [64] assume uniform scales over the observation, which conflicts to the PAPSE model; methods for multivariate GGD as in [71] require several observations belonging to the same distribution, which is hard to collect for audio signals because their distributions are varying momentarily.

Therefore, we present here an iterative algorithm for simultaneously estimating the shape parameter and the scales of GGD based on the method in [64]. This algorithm is composed of two steps: PAPSE step and shape parameter estimation step. At first, we set an initial value for α and iteratively perform the following steps for the observed spectra $\{X_k\}_{k=0}^{N-1}$:

1. (PAPSE step) Estimate the maximum-likelihood scales $\{H_{k,\alpha}\}_{k=0}^{N-1}$ by PAPSE of the present α;
2. (Shape parameter estimation step) since the normalized spectra $\{Y_k \equiv X_k/H_{k,\alpha}\}_{k=0}^{N-1}$ has approximately uniform scales, estimate the shape parameter from $\{Y_k\}_{k=0}^{N-1}$ by the method in [64] and update α.

Our preliminary test showed that the maximum-likelihood estimation in [64] often results in negative αs, which causes computational instability. Thus after this, we use the moment-based method to approximate maximum-likelihood estimate of α in the shape parameter estimation step. In principle, the moment-based method estimates α by solving

$$F(\alpha) \equiv \frac{\Gamma(2/\alpha)}{\sqrt{\Gamma(1/\alpha)\Gamma(3/\alpha)}} = \frac{m_1}{\sqrt{m_2}} \tag{3.40}$$

where m_1 and m_2 are respectively the empirical first and second moments of $\{Y_k\}_{k=0}^{N-1}$. Although we cannot explicitly calculate the inverse of $F(\alpha)$, its closed-form approximation is also proposed in [64]. When I is small enough, it is easier to choose α from its candidates $\{\alpha_i\}_{i=1}^{I}$ that makes $F(\alpha_i)$ closest to $m_1/\sqrt{m_2}$.

Summarizing the above, we can estimate the shape parameter α from a given frame of spectra by an iteration of closed-form analysis. If the method used in the shape parameter estimation step gives a maximum-likelihood estimate of α, the algorithm proposed above makes the likelihood monotonically increase by the iteration, which proves its convergence. However, because of the approximation of shape parameter estimation by the moment-based method, we cannot prove the convergence or optimality of this algorithm. Therefore, its validity must be checked by some experiments, which will be done in the later chapter.

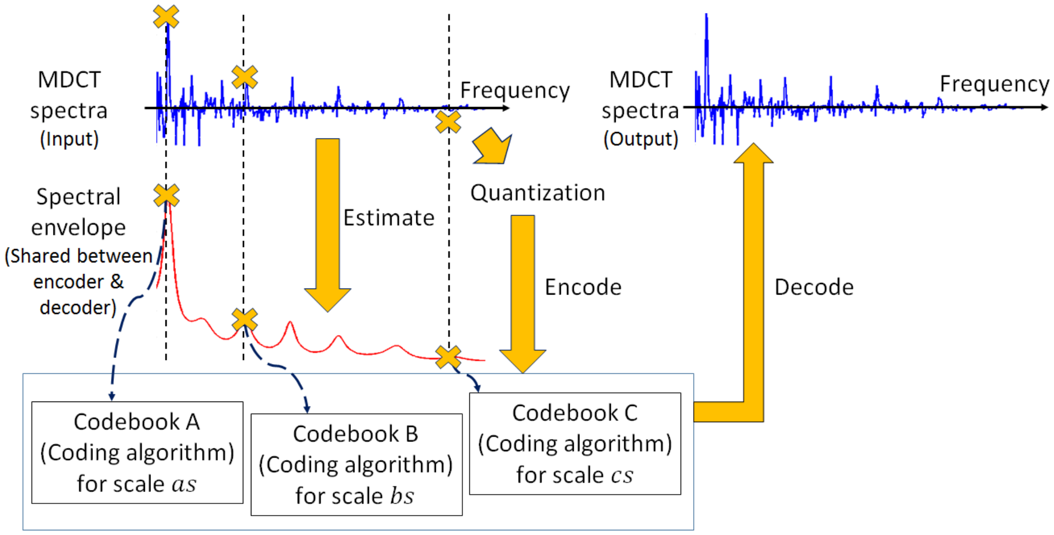


Fig. 3.9. Example of the use of the conventional spectral envelope in encoding and decoding.

3.4 Shaping by bit plane rearrangement focusing on model discreteness

3.4.1 Motivation

The above sections were aimed at the envelope estimation of the input spectra, which can tune the resolution and the shape parameters of the assumed distributions. In this section, we mainly focus on the use of the envelope for discrete-valued frequency spectra.

As introduced in the previous chapter, the spectral envelopes are representing the scales of the distributions of each frequency. Therefore, to fully make use of the envelopes in encoding spectra, we have to quantize the spectra and then apply some codebooks or coding algorithms optimized for the corresponding scales. This is described in Fig. 3.9: Say there are quantized spectra $\hat{X}_0$, $\hat{X}_1$, and $\hat{X}_2$, and corresponding estimated envelope values as, bs, and cs, the values shared between the encoder and the decoder via quantized model parameters. This envelope gives the distribution assumptions for the spectra as $p_X(X|as)$, $p_X(X|bs)$, and $p_X(X|cs)$, which become the clues for the codebooks encoding each quantized spectrum.

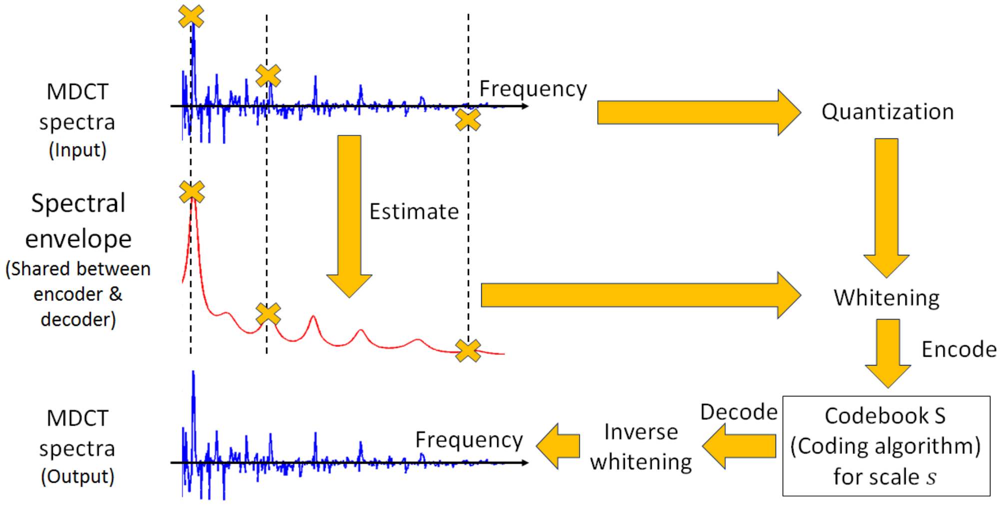


Fig. 3.10. Example of the use of the proposed spectral envelope in encoding and decoding.

This scheme requires different codebooks or algorithms for each of the frequencies based on their different scales, and thus it is preferable, from the perspective of computational resources, to shape the spectra into uniform scales using the envelopes. In other words, if we can whiten the spectra as $\hat{X}_0/a$, $\hat{X}_1/b$, and $\hat{X}_2/c$, their assumed distribution will be uniform as $p_X(X|s)$, as long as the envelope estimation is precise enough. However, the whitened spectra divided by the envelope values are not integers in general, which cannot be encoded directly. That is the reason why this section proposes another shaping method that realizes the invertible 'division' of envelopes in the integer domain, enabling us to design a coding scheme as in Fig. 3.10 that requires only one pattern of codebook or coding algorithm.

In time-domain LP, we can realize this kind of whitening, as in MPEG-4 ALS [11, 12], by using rounding operation because it is a convolution. It makes the codec design more flexible to realize this kind of whitening in the frequency domain for spectral envelopes not limited to the LP model. One simple idea to implement the whitening is to use the logarithmic envelope of the base 2, round its values, and remove the least-significant bits (LSBs) of each quantized spectrum depending on the corresponding rounded values in the logarithmic envelope. This operation is similar to the one in GR encoding introduced in the previous chapter, which divides the input integers into the reciprocal and remainder of powers of 2. However, this method can only be used where the logarithmic envelope values

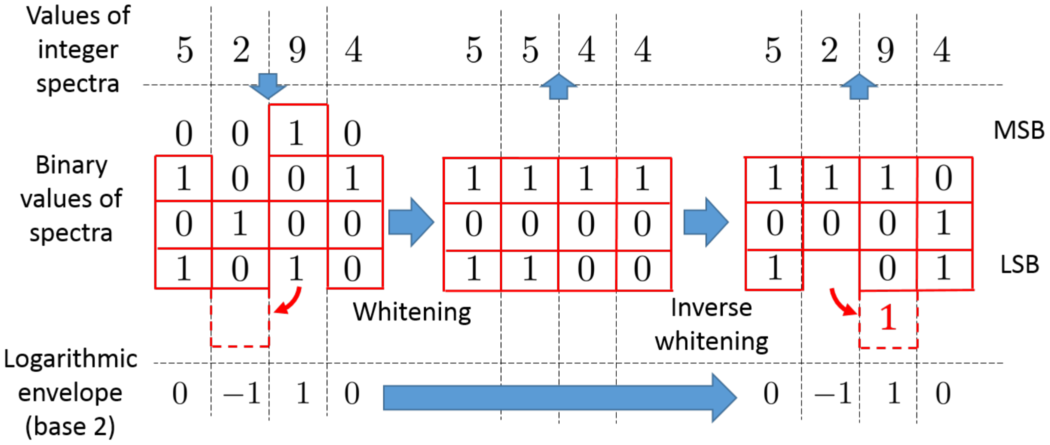


Fig. 3.11. Whitening of integer spectra with LSBM. Using the logarithmic envelope, we can inverse transform the integer spectra without any loss of information.

more than 0 and thus cannot be applied for small-scaled spectra. Therefore, the following section proposes a smarter method that shapes the integer values in the binary domain to achieve more efficient whitening. We focus on the absolutes of spectra, of which poles or phases are usually coded independently because of the symmetry of assumed distributions.

3.4.2 Bit plane rearrangement by least significant bit management

Here, we can use the property that the logarithmic values of the spectral envelopes: They can be divided into the terms constant over frequencies and ones variable and summing up to zero over frequencies. This property enables us to realize invertible division in the integer region by managing the LSBs of the quantized values of the spectra. For instance, in the LP model shown in Eq. (3.39), the σ corresponds to the constant term, and the rest corresponds to the variable terms of which logarithm sum up to zero over the frequency k. The proposed method, LSB management (LSBM), performs a rearrangement of bit plane as in Fig. 3.11, to spectra $\{5, 2, 9, 4\}$ as an example. The logarithmic envelope values are first divided into the constant term, say 3, and the variable terms aggregating zero, say $\{0, -1, 1, 0\}$. Then, the LSBs are taken off from the spectra following their corresponding variable terms if they are positive: 1 bit taken off from 9. If they are negative, the LSBs taken off are set to the LSBs of the spectra following them: 1 bit set to 2. Since the total of the variable terms is zero, if the envelope estimation is performed with enough precision, the processed spectra can be represented within the range depending on the constant term: $\{5, 5, 4, 4\}$ are all in the range of 0 to 2^3. By predetermining the rules of the order for taking and setting the LSBs, the correct quantized spectra can be reconstructed by reversing the processes.

This shaping method of integer spectra can approximately make their scales uniform, enabling us to compress them by a single codebook or coding algorithm optimally. It is especially useful where we have to deal with a strictly-fixed bit rate, which will be shown in Chap. 8.

3.5 Conclusion of Chapter 3

This chapter focused on the signal modeling of speech and audio, mainly based on the spectral envelope that represents and parametrizes the scales of the distributions assumed for frequency spectra. Since how to estimate and apply the envelopes greatly influences the compression efficiency of the overall coding processes, we proposed three independent shaping methods to enhance the usefulness of the envelopes.

The first one was frequency warping by non-negative sparse matrices, which shapes the spectra before LP to warp the frequency resolution of the all-pole model representing envelopes. Additionally, a technique was explained for using the envelope as perceptual weights in this case.

The second one was amplitude warping by powering operations, which also shapes spectra before LP to change the assumption of spectra into a more comprehensive class of distribution, generalized Gaussian distribution. Also, an estimation technique was presented to find the appropriate shapes of assumed distribution for respective input frames.

The third one was whitening by bit-plane rearrangement, which shapes the quantized spectra to realize an invertible division by their envelope in integer domain for making their scales uniform over frequencies.

All the three shaping methods enables us to design coding schemes with more reasonable assumptions of speech and audio while saving the increase in required computational resources. The actual applications and evaluations of them will be presented in the later chapters.

Chapter 4

Perception model extension by block-based shaping

4.1 Introduction of this chapter

As mentioned in Chap. 2, perception modeling often depends on the overall coding scheme, the strategies of optimizing the trade-offs of rate and distortion. For perception modeling, some conventional schemes using LP such as [40, 72] approximate the psycho-acoustic model by weighting based on smoothed spectral envelopes. However, they may be suboptimal from the signal modeling point of view because they simultaneously use the estimated LP coefficients both for deciding the distributions and perceptual weighting for the signals while the coefficients are optimal only for estimating the distributions of unweighted signals. In other cases, such as companding used in nearly instantaneous compandable audio matrix (NICAM) [39], the perception modeling is included in the quantization processes, which need careful tuning of assigning bits depending on total bit rates.

This chapter aims to introduce a qualitative approximation of the psycho-acoustic model, which emphasizes spectral valleys and deemphasizing spectral peaks and their neighboring frequencies without affecting the signal modeling nor code modeling, in other words, the perception modeling available for any coding strategies. This model is expected to make the codec design more flexible and enhance the affinity of the other proposed methods.

From the motivation mentioned above, the method for perception modeling is preferable to have no auxiliary parameter required to be shared between the encoder and decoder. Therefore, we here show a shaping method extending the usage of the block companding in NICAM into a non-parametric process. The block companding in NICAM is a parametric method that can satisfy the characteristics required for psycho-acoustic models. It decides in the NICAM encoder the block-wise representatives, represents them respectively as headers, and compands the blocks by scaling based on their representatives. Then in the NICAM decoder, the blocks are inversely scaled based on the representatives read from the headers. It is expected that applying this kind of companding to some subband spectra will approximate the perceptual weighting. To avoid the auxiliary parameters, the proposed method, self-determined block companding (SDBC), is designed as headerless by modifying the way of deciding the representatives and their corresponding scaling. Besides, avoiding the headers reduces bit rates as well as the redundancies between the block-wise representatives.

The following sections explain the proposed extensions of the block companding and its use in the perceptual weighting.

4.2 Shaping by block companding focusing on perceptual weighting

4.2.1 Self-determined block companding

Let us first show the processes in the proposed companding method applying to, for example, some real-valued frequency spectra X_k. Taking into account that the representatives are expected to represent some kind of magnitude of their blocks, SDBC uses norms for evaluating the representatives such as ℓ_1 norm:

$$\bar{X} \equiv f_{\mathrm{r}}(\{X_k\}_{k=0}^{M-1}) \equiv \frac{1}{M} \sum_{k=0}^{M-1} |X_k| \tag{4.1}$$

where M respectively indicate the block length.

As block companding not requiring any headers or auxiliary parameters, the scaling of spectra $\{X_k\}_{k=0}^{M-1}$ to $\{Y_k\}_{k=0}^{M-1}$ must have an inverse scaling using only the information of $\{Y_k\}_{k=0}^{M-1}$ while still meeting the characteristics mentioned above. Therefore, the scaling of the block in SDBC is defined as follows:

$$Y_k \equiv \frac{g_\mathrm{c}\left(\bar{X}\right)}{\bar{X}} X_k \ \ (k = 0, \ldots, M-1) \tag{4.2}$$

where $g_\mathrm{c}\left(\bar{X}\right)$ is a companding function using a generalized logarithmic function [73] and constants μ_c and γ_c as

$$g_\mathrm{c}\left(\bar{X}\right) \equiv \frac{\mathrm{glog}\left(1+\mu_\mathrm{c}\left|\bar{X}\right|, \gamma_\mathrm{c}\right)}{\mathrm{glog}(1+\mu_\mathrm{c}, \gamma_\mathrm{c})}; \tag{4.3}$$

$$\mathrm{glog}(x, \gamma_\mathrm{c}) \equiv \begin{cases} (x-1)/\gamma_\mathrm{c} & (\text{if } \ 0 < |\gamma_\mathrm{c}| \leq 1) \\ \log(x) & (\text{if } \ \gamma_\mathrm{c} = 0) \end{cases}. \tag{4.4}$$

The companding function above can be tuned by changing γ_c from $\gamma = 0$, corresponding to μ-law [1], to $\gamma_\mathrm{c} = 1$, corresponding to no companding.

By applying the scaling, the norm of the block will be companded as $g_\mathrm{c}\left(\bar{X}\right)$, which is the trick for the inverse companding. In the inverse companding, since there is no auxiliary parameter representing the original representative $\bar{X}$, the representative of the companded block is evaluated as is done in the companding:

$$\bar{Y} \equiv f_\mathrm{r}(\{Y_k\}_{k=0}^{M-1}). \tag{4.5}$$

Owing to the positive homogeneity of the norm $f_\mathrm{r}(\{Y_k\}_{k=0}^{M-1})$, Eq.(4.5) can be rewritten by inserting Eq.(4.2) as

$$\bar{Y} = f_\mathrm{r}\left(\left\{\frac{g_\mathrm{c}\left(\bar{X}\right)}{\bar{X}} X_k\right\}_{k=0}^{M-1}\right) = \frac{g_\mathrm{c}\left(\bar{X}\right)}{\bar{X}} f_\mathrm{r}\left(\{X_k\}_{k=0}^{M-1}\right) = \frac{g_\mathrm{c}\left(\bar{X}\right)}{\bar{X}} \bar{X} = g_\mathrm{c}\left(\bar{X}\right). \tag{4.6}$$

Therefore, applying an inverse scaling using the inverse companding function $g_\mathrm{c}^{-1}\left(\bar{Y}\right)$ to the companded block results in

$$\frac{g_\mathrm{c}^{-1}\left(\bar{Y}\right)}{\bar{Y}} Y_k = \frac{\bar{X}}{g_\mathrm{c}\left(\bar{X}\right)} Y_k = \frac{\bar{X}}{g_\mathrm{c}\left(\bar{X}\right)} \cdot \frac{g_\mathrm{c}\left(\bar{X}\right)}{\bar{X}} X_k = X_k \tag{4.7}$$

because of Eq.(4.2). Eventually, the original spectra $\{X_k\}_{k=0}^{M-1}$ get inverse companded from $\{Y_k\}_{k=0}^{M-1}$ without any auxiliary parameters.

4.2.2 Perceptual weighting by companding

To use SDBC for the perceptual weighting of audio spectra, we can apply the companding to subbands of the input signals. For instance, we here show a simple scaler quantization for N-length spectra $\{X_k\}_{k=0}^{N-1}$ combined with it, illustrated in Fig. 4.1.

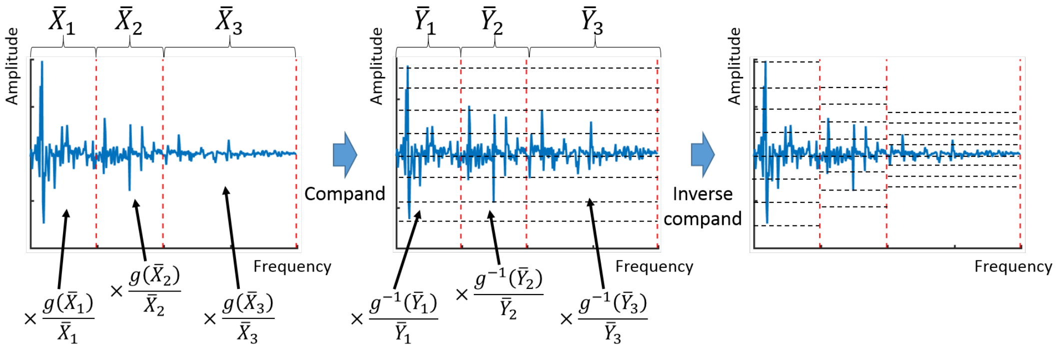


Fig. 4.1. Example of block companded real-valued spectra given by discrete cosine transform.

In companding, SDBC first evaluates the representatives for given subbands $[K_0, K_1)$, $[K_1, K_2)$, …, $[K_{I-1}, K_I)$ $(0 = K_0 < K1 < \cdots < K_I = N)$:

$$\bar{X}_i = f_\mathrm{r}\left(\{X_k\}_{k=K_i}^{K_{i+1}-1}\right) \quad (i = 0, \ldots, I-1). \tag{4.8}$$

Then, the blocks are scaled in accordance with their representatives for each $i = 0, \ldots, I-1$ as

$$Y_k = \frac{g_\mathrm{c}\left(\bar{X}_i\right)}{\bar{X}_i} X_k \ (k = K_i, \ldots, K_{i+1} - 1), \tag{4.9}$$

and $\{Y_k\}_{k=0}^{N-1}$ becomes the companded spectra.

In inverse companding, SDBC evaluates the representatives from the companded spectra as is done in the companding:

$$\bar{Y}_i = f_\mathrm{r}\left(\{Y_k\}_{k=K_i}^{K_{i+1}-1}\right) \quad (i = 0, \ldots, I-1). \tag{4.10}$$

Then, the blocks are inversly scaled in accordance with their representatives for each $i = 0, \ldots, I-1$ as

$$X_k = \frac{g_\mathrm{c}^{-1}\left(\bar{Y}_i\right)}{\bar{Y}_i} Y_k \ (k = K_i, \ldots, K_{i+1} - 1), \tag{4.11}$$

and the original spectra $\{X_k\}_{k=0}^{N-1}$ are given if the companded spectra are not quantized.

Since subbands including spectral peaks tend to have larger-valued representatives, SDBC generally deemphasizes the spectral peaks and their neighboring spectra compared to spectral valleys, satisfying the characteristics of perceptual weighting. Therefore, quantizing uniformly in the companded domain makes the quantization noise shaped approximately based on the masking threshold. Moreover, SDBC is invertible as long as the function $f_{\mathrm{r}}(\{X_k\}_{k=0}^{M-1})$ is positive homogeneous and the companding function $g_{\mathrm{c}}\left(\bar{X}\right)$ is monotonic, which enables us to design companding flexibly.

The above example of the use of SDBC only applied simple scalar quantization. However, this companding shapes the input to have uniform perceptual importance so that it can be combined with any optimization processes that minimize simple criteria such as square errors and absolute errors.

4.3 Conclusion of Chapter 4

This chapter presented a shaping method for perception modeling that equalizes the perceptual importance without any additional information. The proposed companding method, SDBC, performs block-wise scaling of subband frequency spectra qualitatively approximating the perceptual weighting based on psycho-acoustic models. Additionally, it is carefully designed to be invertible and thus capable of being integrated freely into various kinds of signal and code modelings, or optimization strategies in compression. Its actual effects on perceptual quality of the codecs are evaluated in the later chapters with some explanations of applications it enables us to design.

Chapter 5

Code model extension by distribution-based shaping

5.1 Introduction of this chapter

Code modeling is the basis of the coding scheme, and their performance influences the potential of the overall coding efficiency. They can be divided into two types: Table-based models, which are practically dependent on coding tables using, for example, Huffman coding and arithmetic coding [46, 74] after some quantization or sometimes direct coding by vector quantization; structured models, which are independent of coding tables because of using well-structured codes such as GR codes [41, 42] and their derivatives after some quantization. Indeed, the table-based models can construct theoretically optimal codes for many classes of input distributions, but they often require many memory, computational costs, or limitations on the input values. On the other hand, structured models do not need the limitations nor high computational costs, but they lack in flexibility for the sake of their small classes of input assumptions.

Lossless source codes, in other words, entropy codes, especially FV codes, play an essential part, both theoretically and practically, in those models. Even in lossy source coding, FV codes are used with quantization processes, and it is known that optimal FV codes can achieve the fundamental limits of rate-distortion by combining with appropriate quantization [75, 76]. The GR code mentioned above is widely used among such codes, well-structured and sometimes extended or combined with other codes, [46, 48–50, 77–79].

As explained in the previous chapter, the conventional GR code has a coding parameter $R(\geq 0)$, from now on Rice parameter, and the input positive integers are first divided into the quotient and the remainder of 2^R, by using bit-shift operations for example. The quotient and the remainder are respectively represented by a prefix unary and an R-bit suffix code. Thus, the code length of GR code for a positive integer X, omitting the rounding effects, is written as

$$\mathcal{L}_1^{\mathrm{GR}}[X|2^R] = \frac{X}{2^R} + R + 1, \tag{5.1}$$

corresponding to its approximately optimal distribution $p_1[X|2^R]$ where

$$p_1[X|\phi] = \frac{1}{Z_1(\phi)} 2^{-\frac{X}{\phi}}, \;\; Z_1(\phi) = \sum_{k=0}^{\infty} 2^{-\frac{k}{\phi}} \tag{5.2}$$

with $\phi = 2^R$. In other words, the GR code of a Rice parameter R is optimal for a discrete Laplacian distribution with a scale of 2^R, or a geometric distribution $p_{\mathrm{Geo}}[X|2^{-1/2^R}]$ using Eq. (2.10). This can be used as an approximation of the continuous Laplacian distribution for a positive real number x,

$$\tilde{p}_1(x|\phi) = \frac{1}{\phi} \exp\left(- \left| \frac{x}{\phi} \right| \right), \tag{5.3}$$

a well-known distribution for assuming the statistics of frequency spectra [59, 80, 81]. Owing to its optimality and simple arithmetic for arbitrary positive integers requiring low memory and computational cost, GR code is used in many standards such as ITU-T G.711.0 [50], MPEG-4 ALS [11, 12, 49], SHORTEN [48], and Lossless JPEG (JPEG-LS) [77, 82].

However, in low-rate lossy source coding, symbols given by quantized source inputs often contain a large portion of small-valued integers with many zeros, corresponding to the distributions with small scale ϕ: Distribution of coarsely quantized sources has small scales, while the counterpart of finely quantized sources has large ones. For the non-negativity of Rice parameter R, GR code cannot adapt to source symbols belonging to the distribution with scale ϕ smaller than one. This fact results in a more significant gap between the average GR code length and the source entropy as the scale ϕ gets small, namely, as the entropy of the source gets low. Moreover, as explained in Chap. 3 and also investigated in Chap. 6, the assumptions will be more reasonable for speech and audio signals to use a more general class of distributions.

The goal of this chapter is to design, maintaining the low complexity, a more flexible well-structured code useful for the code modeling in speech and audio coding. Therefore, we here focus on GR code and propose two independent shaping methods to efficiently use it for various scales and shapes of distributions seen in speech and audio signals. One is for extending GR code to adapt to Laplacian distributions with lower scales, which enables us to approximately cover the whole range of scales by combining it with the conventional one. Another one is for extending GR code to adapt to the GGDs [64, 71], which can be easily integrated into the first extension.

The following sections show that these kinds of extensions can be realized by controlling the scales and shapes of the distributions by some shaping methods, approximately transforming them into Laplacian distributions with large-valued scales, which is optimal for the conventional GR encoding. Although the following discussions mention only the cases of non-negative integer inputs, the whole range of integer can be dealt with, for instance, by representing the polarity by one bit for each non-zero symbol or by mapping the symbols by folding and interleaving as in [83].

5.2 Shaping by bit inverting for sparse-distributed sources

5.2.1 Bit inverting for shaping scales

We here focus on sparse inputs, containing a large portion of small-valued integers with many zeros and belonging to small-valued scales. They are often seen in some cases as in [79, 84, 85], where the scaler quantizer uses relatively large quantization steps compared to the standard deviation of the sources. Among them, the code in [84] combines run-length code with GR code and practically tunes the code length for runs of zeros to encode such quantized source symbols, whose entropy gets lower by the coarse quantization. Indeed applying appropriate run-length codes makes lower average code length than simple GR code for such symbols, but its optimality is not guaranteed. Therefore, this section aims at realizing an extension of GR code, by simple arithmetic, for optimally encoding low-entropy Laplacian-distributed source symbols.

The basic idea of optimal codes for sparse source symbols can be interpreted as follows. Let us think of, for example, stationary memoryless source symbols belonging to small-scaled Laplacian distributions $p_1[X|(K+1)^{-1}]$, where K is non-negative. The ideal code length for each source symbol X is written as

$$\begin{aligned}\mathcal{L}_{\text{ideal}}[X|(K+1)] &= -\log_2 p_1[X|(K+1)^{-1}] \\ &= -\log_2 \frac{1}{Z_1\left((K+1)^{-1}\right)} 2^{-(K+1)X} \\ &= (K+1)X + \log_2 Z_1\left((K+1)^{-1}\right), \end{aligned} \tag{5.4}$$

which become, as the sources get sparser and K larger, longer at $X \neq 0$ and shorter at $X = 0$. To achieve such codes, we have to reinterpret the Laplacian sources using the definition of geometric sources.

Let us focus on encoding a geometric source $p_{\text{Geo}}[X|\theta]$ using unary codes, with each of their codewords containing runs of code symbol ‘0’s and a single one ‘1’. The emergence rates of the code symbols ‘0’ and ‘1’ are respectively θ and $(1-\theta)$, which is trivial interpreting the code symbols as Bernoulli trials. The inversion of the encoded codewords reverses the emergence rates with the code symbols, which can still be regarded as Bernoulli trials. Thus, unary decoding the inverted codewords changes the input geometric source into $p_{\text{Geo}}[x|1-\theta]$. Note that, in practice, if the unary code ends with a code symbol ‘0’, we have to terminate it by ‘1’ in unary decoding, which corresponds to the code termination mentioned in the following sections.

Here, we call the integer conversion, the consecutive processes of unary coding, bit inversion, and unary decoding, as the unary-domain bit inversion, which can be written as a function $F_{\text{UBI}}[\cdot]$ uniquely converting an integer sequence $\mathbf{x} = \underline{0^{N_{\text{start}}} x_1 0^{N_1} x_2 0^{N_2} \cdots 0^{N_{m-1}} x_m 0^{N_m}}$ as

$$F_{\text{UBI}}\left[\mathbf{x}\right] = \underline{N_{\text{start}} 0^{x_1-1} (N_1+1) 0^{x_2-1} (N_2+1) \cdots 0^{x_m-1} N_m}, \tag{5.5}$$

where $\{x_1, x_2, \cdots, x_m\}$ and $\{N_{\text{start}}, N_1, N_2, \cdots, N_m\}$ are respectively positive and non-negative integers for a non-negative integer m, and $\underline{0^N}$ is for a run of $\underline{0}$s with length $N(\geq 0)$. Even if $\theta < 0.5$ as in the above case $\theta = 2^{-(K+1)} (K > 0)$, GR code can perform as well as in the case of $\theta > 0.5$, as in case of $\theta = 2^{-1/2^R} (R > 0)$, by converting the input source symbols before encoding.

This proposed shaping by the unary-domain bit inversion, followed by GR encoding of Rice parameter $R = K$, gives us an approximately optimal code for the sparse sources mentioned above because $2^{-(K+1)} \simeq 1 - 2^{-1/2^R}$ at large K. When the inputs are written in the binary form of Bernoulli trials, the code become identical to the code presented in [86], called the elementary Golomb code: It is one of the run-length codes defined over the binary inputs with some coding parameter L of a positive integer; it gives the codeword '0' for the L-length runs of zeros in inputs and the codeword '1' followed by a phased-in binary code representing the length of the zero runs shorter than L, constructing a Huffman code for a Tunstall extension [87]. In other words, the encoding with the unary-domain bit inversion is an extended usage of the elementary Golomb code aimed at encoding general integer inputs.

The derived code, extended-domain GR (XDGR) code, cannot be described by a simple code tree usually used in Huffman coding. Instead, the almost instantaneous FV (AIFV) code scheme [88–90] allows us to represent its simple structure, discuss its theoretical optimality, and construct its coding algorithm as an extension of the conventional GR coding.

AIFV codes have multiple code trees having incomplete nodes, nodes still capable of increasing children, and source symbols are assigned to their incomplete nodes as well as their leaves. They guarantee their decodability by switching the trees to encode source symbols, under the switching rules, after encoding at the incomplete nodes. In other words, recursively integrating simple multiple code trees gives us a single complex code tree for efficiently encoding the Cartesian products of source symbols. One of the main advantages of these code trees is that we can assign source symbols to their roots, realizing codewords with zero code length.

The following sections first present, using the multiple code trees, the code gained by the combination of the proposed bit inverting and the conventional GR codes. Then, for simplicity, we show its asymptotic optimality of the code length based on the small-scaled Laplacian distributions $p_1[X|(K+1)^{-1}]$ when the K is large enough. Afterward, to discuss more precise efficiency of the proposed XDGR code, we analyze the average code length without the approximation of $2^{-(K+1)} \simeq 1 - 2^{-1/2^R}$ and prove some symmetry between the conventional GR code. Finally, further extensions are shown, based on Golomb code, a more general class of structured code, including GR code.

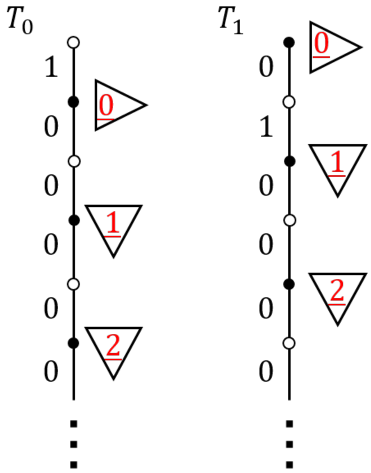
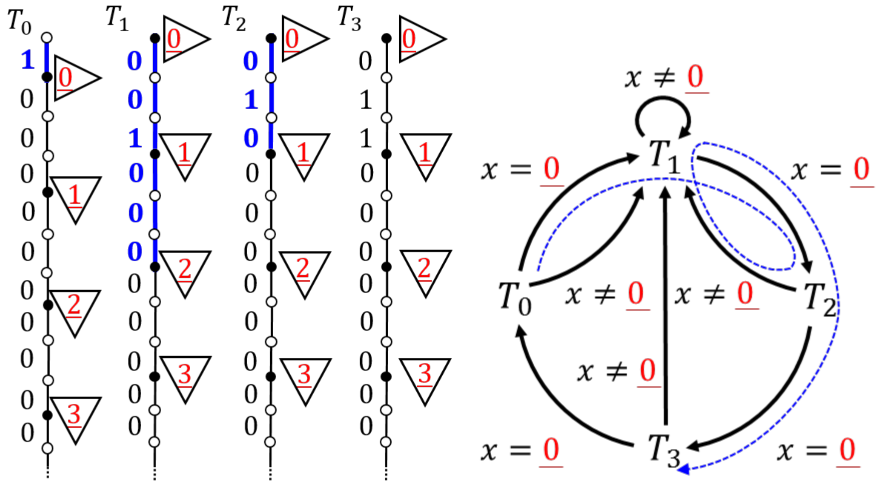


Fig. 5.1. Double code trees of the proposed XDGR code at $K = 1$. Binary located next to the edges and triangled numbers located next to the nodes respectively indicates the codewords and the assigned source symbols. Right-pointing and down-pointing triangles respectively show the switching rules of the code trees $T_0 \to T_1 \to T_0$ and $T_0 \to T_1 \to T_1$. Incomplete nodes with and without symbols assigned are depicted as black and white dots, respectively.

5.2.2 Multiple-code-tree representation

For simplicity, we first explain the code in the case of $K = 1$. From now on, we will underline the input decimal integers to distinguish from binary codewords. It can be said from Eq. (5.4) that the optimal code for $p_1[X|2^{-1}]$ assigns double code length compared to unary in case $X \neq 0$ to assign less code length to $X = 0$. The proposed XDGR code of $K = 1$ can be written in double code trees, as in Fig. 5.1: They are constructed only by incomplete nodes with source symbols assigned in ascending order to every two nodes, to achieve lower average code length for $X = 0$. The code tree T_0 has a code symbol '1' only in the first edge, and T_1 has it only in the second one. Note that this code corresponds to 3-ary AIFV in [88] with code alphabet $\mathcal{A} = \{1, 00, 01\}$, with its code-tree switching rules modified within a decodable range: Cyclic switching as $T_0 \to T_1 \to T_0$ when $X = 0$ and one directional switching to T_1 otherwise. The first code symbol '1' in the total encoding can be omitted because it is trivial.

Before we go to the generalized case of K, we show an example of the coding processes



(a) Code trees. (b) Switching of the code trees.

Fig. 5.2. Example of encoding input source symbol sequence $\mathbf{X} = \underline{0012000}$ using the multiple-code-tree representation of XDGR code with $K = 2$. The blue curled arrow in (b) traces the transitions of the code trees.

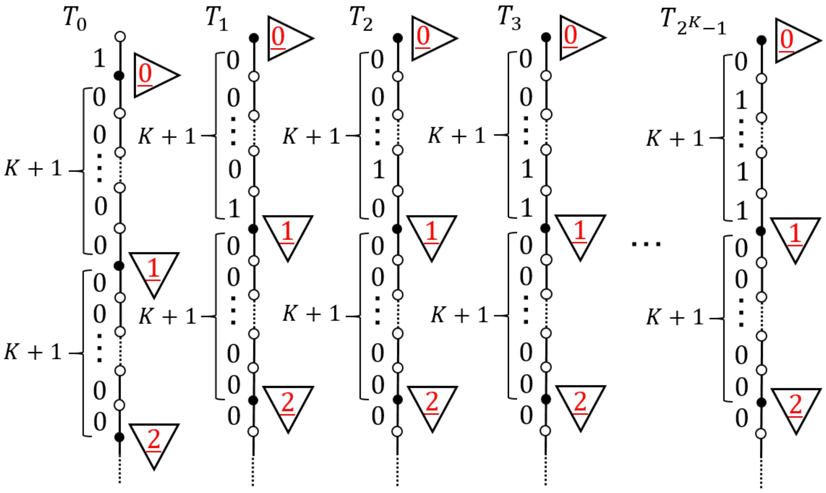
of the code. Say the input source symbols are $\mathbf{X} = \underline{00100}$, each in decimal, the first source symbol $\underline{0}$ is encoded by T_0: Codeword is '1', and the code tree is switched to T_1. Then, the second source symbol $\underline{0}$ can be encoded by a null codeword, and the code tree is switched to T_0. The third source symbol $\underline{1}$ is encoded by T_0 as '100' and the code tree turns to T_1. The code tree switches to T_0 again with a null codeword for the fourth source symbol $\underline{0}$. Finally, the fifth source symbol $\underline{0}$ is encoded by T_0 as '1', resulting in the total codeword sequence '1 (NULL) 100 (NULL) 1', where '(NULL)' indicates a codeword with zero code length. The first '1' is trivial so that the encoding result becomes '1001' with 4 bits of code length, shorter than the unary representation '110111' and the run-length GR code proposed in [84] '01000' with 1-bit run length and Rice parameter 0.

When decoding '1001', we first add the first trivial '1' to it as '11001'. Then, read the code from the beginning until it cannot be decoded by T_0: '11' is not decodable by T_0 so that the first source symbol must be $\underline{0}$, and the codeword '1001' should be decoded by T_1. The second source symbol must be $\underline{0}$ because T_1 has no '1' in the root, and the codeword '1001' should be decoded by T_0. T_0 can decode '100' but cannot decode '1001', which indicates the third source symbol is $\underline{1}$. Similarly, the rest source symbol '00' can be decoded, resulting in decoded $\mathbf{X} = \underline{00100}$.

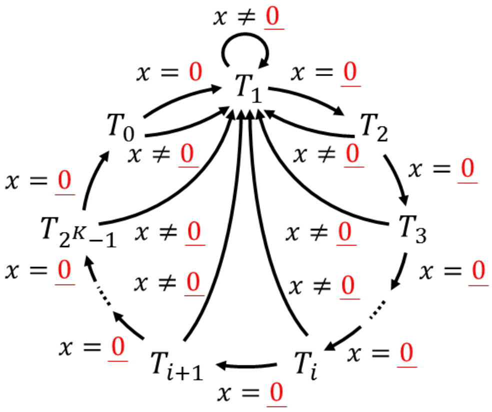
For another example, think of encoding the input decimal source symbols $\mathbf{X} = \underline{0012000}$ with $K = 2$, using four code trees in Fig. 5.2 (a) starting from T_0 of which switching rules described in Fig. 5.2 (b). As highlighted in the figures, each source symbol is respectively encoded by the code trees T_0, T_1, T_2, T_1, T_1, T_2, and T_3 into the codewords '1', '(NULL)', '010', '001000', '(NULL)', '(NULL)', and '(NULL)'. Omitting the trivial codeword '1' in the beginning and terminating the codeword by '1', the output codeword will be '(1) (NULL) 010 001000 (NULL) (NULL) (NULL) 1', i.e., '0100010001'.

Although we simplified the coding processes in the above example, it has to be noted that this code requires, to guarantee the decodability, at least the length of the input source-symbol sequence to be shared between the encoder and the decoder, as the arithmetic coding does. Since the source symbol $\underline{0}$ is assigned to the roots of the tree, of which codeword has zero code length, the termination of the encoding and decoding should be slightly modified under the coding conditions. If both the length of the input source-symbol sequence and of the codeword sequence are shared, the encoder can terminate the process after encoding the last source symbol even if its code length is zero. In this case, the decoder decodes the codeword sequence as above, resulting in less output source symbols than expected. Therefore, the source symbol $\underline{0}$ should be added to the output source symbols to complete the decoding. In cases where the decoder knows only the length of the input source-symbol sequence, the encoder should terminate the codeword sequence by the codeword '1' when the last codeword symbol has zero length. This terminating codeword '1' helps the decoder to realize the end of the codeword sequence by checking whether the length of the output source-symbol sequence exceeds the expectation. After the termination of the decoding, the exceeded output source symbols can be omitted to obtain a complete sequence.

It can be expected from Eq. (5.4) that for general non-negative integers K, it is preferable to make the average code length for zero even shorter by assigning source symbols to every $(K+1)$ nodes. As expected, the proposed XDGR code for K takes 2^K code trees as in Fig. 5.3: T_0 with '1' only in the first edge and T_1 to T_{2^K-1} with first $(K+1)$ edges that have '0' in the first one and at least one edge with '1'. This code corresponds to (2^K+1)-ary AIFV in [88] with code alphabet $\mathcal{A} = \{1, 00\cdots00, 00\cdots01, \cdots, 01\cdots11\}$, and thus its decoding delay is at most $(K+1)$ bits. The switching rules are the same as



(a) Code trees.



(b) Switching rules of the code trees.

Fig. 5.3. Multiple-code-tree representation of XDGR code with a non-negative integer K. Triangled and underlined red numbers located next to the nodes in (a) indicate the assigned source symbols. Right-pointing and down-pointing triangles respectively show the switching rules of the code trees $T_0 \to T_1 \to \cdots \to T_{2^K-1} \to T_0$ and $T_i \to T_1$ for every $0 \leq i < 2^K$, corresponding to (b).

the double code trees: Cyclic switching as $T_0 \to T_1 \to \cdots \to T_{2^K-1} \to T_0$ when $X = \underline{0}$ and one directional switching to T_1 otherwise.

In the sense of decodability, the first $(K+1)$ edges of the code trees T_1 to T_{2^K-1} can be designed arbitrarily as long as the first $(K+1)$ edges differ to each other and have '0'

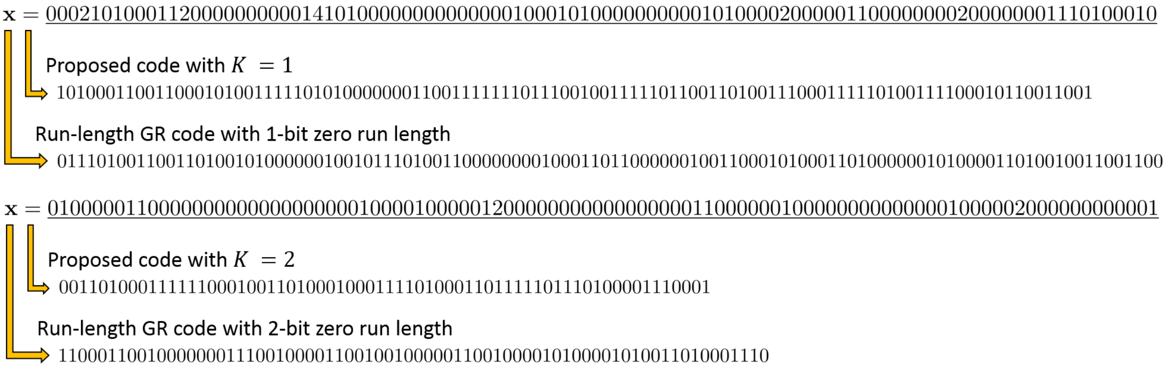


Fig. 5.4. Examples of the proposed code for 100 symbols at $K = 1$ and $K = 2$. Input X contains non-negative decimal integers. Each run length is set to minimize the run-length GR code. The proposed code terminates its total codeword sequence by '1' in case decoder do not know the total code length to decode.

in the first edge. However, by sorting the order of T_1 to T_{2^K-1} for the first $(K+1)$ edges to be in ascending order of $(K+1)$-bit binary numbers, these edges can be interpreted as run-length codes and make the coding algorithm simple. In addition, the code becomes unary, the GR code at $R = 0$, at $K = 0$ if we use $T_1 = T_0$.

The coding algorithm can be reorganized as an extension of unary coding as in Algorithms 1 and 2. These algorithms assume that only the parameter K and the length N of the input source-symbol sequence are shared between the encoder and the decoder. The encoder outputs codewords in $(K + 1)$-bit units following the values of source symbols. On the other hand, the decoder reads $(K+1)$-bit units of '0's until any '1' appears in the $(K+1)$ bits, similar to the unary decoder reading '0's until the stop bit '1' appears. The variable T, the number of current code tree, indicates the number of the source symbols appeared before the next non-zero source symbol.

The proposed XDGR code has a very similar structure to the run-length GR code in [84] except in the assignment of the source symbols after '1'. Fig. 5.4 shows some examples. One may think it is redundant to use many '0's between the symbols. However, assigning source symbols to every $(K + 1)$ nodes enables us to make the first $(K + 1)$ bit of each tree serve both as the run length and the stop bit, relatively shortening the code length for smaller-valued symbols.
]

Algorithm 1 Encode $\mathbf{X} = X_0 X_1 \cdots X_{N-1}$ by XDGR code

```
Set $n = 0$, $T = 0$.
while $n < N$ do
  if $X_n = \underline{0}$ then
    Increment $T$ by 1 and replace it with its modulo $2^K$.
  else
    Output $(K + 1)$-bit binary representing $T$.
    Output $(K + 1)(X_n - 1)$ ‘0’s.
    Set $T = 1$.
  end if
  if $T = 0$ then
    Output ‘1’.
  end if
  Increment $n$ by 1.
end while
if $T \neq 0$ then
  Output ‘1’.
end if
```

5.2.3 Average code length

Average code length of the proposed XDGR code for encoding source symbol X belonging to $p_1[X|(K+1)^{-1}]$ is given by the expectation of code length based on the stationary probabilities of the code trees. From the above code-tree switching rules, the transition probability matrix H of the code trees can be written as

$$H = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 & p_z \\ 1 & 1-p_z & 1-p_z & \cdots & 1-p_z & 1-p_z \\ 0 & p_z & 0 & 0 & \cdots & 0 \\ 0 & 0 & p_z & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ 0 & \cdots & & 0 & p_z & 0 \end{bmatrix} \tag{5.6}$$

Algorithm 2 Decode N source symbols from a binary codeword sequence by XDGR code

```
Set $n = 0$, $T = 0$.
while $n < N$ do
    Set $l = 0$.
    while the first $(K + 1)$ bits of the codeword sequence is all ‘0’ do
        Read out $(K + 1)$ bits from the codeword sequence.
        Increment $l$ by 1.
    end while
    if $T = 0$ then
        Output $\underline{l}$.
    else
        Output $\underline{(l + 1)}$.
    end if
    if the first bit of the codeword sequence is ‘1’ then
        Read out 1 bit from the codeword sequence.
        Output $(2^K - 1)$ $\underline{0}$s.
        Set $T = 0$.
        Increment $n$ by $2^K$.
    else
        Read out $(K + 1)$ bits from the codeword sequence.
        Replace $T$ by the read $(K + 1)$-bit binary number.
        Output $(T - 1)$ $\underline{0}$s.
        Increment $n$ by $T$.
    end if
end while
Omit the exceeded symbols after the $N$-th output symbol.
```

where $p_z = p_1[0|(K+1)^{-1}]$. Let p_i^* $(0 \leq i < 2^K)$ be the stationary probability of code tree T_i. The stationary probabilities $\mathbf{p}^* = [p_0^*, p_1^*, \cdots, p_{2^K-1}^*]^\mathsf{T}$ should satisfy

$$\mathbf{p}^* = H\mathbf{p}^* \quad \text{and} \quad \mathbf{1}^\mathsf{T}\mathbf{p}^* = 1 \tag{5.7}$$

where $\mathbf{1} = [1, 1, \cdots, 1]^{\mathsf{T}}$, which leads to

$$p_i^* = \begin{cases} \dfrac{p_z^{2^K-1}}{\sum_{j=0}^{2^K-1} p_z^j} & (\text{if } \ i = 0) \\ \\ \dfrac{p_z^{i-1}}{\sum_{j=0}^{2^K-1} p_z^j} & (\text{otherwise}) \end{cases}. \tag{5.8}$$

Taking into account that the code length $\mathcal{L}_{T_i}[X|K]$ for source symbol X at code tree T_i is written as

$$\mathcal{L}_{T_i}[X|K] = \begin{cases} (K+1)X + 1 & (\text{if } \ i = 0) \\ (K+1)X & (\text{otherwise}) \end{cases}, \tag{5.9}$$

the average code length $\mathcal{L}_1^{\mathrm{XR}}[X|K]$ of the proposed XDGR code for source symbol X is given as

$$\begin{aligned} \mathcal{L}_1^{\mathrm{XR}}[X|K] &= \sum_{i=0}^{2^K-1} \mathcal{L}_{T_i}[X|K] p_i^* \\ &= (K+1)X + \frac{p_z^{2^K-1}}{\sum_{j=0}^{2^K-1} p_z^j}. \end{aligned} \tag{5.10}$$

To show the similarity of the optimal code length $\mathcal{L}_1^{\mathrm{ideal}}[X|K]$ and $\mathcal{L}_1^{\mathrm{XR}}[X|K]$, we further develop the equations. Since

$$Z_1\left((K+1)^{-1}\right) = \sum_{X=0}^{\infty} 2^{-(K+1)X} = \frac{1}{1 - 2^{-(K+1)}}, \tag{5.11}$$

$\mathcal{L}_1^{\mathrm{ideal}}[X|K]$ can be rewritten as

$$\mathcal{L}_1^{\mathrm{ideal}}[X|K] = (K+1)X + \log_2 \frac{1}{1 - 2^{-(K+1)}} \tag{5.12}$$

and $\mathcal{L}_1^{\mathrm{XR}}[X|K]$ as

$$\begin{aligned} \mathcal{L}_1^{\mathrm{XR}}[X|K] &= (K+1)X + \frac{p_z^{2^K-1}}{1 - p_z^{2^K}}(1 - p_z) \\ &= (K+1)X + \frac{(1 - 2^{-(K+1)})^{2^K-1}}{1 - (1 - 2^{-(K+1)})^{2^K}} 2^{-(K+1)} \end{aligned} \tag{5.13}$$

from $p_z = Z_{(K+1)^{-1}}^{-1} = (1 - 2^{-(K+1)})$. In the extreme case of $K \gg 1$, the second terms in Eqs. (5.12) and (5.13) respectively become

$$\log_2 \frac{1}{1 - 2^{-(K+1)}} \simeq \frac{1}{\ln 2} 2^{-(K+1)} \simeq 2^{-(K+1)} \times 1.4427 \tag{5.14}$$

and

$$\frac{(1 - 2^{-(K+1)})^{2^K-1}}{1 - (1 - 2^{-(K+1)})^{2^K}} 2^{-(K+1)} \simeq \frac{1}{\sqrt{e} - 1} 2^{-(K+1)} \simeq 2^{-(K+1)} \times 1.5415 \tag{5.15}$$

because

$$\lim_{K\to\infty}(1-2^{-(K+1)})^{2^K}=\frac{1}{\sqrt{e}}. \tag{5.16}$$

Therefore, if $K \gg 1$, namely the entropy of the source symbols is significantly low, the difference of the proposed XDGR code length from the optimal code length will only depends on $2^{-(K+1)}$ as

$$\mathcal{L}_1^{\mathrm{XR}}[X|K]-\mathcal{L}_1^{\mathrm{ideal}}[X|K]\simeq 2^{-(K+1)}\times 0.0988, \tag{5.17}$$

which proves the high performance of the proposed XDGR code.

It should also be noted that $\mathcal{L}_1^{\mathrm{XR}}[X|0]=X+1$ coincides with the GR code of $R=0$, and the parameter K corresponds to the "negative domain" of Rice parameters.

5.2.4 Asymptotic optimality

The main concern here is whether the difference of the code length shown above becomes negligible compared to the total code length. The entropy S_K of $p_1[X|(K+1)^{-1}]$ is calculated as

$$\begin{aligned}S_K&=\sum_{X=0}^{\infty}\mathcal{L}_1^{\mathrm{ideal}}[X|K]p_1[X|(K+1)^{-1}]\\&=\frac{(K+1)}{Z_1\left((K+1)^{-1}\right)}\sum_{X=0}^{\infty}X2^{-(K+1)X}+\log_2 Z_1\left((K+1)^{-1}\right)\\&=\frac{(K+1)2^{-(K+1)}}{1-2^{-(K+1)}}+\log_2\frac{1}{1-2^{-(K+1)}}.\end{aligned} \tag{5.18}$$

On the other hand, the total average code length $\bar{S}_K$ of the proposed XDGR code, coding the source symbols belonging to $p_1[X|(K+1)^{-1}]$, is given by

$$\begin{aligned}\bar{S}_K&=\sum_{X=0}^{\infty}\mathcal{L}_1^{\mathrm{XR}}[X|K]p_1[X|(K+1)^{-1}]\\&=\frac{(K+1)2^{-(K+1)}}{1-2^{-(K+1)}}+\frac{(1-2^{-(K+1)})^{2^K-1}}{1-(1-2^{-(K+1)})^{2^K}}2^{-(K+1)},\end{aligned}$$

and thus the coding loss D_K, defined here as a relative difference from the entropy, converges to

$$\begin{aligned}\lim_{K\to\infty} D_K &= \lim_{K\to\infty} \frac{\bar{S}_K - S_K}{S_K} \\ &= \lim_{K\to\infty} \frac{\frac{1}{\sqrt{e}-1}2^{-(K+1)} - \frac{1}{\ln 2}2^{-(K+1)}}{\frac{(K+1)2^{-(K+1)}}{1-2^{-(K+1)}} + \frac{1}{\ln 2}2^{-(K+1)}} \\ &= \lim_{K\to\infty} \frac{\frac{1}{\sqrt{e}-1} - \frac{1}{\ln 2}}{\frac{(K+1)}{1-2^{-(K+1)}} + \frac{1}{\ln 2}} \\ &= 0. \end{aligned} \tag{5.19}$$

So, if the scale ϕ of the distribution in Eq. (5.2) is small enough, or the optimal parameter K is large enough, the coding loss of the proposed XDGR code can be neglected compared to the total code length.

5.2.5 Parameter estimation for extended-domain Golomb–Rice code

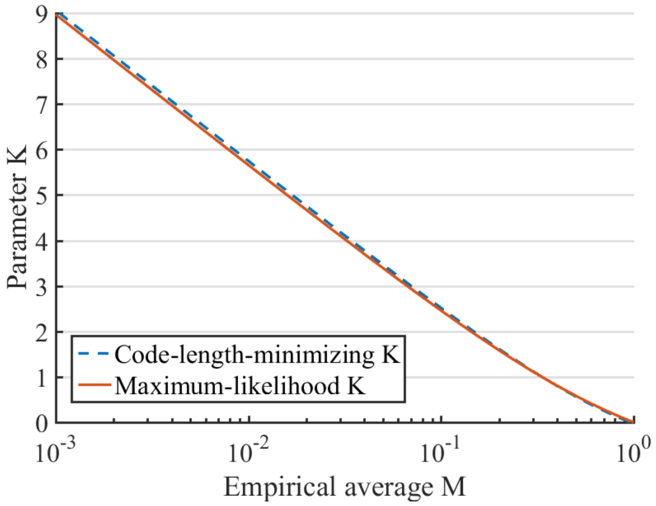
To make use of the proposed XDGR code for distributions of variable scale ϕ, in the similar way as using the conventional GR code, the parameter K introduced above, the parameter that decides how many code trees to use, should be estimated from an input source-symbol sequence. In the conventional GR coding, the Rice parameter R is often estimated to minimize the approximate total code length $B_{\mathrm{GR}}(r)$ of the input source-symbol sequence $\{X_i\}_{i=0}^{N-1}$:

$$B_{\mathrm{GR}}(r) = \sum_{i=0}^{N-1} \left(\frac{X_i}{2^R} + R + 1 \right), \tag{5.20}$$

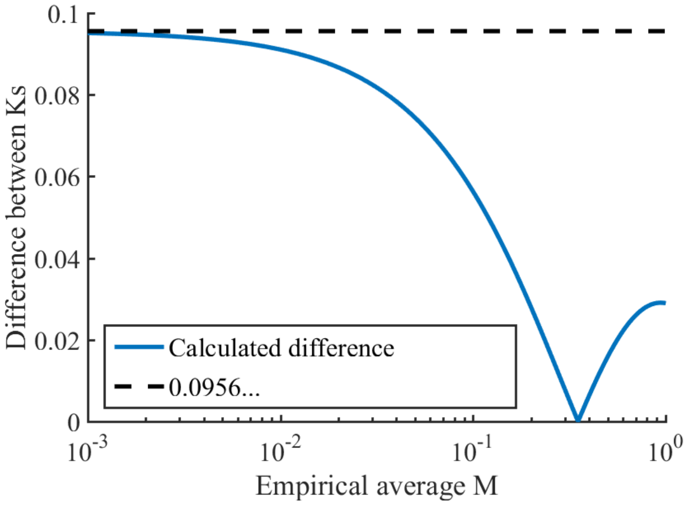
which can be minimized at its stationary point

$$r_{\mathrm{opt}} = \log_2 ((\ln 2)M) \tag{5.21}$$

where $M = \frac{1}{N}\sum_{i=0}^{N-1} X_i$ indicates the empirical average. This estimated parameter R_{opt} can be used after rounding for coding and approximately matches to the maximum-likelihood estimate of R for the source distribution $p_1[X|2^R]$. On the other hand, in the proposed XDGR coding, we cannot explicitly derive the direct estimate of K for minimizing the total code length. Therefore, we discuss here using the maximum-likelihood estimate of K for the source distribution $p_1[X|(K+1)^{-1}]$ and its correspondence to the code-length-minimizing K.



(a) K estimated in case of each empirical average M.



(b) Absolute difference between estimated Ks.

Fig. 5.5. Numerical comparison of parameter K estimations.

The log-likelihood of $\{X_i\}_{i=0}^{N-1}$ for the source distribution $p_1[X|(K+1)^{-1}]$ is written, using Eq. (5.11), as

$$\begin{aligned} B_{\mathrm{LH}}(K) &= \sum_{i=0}^{N-1} \log_2 p_1[X_i|(K+1)^{-1}] \\ &= \sum_{i=0}^{N-1} \left(-(K+1)X_i - \log_2 Z_1\left((K+1)^{-1}\right)\right) \\ &= -\sum_{i=0}^{N-1}(K+1)X_i + N\log_2\left(1-2^{-(K+1)}\right), \end{aligned} \tag{5.22}$$

which can be maximized at its stationary point

$$K_{\mathrm{est}} = \log_2\left(1+\frac{1}{M}\right) - 1. \tag{5.23}$$

When the code-length-minimizing $K = K_{\mathrm{opt}}$ satisfies $K_{\mathrm{opt}} \gg 1$, the empirical average M must be small enough, and thus the maximum-likelihood estimate K_{est} approximately values as

$$K_{\mathrm{est}} \simeq \log_2\left(\frac{1}{M}\right) - 1 = -\log_2 M - 1 \tag{5.24}$$

from $M^{-1} \gg 1$. In case $K \gg 1$, K_{opt} can be explicitly derived from the stationary point of the total code length

$$\begin{aligned} B_{\mathrm{Prop}}(K) &= \sum_{i=0}^{N-1} \mathcal{L}_1^{\mathrm{XR}}[X_i|K] \\ &\simeq \sum_{i=0}^{N-1}(K+1)X_i + \frac{N}{\sqrt{e}-1}2^{-(K+1)} \end{aligned} \tag{5.25}$$

as

$$K_{\mathrm{opt}} \simeq -\log_2 M - 1 + \log_2\frac{\ln 2}{\sqrt{e}-1} \simeq -\log_2 M - 1 + 0.0956, \tag{5.26}$$

matching to K_{est} when rounded. That means the maximum-likelihood estimate of K by Eq. (5.23) gives a nearly optimal parameter for code length when $K \gg 1$.

Fig. 5.5 shows the numerical comparison between the maximum-likelihood estimate K_{est} and the actual code-length-minimizing estimate K_{opt} for each empirical average M: the former comes from Eq. (5.23) and the latter from the relation derived from the stationary point of the code length $B_{\mathrm{Prop}}(K)$, without assuming $K \gg 1$, as

$$M = -\frac{\mathrm{d}}{\mathrm{d}K}\frac{(1-2^{-(K+1)})^{2^K-1}}{1-(1-2^{-(K+1)})^{2^K}}2^{-(K+1)}. \tag{5.27}$$

Both estimation slightly differ to each other at any $K \geq 0$ with their difference converging to $\log_2\frac{\ln 2}{\sqrt{e}-1} \simeq 0.0956$ as expected.

Following the above discussions, Eq. (5.23) gives K that nearly minimizes the total code length of the proposed XDGR code. Furthermore, this proposed XDGR extension can be interpreted as extending Rice parameters to the negative domain. The proposed XDGR code is expected to be used with the conventional GR code as unified GR code by estimating the parameters R and K based on Eq. (5.23) or the combination of Eqs. (5.21) and (5.23).

5.2.6 Symmetry of relative redundancy to Golomb–Rice code

Here, for more precise investigation, we compare the relative redundancy of the proposed XDGR and the conventional GR codes in encoding the geometric source $p_{\mathrm{Geo}}[X|\theta]$. The average code length $\bar{\mathcal{L}}_1^{\mathrm{XR}}(\theta|K)$ of the proposed XDGR code with the parameter K is given from the stationary probability p_i^* ($i = 0, 1, \cdots, 2^K - 1, \sum_i p_i^* = 1$) of each code tree T_i as

$$\begin{aligned}\bar{\mathcal{L}}_1^{\mathrm{XR}}(\theta|K) &= p_0^* \sum_{X=0}^{\infty} p_{\mathrm{Geo}}[X|\theta][(K+1)X+1] + \sum_{i=1}^{2^K-1} p_i^* \sum_{X=1}^{\infty} p_{\mathrm{Geo}}[X|\theta](K+1)X \\ &= (K+1)\sum_{X=0}^{\infty} p_{\mathrm{Geo}}[X|\theta]X + p_0^*.\end{aligned} \tag{5.28}$$

Similar to the above discussions, the stationary probability p_0^* is derived as

$$p_0^* = \frac{(1-\theta)^{2^K-1}}{\sum_{j=0}^{2^K-1}(1-\theta)^j} = \frac{\theta(1-\theta)^{2^K-1}}{1-(1-\theta)^{2^K}}, \tag{5.29}$$

which makes

$$\bar{\mathcal{L}}_1^{\mathrm{XR}}(\theta|K) = \frac{(K+1)\theta}{1-\theta} + \frac{\theta(1-\theta)^{2^K-1}}{1-(1-\theta)^{2^K}}. \tag{5.30}$$

For conventional GR code, its average code length $\bar{\mathcal{L}}_1^{\mathrm{R}}(\theta|R)$ with Rice parameter R is written as

$$\begin{aligned}\bar{\mathcal{L}}_1^{\mathrm{R}}(\theta|R) &= \sum_{X=0}^{\infty} p_{\mathrm{Geo}}[X|\theta]\left(R+1+\lfloor 2^{-R}X \rfloor\right) \\ &= R+1+\sum_{X=0}^{\infty} p_{\mathrm{Geo}}[X|\theta]2^{-R}X - \sum_{i=1}^{2^R-1}\sum_{X=0}^{\infty} p_{\mathrm{Geo}}[2^R X + i|\theta]2^{-R}i \\ &= R+1+\frac{2^{-R}\theta}{1-\theta} - \left(\frac{2^{-R}\theta}{1-\theta} - \frac{\theta^{2^R}}{1-\theta^{2^R}}\right) \\ &= R+1+\frac{\theta^{2^R}}{1-\theta^{2^R}}\end{aligned} \tag{5.31}$$

where $\lfloor\cdot\rfloor$ is a flooring operation.

Therefore, the entropy of the source

$$\begin{aligned}\bar{\mathcal{L}}_1^{\text{ideal}}(\theta) &= \sum_{X=0}^{\infty} -p_{\text{Geo}}[X|\theta]\log_2 p_{\text{Geo}}[X|\theta] \\ &= (1-\theta)^{-1}[-\theta\log_2\theta - (1-\theta)\log_2(1-\theta)]\end{aligned} \tag{5.32}$$

leads to the respective relative redundancy $\mathcal{R}_1^{\text{XR}}(\theta|K)$ and $\mathcal{R}_1^{\text{R}}(\theta|R)$:

$$\begin{aligned}\mathcal{R}_1^{\text{XR}}(\theta|K) &= \frac{\bar{\mathcal{L}}_1^{\text{XR}}(\theta|K)}{\bar{\mathcal{L}}_1^{\text{ideal}}(\theta)} - 1 \\ &= \left[(K+1)\theta + \frac{\theta(1-\theta)^{2^K}}{1-(1-\theta)^{2^K}}\right] \cdot [-\theta\log_2\theta - (1-\theta)\log_2(1-\theta)]^{-1} - 1\end{aligned} \tag{5.33}$$

$$\begin{aligned}\mathcal{R}_1^{\text{R}}(\theta|R) &= \frac{\bar{\mathcal{L}}_1^{\text{R}}(\theta|R)}{\bar{\mathcal{L}}_1^{\text{ideal}}(\theta)} - 1 \\ &= \left[(R+1)(1-\theta) + \frac{(1-\theta)\theta^{2^R}}{1-\theta^{2^R}}\right] \cdot [-\theta\log_2\theta - (1-\theta)\log_2(1-\theta)]^{-1} - 1,\end{aligned} \tag{5.34}$$

which proves the symmetry between the proposed XDGR and the conventional GR codes as

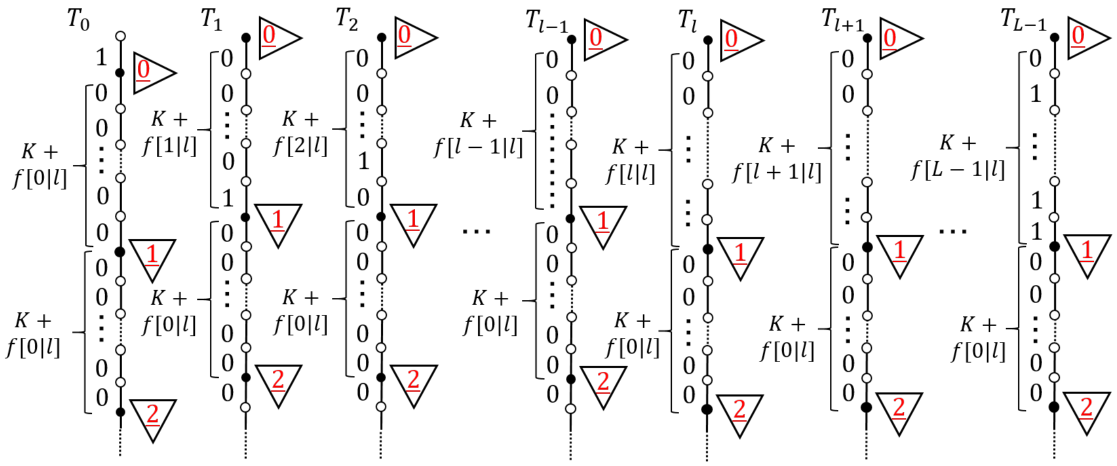
$$\mathcal{R}_1^{\text{XR}}(1-\theta|K) = \mathcal{R}_1^{\text{R}}(\theta|R) \text{ when } K = R. \tag{5.35}$$

The above proof of the symmetry guarantees the codes show symmetric relative redundancy to each other when encoding geometric sources with corresponding coding parameters K and R. Therefore, the optimality of the proposed XDGR is approximated as precise as that of the conventional GR code.

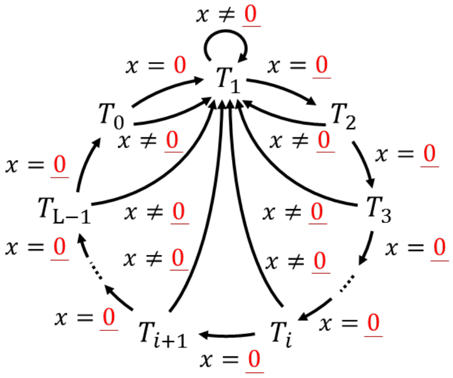
5.2.7 Further code extension

The discussion above has shown that the proposed shaping by the unary-domain bit inversion realizes a code having a strong symmetry between GR codes in the sense of relative redundancy, especially between their parameters K and R. Now, there is a possibility of combining the proposed shaping with Golomb code, which enables us a more precise tune to the scale ϕ of the input sources using the Golomb parameter S. This combination, extended-domain Golomb (XDG) code, can also be interpreted by multiple code trees generalized from the ones mentioned in the previous sections.

By using multiple code trees and a conditional branch function $f[i|l]$ which outputs 0 if $i < l$ and 1 otherwise, the proposed code can be represented as in Fig. 5.6: It is



(a) Code trees.



(b) Switching rules of the code trees.

Fig. 5.6. Multiple-code-tree representation of the proposed XDG code with a positive integer L as an extension of the code in Fig. 5.3. L is written with non-negative integers K and $l\,(< 2^{K-1})$ as $L = 2^K - l$. $f[i|l]$ is a conditional branch function that outputs 0 if $i < l$ and 1 otherwise.

constructed with L code trees, which naturally come from the phased-in binary code included in the conventional Golomb code. When the parameter L is a power of two, in other words, $l = 0$, the proposed code becomes identical to the conventional XDGR code. The encoding and decoding steps are shown in Algorithms 3 and 4, which is an extension of the unary code as well as the conventional XDGR code.

Fig. 5.7 shows the encoding example of the proposed XDG code compared to the XDGR

Algorithm 3 Encode $\mathbf{X} = X_0X_1\cdots X_{N-1}$

```
Set $n = 0$, $T = 0$.
while $n < N$ do
  if $X_n = \underline{0}$ then
    Increment $T$ by 1 and replace it with its modulo $L$.
  else
    Output '0' and phased-in binary code of parameter $L$ representing $T$.
    Output $(K + f[0|l])(X_n - 1)$ '0's.
    Set $T = 1$.
  end if
  if $T = 0$ then
    Output '1'.
  end if
  Increment $n$ by 1.
end while
if $T \neq 0$ then
  Output '1'.
end if
```

code. Since the proposed code has a finer parameter to tune, there are some cases where it can achieve a shorter code length than the XDGR code is achievable. Indeed, finer tunable parameters mean, if we want to switch them sequence by sequence, more bits to represent themselves. However, they make the coding design more flexible, and we can control the costs for sending the parameter Ls by limiting the range of them or dealing with longer sequences. The following sections show the proof that the proposed XDG code is identical to the conventional Golomb code with the unary-domain bit inversion, which we here call the inverted Golomb code, for convenience, and their symmetry in the sense of relative redundancy.

Algorithm 4 Decode N source symbols from a binary codeword sequence

```
Set $n = 0$, $T = 0$.
while $n < N$ do
    Set $i = 0$.
    while the first $(K + f[0|l])$ bits of the codeword sequence is all ‘0’ do
        Read out $(K + f[0|l])$ bits from the codeword sequence.
        Increment $i$ by 1.
    end while
    if $T = 0$ then
        Output $\underline{i}$.
    else
        Output $\underline{(i + 1)}$.
    end if
    Read out 1 bit from the codeword sequence.
    if the read bit is ‘1’ then
        Output $(L - 1)$ $\underline{0}$s.
        Set $T = 0$.
        Increment $n$ by $L$.
    else
        Read out phased-in binary code of parameter $L$ from the codeword sequence.
        Replace $T$ by the number represented by the phased-in binary code.
        Output $(T - 1)$ $\underline{0}$s.
        Increment $n$ by $T$.
    end if
end while
Omit the exceeded symbols after the $N$-th output symbol.
```

5.2.8 Equivalence to inverted Golomb code

Preparation of proof

To prove the equivalence between the inverted Golomb code $C_{\mathrm{iG}}[\mathbf{x}|L]$ for Golomb parameter $L = 2^K - l$ $(0 \leq l < 2^{K-1})$ and XDG code $C_{\mathrm{XDG}}[\mathbf{x}|L]$ for parameter L, we compare

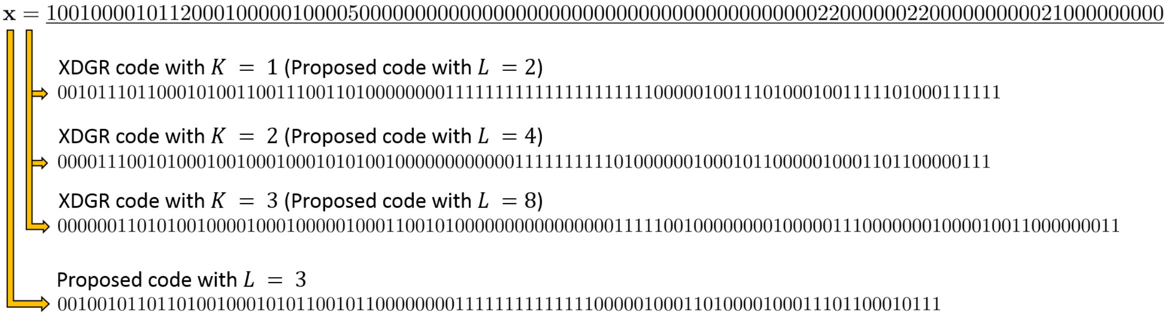


Fig. 5.7. Examples of XDGR code and the proposed code for 100 symbols with each parameter. Input X contains non-negative decimal integers. XDGR code cannot shorten the code than at $K = 2$ while the proposed code achieves the shortest code length by $L = 3$, with L being the extension of K.

the codewords contained in the code symbol sequence. We here use the input sequence $\mathbf{x} = \underline{0^{N_{\text{start}}} x_1 0^{N_1} x_2 0^{N_2} \cdots 0^{N_{m-1}} x_m 0^{N_m}}$, which represents any input sequences by setting non-negative integers $\{m, N_{\text{start}}, N_1, N_2, \cdots, N_m\}$ and positive integers $\{x_1, x_2, \cdots, x_m\}$. The cases are divided depending on the following conditions:

1. Either encoding from the beginning or the middle of $\mathbf{x}$;
2. whether encoding the last input symbol of $\mathbf{x}$.

The second condition matters when the decoder does not know the end of the codeword sequence. From a practical perspective, we set the main discussion on this case, where the decoder only knows the number of integers.

Before going on to the proof, we define some expressions: The decimal input integers and the binary codewords are respectively underlined and quoted as $\underline{x}$ and ‘0’; we refer to $\underline{0^x}$, ‘0^x’, and ‘1^x’ as the runs of $\underline{0}$s, ‘0’s, and ‘1’s with length x; the phased-in binary code for parameter L representing $\underline{x}$ is written as ‘$\mathrm{PBC}_L[x]$’; we use $x\%L \equiv x - \lfloor x/L \rfloor \times L$ for remainders where $\lfloor \cdot \rfloor$ is a flooring operation; ceiling operations are written using $\lceil \cdot \rceil$.

Using these expressions, Golomb code $C_{\mathrm{G}}[\underline{x}|L]$ for Golomb parameter L of $\underline{x}$ can be written as ‘$0^{\lfloor x/L \rfloor} 1 \mathrm{PBC}_L[x\%L]$’. However, to show the correspondence to the code shown in Fig. 5.6, we here use the unary part inverted as ‘$1^x 0$’, and thus ‘$1^{\lfloor x/L \rfloor} 0 \mathrm{PBC}_L[x\%L]$’ for $\underline{x}$, without loss of generality. It should also be noted that ‘$\mathrm{PBC}_L[0]$’ is equivalent to ‘$0^{K+f[0|l]-1}$’, and the first part of the codewords in each tree T_i ($i \neq 0$) in Fig. 5.6 (a) is given as ‘$0\mathrm{PBC}_L[i]$’.

Codewords in inverted Golomb code

As explained in the previous section, we defined the inverted Golomb code as a code given by the combination of the unary-domain bit inversion in Eq. (5.5) and Golomb encoding, namely, Golomb code for $F_{\mathrm{UBI}}[\mathbf{x}]$. Since Golomb code is uniquely decodable and the inversion $F_{\mathrm{UBI}}[\cdot]$ is invertible, the inverted Golomb code $C_{\mathrm{iG}}[\mathbf{x}|L] = C_{\mathrm{G}}[F_{\mathrm{UBI}}[\mathbf{x}]|L]$ is also uniquely decodable when the decoder knows both the end of the codeword sequence and the number of integers to decode. However, when the decoder does not know the end of the codeword sequence, it will not be uniquely decodable or will be redundant at the end of the sequence if we use the Golomb encoding output for $F_{\mathrm{UBI}}[\mathbf{x}]$ as is. Therefore, we have to think about termination in this case. The following shows the codeword sequence using a termination, which keeps the decodability and minimizes the redundancy.

In the case where the decoder only knows the number of integers to decode, the output of the unary-domain bit inversion should not end with runs of zeros. This is because the decoder cannot know the amount of m in the input $\mathbf{x}$, and $F_{\mathrm{UBI}}^{-1}[\cdot]$ will fail when the runs of zeros are followed by some other consecutive code beginning with zero. On the other hand, when $F_{\mathrm{UBI}}[\mathbf{x}]$ ends with a non-zero integer, the decoder can detect the end of the code by counting the output integers even if any other codes follow. Therefore, under the condition discussed here, the bit-inversion should use the function

$$\hat{F}_{\mathrm{UBI}}\left[\mathbf{x}\right] = \underline{N_{\mathrm{start}}0^{x_1-1}(N_1+1)0^{x_2-1}(N_2+1)\cdots 0^{x_m-1}(N_m+1)}. \tag{5.36}$$

Although it is not surjective since it ends with a non-zero integer, we here assume that the codes are all terminated and do not consider inverse converting $\hat{F}_{\mathrm{UBI}}[\mathbf{x}]$ ending with a zero.

i-a) Encoding $\underline{0^{N_{\mathrm{start}}}\cdots}$ $(m \neq 0)$

Let us think of the inverted Golomb encoding of $\underline{0^{N_{\mathrm{start}}}\cdots}$ followed by some positive integer. The unary-domain bit inversion of the concerning part is $\underline{N_{\mathrm{start}}}$. Therefore, the output of the inverted Golomb encoding is

$$\text{‘}1^{\lfloor N_{\mathrm{start}}/L \rfloor}0\mathrm{PBC}_L[N_{\mathrm{start}}\%L]\text{’}. \tag{5.37}$$

i-b) Encoding $\underline{0^{N_{\mathrm{start}}}}$ $(m = 0,\ N_{\mathrm{start}} \neq 0)$

Circumstances will change if no positive integer follows the input $\underline{0^{N_{\mathrm{start}}}}$. The Golomb code for $\hat{F}_{\text{UBI}}[\underline{0^{N_{\text{start}}}}]$ is the same as in Eq. (5.37). However, the decoder can recognize from the codeword '$1^{\lfloor N_{\text{start}}/L \rfloor}$' that there are $(\lfloor N_{\text{start}}/L \rfloor \times L)$ or more zeros to output. Awing to this fact, if the decoder knows the number of the integers to decode (here, N_{start}), the encoder needs only to output

$$\text{`}1^{\lceil N_{\text{start}}/L \rceil}\text{'} \tag{5.38}$$

to guarantee the decodability.

ii-a) Encoding $\underline{\cdots x_i 0^{N_i} \cdots}$ $(m \neq 0,\ 0 < i < m)$

Next, let us think of the inverted Golomb encoding of $\underline{\cdots x_i 0^{N_i} \cdots}$ $(1 < i < m)$ which follows some positive integer input. In this case, the inputs of the concerning part become $\underline{0^{x_i - 1}(N_i + 1)}$ by the unary-domain bit inversion, resulting in $C_{\text{iG}}[\mathbf{x}|L]$ as

$$\begin{aligned} &\text{`}0\text{PBC}_L[0]0\text{PBC}_L[0]\cdots 0\text{PBC}_L[0]1^{\lfloor (N_i+1)/L \rfloor}0\text{PBC}_L[(N_i+1)\%L]\text{'} \\ =\ &\text{`}0^{(K+f[0|l])(x_i-1)}1^{\lfloor (N_i+1)/L \rfloor}0\text{PBC}_L[(N_i+1)\%L]\text{'}. \end{aligned} \tag{5.39}$$

ii-b) Encoding $\underline{\cdots x_m 0^{N_m}}$ $(m \neq 0)$

The Golomb code for $\hat{F}_{\text{UBI}}[\underline{\cdots x_m 0^{N_m}}]$ is the same as in Eq. (5.39) at $i = m$. To discuss the termination for this part, we have to consider both cases where the input sequence ends with non-zero (in case of $N_m = 0$) and zero (in case of $N_m \neq 0$).

When $N_m = 0$, the codeword $0\text{PBC}_L[(N_m+1)\%L]$ in the tail of Eq. (5.39) is redundant for the same reason mention in i-b). Therefore,

$$\text{`}0^{(K+f[0|l])(x_m-1)}1^{\lceil (N_m+1)/L \rceil}\text{'} \tag{5.40}$$

is enough for $C_{\text{iG}}[\mathbf{x}|L]$ to keep the decodability.

When $N_m \neq 0$, the input converted by the unary-domain bit inversion is $\hat{F}_{\text{UBI}}[\underline{\cdots x_m}] = \underline{\cdots 0^{x_m - 1}}$, of which corresponding Golomb code is '$0^{(K+f[0|l])(x_m-1)}$'. Leaving the end with '0', in this case, may cause a decoding error when the decoder does not know the end of the codeword sequence: If there comes another '$0^{(K+f[0|l])}$' after the codeword sequence, the decoder will output $\underline{(x_m + 1)}$ as the last integer. Therefore, the encoder has to terminate the codeword with minimum bits. If the decoder knows the number of the inputs, it is reasonable for the encoder to terminate the codeword sequence with '1': The decoder can determine x_m when it reads the last '1' and recognize it has finished reading the integers by counting decoded ones. This results in $C_{\mathrm{iG}}[\mathbf{x}|L]$ identical to Eq. (5.40), which is uniquely decodable.

Codewords in XDG code

The following shows the codeword sequence of the proposed XDG code, with the cases divided the same as above. This code also needs termination, and we show it is identical to the one used for the inverted Golomb code.

i-a) Encoding $\underline{0^{N_{\mathrm{start}}}\cdots}$ $(m \neq 0)$

XDG encoding begins with the code tree T_0 and outputs ‘1’ for input $\underline{0^L}$ so that ‘$1^{\lfloor N_{\mathrm{start}}/L \rfloor}$’ is output at first. Then, the following positive integer will be encoded by the code tree $T_{N_{\mathrm{start}}\%L}$, and thus we can expect the encoder will output at least ‘0PBC$_L[N_{\mathrm{start}}\%L]$’. Therefore, the codeword in $C_{\mathrm{XDG}}[\mathbf{x}|L]$ which the encoder can determine from the concerning part of the input is

$$\text{‘}1^{\lfloor N_{\mathrm{start}}/L \rfloor}0\mathrm{PBC}_L[N_{\mathrm{start}}\%L]\text{’}, \tag{5.41}$$

equivalent to $C_{\mathrm{iG}}[\mathbf{x}|L]$ in Eq. (5.37).

i-b) Encoding $\underline{0^{N_{\mathrm{start}}}}$ $(m = 0,\ N_{\mathrm{start}} \neq 0)$

If no positive integer follows the input $\underline{0^{N_{\mathrm{start}}}}$, the encoder need not to output ‘0PBC$_L[N_{\mathrm{start}}\%L]$’ mentioned above. However, if $N_{\mathrm{start}}\%L \neq 0$, the decoder cannot know only from ‘$1^{\lfloor N_{\mathrm{start}}/L \rfloor}$’ whether this is the end of the codeword or not because there are $N_{\mathrm{start}}\%L$ integers remaining to be decoded. Therefore, it is reasonable to terminate ‘$1^{\lfloor N_{\mathrm{start}}/L \rfloor}$’ with ‘1’ in this case: The decoder can finish the decoding by determining all the remaining integers to be zero. Including $N_{\mathrm{start}}\%L = 0$, $C_{\mathrm{XDG}}[\mathbf{x}|L]$ for this case can be written as

$$\text{‘}1^{\lceil N_{\mathrm{start}}/L \rceil}\text{’}, \tag{5.42}$$

equivalent to $C_{\mathrm{iG}}[\mathbf{x}|L]$ in Eq. (5.38).

ii-a) Encoding $\underline{\cdots x_i 0^{N_i} \cdots}$ $(m \neq 0,\ 0 < i < m)$

XDG encoding begins with the code tree $T_{(N_{i-1}+1)\%L}$ (here, we regard N_0 as $(N_{\mathrm{start}}-1)$) after encoding the previous part. As explained in i-a), the encoder has already outputted

‘$0\mathrm{PBC}_L[(N_{i-1}+1)\%L]$’. Therefore, the input $\underline{x_i}$ leads to the output as

$$‘0\mathrm{PBC}_L[0]0\mathrm{PBC}_L[0]\cdots 0\mathrm{PBC}_L[0] = ‘0^{(K+f[0|l])(x_i-1)}’. \tag{5.43}$$

Then, the encoder starts encoding $\underline{0^{N_i}}$ using the code trees beginning with T_1. If the input sequence continues with a positive integer, it will be encoded with the code tree $T_{(N_i+1)\%L}$ Derived from the same logic as in i-a), the codeword that the encoder can determine from $\underline{0^{N_i}}$ is ‘$1^{\lfloor (N_i+1)/L \rfloor}0\mathrm{PBC}_L[(N_i+1)\%L]$’. Eventually, the total output codeword for the concerning part results in

$$‘0^{(K+f[0|l])(x_i-1)}1^{\lfloor (N_i+1)/L \rfloor}0\mathrm{PBC}_L[(N_i+1)\%L]’, \tag{5.44}$$

equivalent to $C_{\mathrm{iG}}[\mathbf{x}|L]$ in Eq. (5.39).

ii-b) Encoding $\underline{\cdots x_m 0^{N_m}}$ $(m \neq 0)$

The output for $\underline{x_m}$ part is the same as in Eq. (5.43) at $i = m$. If no positive integer follows the input $\underline{\cdots x_m 0^{N_m}}$, the encoder needs not to output ‘$0\mathrm{PBC}_L[(N_{m-1}+1)\%L]$’ as in ii-a). However, if $(N_{m-1}+1)\%L \neq 0$, the decoder cannot know only from ‘$1^{\lfloor (N_i+1)/L \rfloor}$’ whether this is the end of the codeword or not because there are $(N_i+1)\%L$ integers remaining to be decoded. Therefore, as in i-b), we can keep the decodability with minimum bits by terminating the codeword with ‘1’. Including $(N_{m-1}+1)\%L = 0$, $C_{\mathrm{XDG}}[\mathbf{x}|L]$ for the concerning part results in

$$‘0^{(K+f[0|l])(x_m-1)}1^{\lceil (N_m+1)/L \rceil}’, \tag{5.45}$$

equivalent to $C_{\mathrm{iG}}[\mathbf{x}|L]$ in Eq. (5.40).

Since we have $C_{\mathrm{iG}}[\mathbf{x}|L] = C_{\mathrm{XDG}}[\mathbf{x}|L]$ in all cases of i-a), ii-a), i-b) and ii-b), XDG code is uniquely decodable in the same way as the inverted Golomb code.

5.2.9 Symmetry of relative redundancy to Golomb code

As we have shown in Sec. 5.2.6, we here prove the symmetry between the proposed XDG code and the Golomb code. The average code length $\bar{\mathcal{L}}_1^{\mathrm{XG}}(\theta|L)$ of XDG code with the parameter L is given from the stationary probability p_i^* $(i = 0, 1, \cdots, L-1, \sum_i p_i^* = 1)$ of each code tree T_i as

$$\begin{aligned}
\bar{\mathcal{L}}_1^{\mathrm{XG}}(\theta|L) &= p_0^* \sum_{X=0}^{\infty} p_{\mathrm{Geo}}[X|\theta]\{(K+f[0|l])X+1\} \\
&\quad + \sum_{i=1}^{L-1} p_i^* \sum_{X=1}^{\infty} p_{\mathrm{Geo}}[X|\theta]\{(K+f[0|l])X+(f[i|l]-f[0|l])\} \\
&= (K+f[0|l]) \sum_{X=0}^{\infty} p_{\mathrm{Geo}}[X|\theta]X + p_0^* + \sum_{i=1}^{L-1} p_i^* \sum_{X=1}^{\infty} p_{\mathrm{Geo}}[X|\theta](f[i|l]-f[0|l])
\end{aligned} \tag{5.46}$$

Similar to Eq. (5.29), the stationary probabilities $\{p_i^*\}_{i=0}^{l-1}$ are given as

$$p_i^* = \begin{cases} \frac{\theta(1-\theta)^{L-1}}{1-(1-\theta)^L} & (\text{if } i=0) \\ \\ \frac{\theta(1-\theta)^{i-1}}{1-(1-\theta)^L} & (\text{otherwise}) \end{cases}, \tag{5.47}$$

resulting in

$$\begin{aligned}
\bar{\mathcal{L}}_1^{\mathrm{XG}}(\theta|L) &= (K+f[0|l])\frac{\theta}{1-\theta} + \frac{\theta(1-\theta)^{L-1}}{1-(1-\theta)^L} \\
&\quad + \sum_{i=1}^{L-1} \frac{\theta^2(1-\theta)^{i-1}}{1-(1-\theta)^L}(f[i|l]-f[0|l]) \\
&= \frac{K\theta}{1-\theta} + \sum_{i=0}^{L-1} \frac{\theta^2(1-\theta)^{i-1}}{1-(1-\theta)^L} f[i|l] + \frac{\theta(1-\theta)^{L-1}}{1-(1-\theta)^L}
\end{aligned} \tag{5.48}$$

For Golomb code, its average code length $\bar{\mathcal{L}}_1^{\mathrm{G}}(\theta|S)$ with Golomb parameter S is written as

$$\begin{aligned}
\bar{\mathcal{L}}_1^{\mathrm{G}}(\theta|S) &= \sum_{X=0}^{\infty} p_{\mathrm{Geo}}[X|\theta](R + f\left[X - S\left\lfloor S^{-1}X\right\rfloor |s\right] + \left\lfloor S^{-1}X\right\rfloor) \\
&= R + \sum_{i=0}^{S-1}\sum_{X=0}^{\infty} p_{\mathrm{Geo}}[SX+i|\theta] f[i|s] + \sum_{X=0}^{\infty} p_{\mathrm{Geo}}[X|\theta] S^{-1}X \\
&\quad - \sum_{i=1}^{S-1}\sum_{X=0}^{\infty} p_{\mathrm{Geo}}[SX+i|\theta] S^{-1} i \\
&= R + \sum_{i=0}^{S-1} \frac{(1-\theta)\theta^i}{1-\theta^S} f[i|s] + \frac{S^{-1}\theta}{1-\theta} - \left(\frac{S^{-1}\theta}{1-\theta} - \frac{\theta^S}{1-\theta^S}\right) \\
&= R + \sum_{i=0}^{S-1} \frac{(1-\theta)\theta^i}{1-\theta^S} f[i|s] + \frac{\theta^S}{1-\theta^S}
\end{aligned} \tag{5.49}$$

with the same conditional branch function $f[i|l]$ stated above.

Comparing with the entropy, the respective relative redundancy $\mathcal{R}_1^{\mathrm{XG}}(\theta|L)$ and $\mathcal{R}_1^{\mathrm{G}}(\theta|S)$ become

$$\begin{aligned}\mathcal{R}_1^{\mathrm{XG}}(\theta|L) &= \frac{\bar{\mathcal{L}}_1^{\mathrm{XG}}(\theta|L)}{\bar{\mathcal{L}}_1^{\mathrm{ideal}}(\theta)} - 1 \\ &= \left\{ K\theta + \sum_{i=0}^{L-1} \frac{\theta^2(1-\theta)^i}{1-(1-\theta)^L} f[i|l] + \frac{\theta(1-\theta)^L}{1-(1-\theta)^L} \right\} \\ &\cdot[-\theta \log_2 \theta - (1-\theta)\log_2(1-\theta)]^{-1} - 1\end{aligned} \tag{5.50}$$

$$\begin{aligned}\mathcal{R}_1^{\mathrm{G}}(\theta|S) &= \frac{\bar{\mathcal{L}}_1^{\mathrm{G}}(\theta|S)}{\bar{\mathcal{L}}_1^{\mathrm{ideal}}(\theta)} - 1 \\ &= \left\{ R(1-\theta) + \sum_{i=0}^{S-1} \frac{(1-\theta)^2\theta^i}{1-\theta^S} f[i|s] + \frac{(1-\theta)\theta^S}{1-\theta^S} \right\} \\ &\cdot[-\theta \log_2 \theta - (1-\theta)\log_2(1-\theta)]^{-1} - 1\end{aligned} \tag{5.51}$$

Here, the parameters K, R, l, and s satisfy $K = R$ and $l = s$ when $L = S$ so that the symmetry is guaranteed as

$$\mathcal{R}_1^{\mathrm{XG}}(1-\theta|L) = \mathcal{R}_1^{\mathrm{G}}(\theta|S) \text{ when } L = S. \tag{5.52}$$

Note that these average code length and relative redundancy are identical to the case in XDGR and Rice codes when $l = s = 0$.

5.2.10 Parameter estimation for extended-domain Golomb code

The symmetry between XDG and Golomb codes stated above also provides us with some suggestion for the parameter estimation of the proposed code. Owing to the simplicity of the structure of the Golomb code, its code-minimizing parameter S can be approximately estimated from the empirical average M of the input source symbol sequence as an integer close to $(\ln 2)M$. On the other hand, the proposed XDG code is difficult to minimize its actual code length directly.

Therefore, it is practical to derive its optimal parameter L from using the symmetry in Eq. (5.52) and estimating the average of the geometric source $p_{\mathrm{Geo}}[X|\theta]$ when the average of $p_{\mathrm{Geo}}[X|1-\theta]$ is M. The average of $p_{\mathrm{Geo}}[X|\theta]$ and $p_{\mathrm{Geo}}[X|1-\theta]$ are respectively $\theta(1-\theta)^{-1}$ and $(1-\theta)\theta^{-1}$, and thus

$$\frac{1-\theta}{\theta} = M \Rightarrow \frac{\theta}{1-\theta} = \frac{1}{M}. \tag{5.53}$$

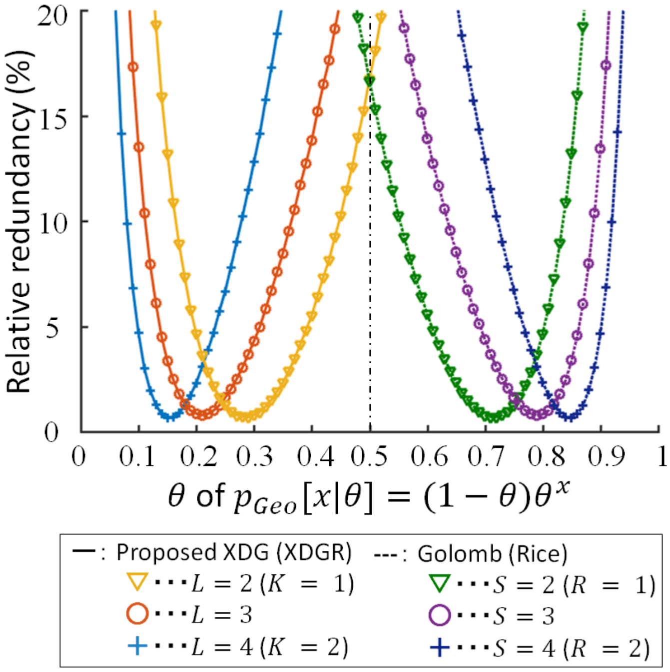


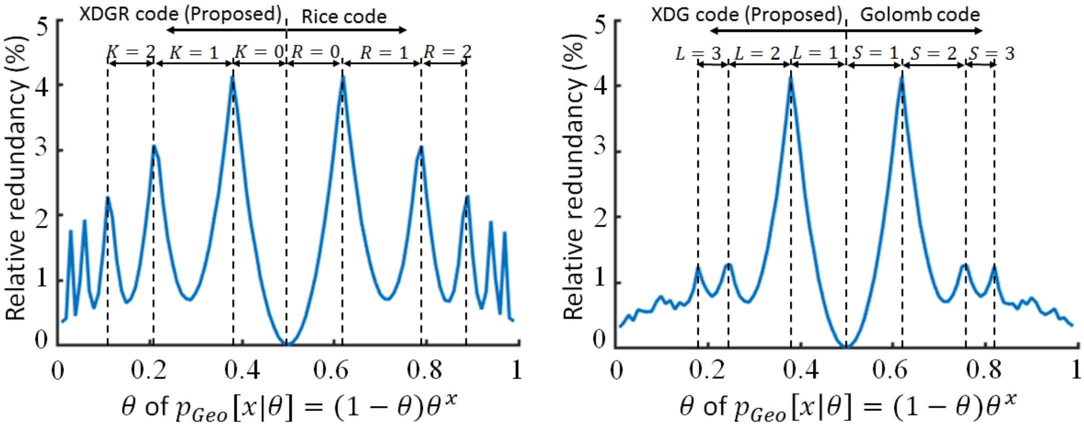
Fig. 5.8. Relative redundancy of the proposed XDG (XDGR) and conventional Golomb (Rice) codes with respective parameters L (K) and S (R) for random number sequences belonging to each geometric source.

Since the optimal parameter S minimizing Golomb code length of the sequence with average M^{-1} is given from $(\ln 2)M^{-1}$, the optimal parameter L for the proposed XDG encoding the sequence with the empirical average M should be an integer close to $(\ln 2)M^{-1}$.

From Eq. (5.52), the estimation of the parameter L is guaranteed to be, if the estimation of the average M is precise enough, as accurate in the sense of relative redundancy as the parameter estimation of Golomb code. It should also be noted that parameters L and K of XDG and XDGR codes have a logarithmic relationship. Therefore, the conclusion above does not conflict with the result in the approximate solution in Eq. (5.24) for code-length minimization of XDGR code.

5.2.11 Simulation by random numbers

We simulated by using random numbers to support the above theories. First, we focused on the code length of the proposed and conventional codes with fixed XDG parameter L, or XDGR parameter K, and Golomb parameter S, or Rice parameter R, to see their



(a) Result for the combined XDGR and Rice code using estimated parameters (K or R).

(b) Result for the combined XDG and Golomb code using estimated parameters (L or S).

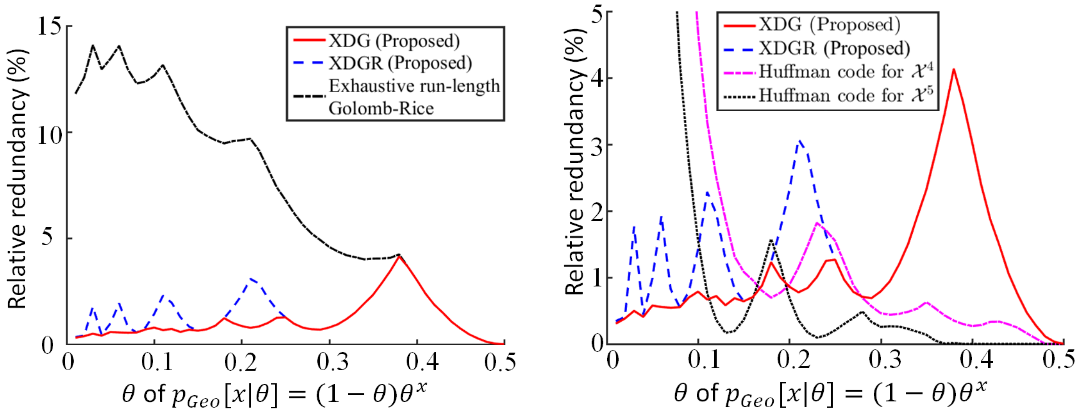
Fig. 5.9. Relative redundancy of the combined codes using estimated parameters for each random number sequences belonging to each geometric source.

symmetry. For each source $p_{\mathrm{Geo}}[x|\theta]$ of $\theta = 0.01, 0.02, 0.03, \cdots, 0.99$, $N = 10^6$ symbols of stationary memoryless non-negative random integer x were generated by the inversion method [91] and coded into the proposed code.

Fig. 5.8 compares the codes for each parameter. It can be seen that the curves of the relative redundancy showed symmetry for $\theta = 0.5$ for each $L = S$, or $K = R$, consistent with the theory. Note that the codes are identical when $L = S = 1$ and $K = R = 0$.

Next, we coded the same sequences into codes combining XDGR with Rice and XDG with Golomb codes to see the symmetry for a further range of the parameters. Their parameters were estimated for each sequence by the method stated previously and the one in [92]. The codes were switched source by source, between XDGR and Rice codes, and between XDG and Golomb codes, depending on which of their estimated parameters were the larger.

Fig. 5.9 shows the results. The relative redundancy of Golomb and Rice codes matched to the result in the other works [93,94]. The symmetry in the curves of relative redundancy appeared both between XDGR and Rice codes and between XDG and Golomb codes, which confirms the parameters to have been correctly estimated. The peaks in relative redundancy of the proposed XDG code were lower than the ones of the proposed XDGR



(a) Comparison with run-length GR code in [84]. Black long dashed short dashed line shows the result for run-length GR code with its bits for run length searched exhaustively.

(b) Comparison with Huffman codes. Purple long dashed short dashed line and black dotted line show the results for Huffman codes optimized for $\mathcal{X}^4$ and $\mathcal{X}^5$, respectively.

Fig. 5.10. Relative redundancy of each code. Red solid line and blue dashed line respectively indicate the results for the proposed XDG and XDGR codes with estimated parameters for each geometric source.

code because it was more flexibly tunable as Golomb code was.

Finally, we compared the relative redundancy with other codes, run-length Golomb–Rice (RLGR) code proposed in [84] and Huffman codes using the sparser half of the geometric sources, $p_{\text{Geo}}[x|\theta]$ with $\theta = 0.01, 0.02, 0.03, \cdots, 0.50$, and generating $N = 10^6$ stationary memoryless non-negative random integer symbols.

Fig. 5.10 shows the comparison results. For Huffman codes, we took a histogram of the existing symbols $\mathcal{X}$ in the random integer sequence for each source and made codebooks optimized for the histograms of its Cartesian products $\mathcal{X}^4$ and $\mathcal{X}^5$. The curve of the proposed XDG code was under one of the proposed XDGR code, both outperforming the run-length GR and Huffman codes, especially at small θs, or sparse sources.

5.3 Shaping by integer mappings for generalized-Gaussian-distributed sources

5.3.1 Integer mappings for controlling shapes

The discussions in the previous chapters for the signal modeling revealed that, with appropriate spectral envelope estimation, we have a chance to enhance the compression efficiency by extending the assumption of the distributions to GGDs. Furthermore, switching their shape parameter α frame by frame depending on the acoustic features of the input signals may also be effective for enhancing compression efficiency. Therefore, if we can design a low complexity well-structured code for various shape parameters, we may make more use of these perspectives. Based on this motivation, we here focus on the discrete type of GGDs

$$p_\alpha[X|\phi] = \frac{1}{Z_\alpha(\phi)} 2^{-\left(\frac{X}{\phi}\right)^\alpha}, \;\; Z_\alpha(\phi) = \sum_{k=0}^{\infty} 2^{-\left(\frac{k}{\phi}\right)^\alpha}, \tag{5.54}$$

as an approximation of discrete distributions strictly derived from the integrals of (3.28) and discuss how to construct a code with its expected code length close to the theoretical limit

$$\mathcal{L}_\alpha^{\mathrm{ideal}}[X|\phi] = -\log_2 p_\alpha[X|\phi] = \left(\frac{X}{\phi}\right)^\alpha + \log_2 Z_\alpha(\phi), \tag{5.55}$$

proportionate to the α-th power of input integer X.

Some previous works tackled this problem by extending the GR code. For distributions with small α, the work in [85] combined GR code with exponential-Golomb (exp-Golomb) code, a code representing integers by a suffix code of which code length written by a unary code. To apply the code to a broader range of α, the work in [84] combined GR code with zero-run-length code. However, the tunable parameters of the method are related to the shape parameter α in a complicated way, resulting in a sophisticated iterative algorithm. Therefore, we here present a shaping method based on the idea in [95].

In the work of [95], a pair of integers is mapped into a single integer, based on the ℓ^2 norm, shaping their distribution into a Laplacian distribution before GR encoding. To decode in this scheme, we have only to read GR code and perform inverse mapping of the decoded integers. This code becomes an FV code for two-dimensional (2-D) non-negative

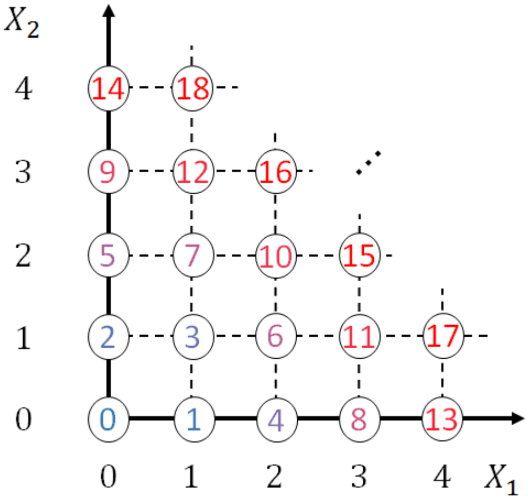


Fig. 5.11. Outputs (numbers in each lattice point) of 2-D integer nesting pairs of integers (X_1, X_2).

integers $\mathcal{X}^2$. Generally, FV codes for 2-D integers do not precisely achieve the entropy of the target distribution unless we assume some exceptional cases: Laplacian of scale 1 or correlated inputs as in [95]. However, FV coding has advantages in computational complexity compared to variable-to-variable-length (VV) coding, such as arithmetic coding.

Extending the above idea to a mapping method between a group of integers and a single integer, we will show that GR code can be flexibly adapted to the GGDs. It is memory- or time-consuming to prepare many mapping patterns for every non-negative integer pairs. Therefore, we here introduce algebraic mappings, which can approximately shape integers into Laplacian distributions.

5.3.2 Integer nesting

Case of 2-D inputs

The mappings to use for GR code in this context have to be bijective: They should be surjective to have all the integers encoded; they should be injective so that the codes are decoded in a completely lossless manner. Here, we first focus on the case where shape parameter α is a natural number p. The mappings required in this case should be bijective in non-negative integer space and raise the input values to their pth power. With the freedom of the input and output spaces of the mappings taken into account, it is

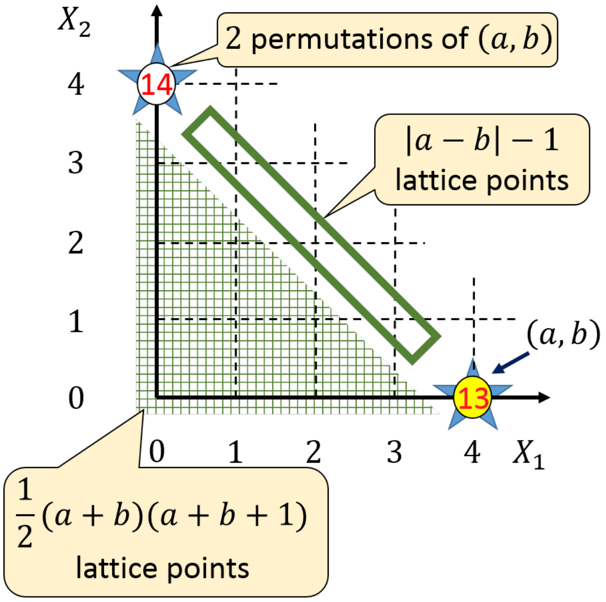
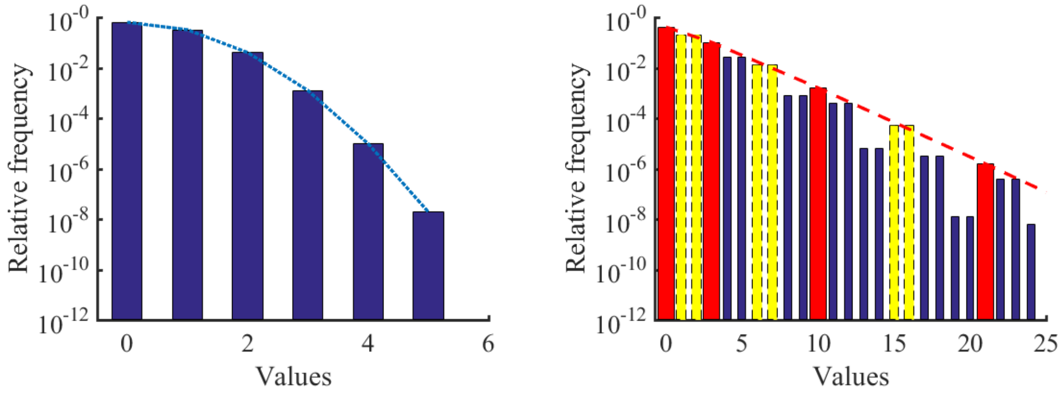


Fig. 5.12. Example of calculating the number for a given lattice point (a, b) in the case of $a \neq b$.

intuitive to employ mappings from p-D integers to a single integer, that we here propose as p-D integer nesting. Of course, it is not impossible to make a mapping between other spaces under the constraints mentioned above, but by focusing on mappings from p-D integer space to 1-D integer space, we can design much simpler rules to make one roughly satisfy the constraints.

The approach of the proposed p-D integer nesting is to number each lattice point in p-D integer space to make the numbers distribute approximately as Laplacian distributions. Nonetheless, there are many possibilities for numbering the lattice points, and it is difficult to analyze theoretically. Therefore, we experimentally, by trial and error, found practical numbering in the order based on specific rules: In comparing two lattice points, the one with the larger ℓ^1 norm should be numbered with a larger value; if the ℓ^1 norms are identical, the one with the larger l^1 distance from the lattice point of uniform elements with the same ℓ^1 norm should be numbered with a larger value.

Let us show an example in the case of $p = 2$, which aims at integers belonging to Gaussian distribution $p_2[X|2^r]$, as in Fig. 5.11. The 2-D integer lattice points (X_1, X_2) are numbered in the order of the ℓ^1 norm, and the lattice points with the same ℓ^1 norm are arranged in ascending order of the differences $|X_1 - X_2|$. The lattice points with



(a) Histogram of GGD of shape parameter $\alpha = 2$.

(b) Histogram of integers given by nesting integers belonging to GGD of shape parameter $\alpha = 2$.

Fig. 5.13. Changes in a histogram of GGD of shape parameter $\alpha = 2$ by 2-D integer nesting. Red fat bars and yellow bars surrounded by dashed lines in (b) respectively show the relative frequency of the values given by nesting two equal integers and by nesting two consecutive integers. The blue dotted line and red dashed line respectively show the outlines of the histogram and the part of the histogram highlighted in red.

the same ℓ^1 norm and $|X_1 - X_2|$ are the permutations of the elements, and their order will not change the expected code length of the encoding mentioned later. Therefore, we arbitrarily ordered in a way that the ones with $X_1 > X_2$ come first when ℓ^1 norm and $|X_1 - X_2|$ are the same.

Fig. 5.12 shows how to calculate the numbering from input integers (a, b) in the case of $a \neq b$. First, we count all the lattice points having the lower ℓ^1 norm, which forms a triangle, including $(a + b)(a + b + 1)/2$ lattice points. Then, the ones having the same ℓ^1 norm and lower differences $|X_1 - X_2|$, which forms a line beside (a, b), are counted as $|a - b| - 1$. Finally, a number is added to the above counts to distinguish (a, b) from its permutation (b, a). As a result, the proposed 2-D integer nesting from (X_1, X_2) to an integer Y based on this numbering is written, with a function $f(\text{statement})$ outputting 1 if the statement is true and 0 otherwise, as

$$Y = \frac{1}{2}(X_1 + X_2)(X_1 + X_2 + 1) + |X_1 - X_2| - f(X_1 > X_2), \tag{5.56}$$

which also holds for $X_1 = X_2$ and is naturally bijective and thus invertible mapping. An inverse mapping, which will be shown in detail later, can be designed by finding $(X_1+X_2)(X_1+X_2+1)/2$, $|X_1-X_2|$, and $f(X_1 > X_2)$ in this order from Y. By encoding the nested integer Y using GR code, say with Rice parameter $r = 0$, the average code length of each X_1 and X_2 will be

$$\begin{aligned}\mathcal{L}_2^{\mathrm{GR}}[X_1, X_2] = &\left(\frac{X_1+X_2}{2}\right)\cdot\left(\frac{X_1+X_2}{2}+0.5\right)\\ &+\left|\frac{X_1-X_2}{2}\right|+\frac{1}{2}f(X_1 \le X_2). \qquad (5.57)\end{aligned}$$

Although the above mapping is heuristic, its effect on expected code length can be interpreted by a rough approximation. The expected code length is given by

$$\begin{aligned}&E_{X_1,X_2}\left[\mathcal{L}_2^{\mathrm{GR}}[X_1, X_2]\right] \qquad (5.58)\\ &=\sum_{X_1=0}^{\infty}\sum_{X_2=0}^{\infty}\mathcal{L}_2^{\mathrm{GR}}[X_1, X_2]p_2[X_1|1]p_2[X_2|1]\end{aligned}$$

and it can be seen from Eq. (5.57) that the first term, depending on the average $(X_1 + X_2)/2$, is dominant in $\mathcal{L}_2^{\mathrm{GR}}[X_1, X_2]$. In addition, among the terms in Eq. (5.59) having the same average $(X_1 + X_2)/2$, the terms with $X_1 \simeq X_2$ are dominant because of $p_2[X_1|1]p_2[X_2|1]$. This results in a coarse approximation of Eq. (5.59) using $X = X_1 \simeq X_2$ and $X(X + 0.5) \simeq X^2$ as

$$E_{X_1,X_2}\left[\mathcal{L}_2^{\mathrm{GR}}[X_1, X_2]\right] \simeq \sum_{X=0}^{\infty}\left(X^2 + \mathit{const.}\right)p_2[X|1] \qquad (5.59)$$

where the *const.* will be a little larger than 0.5 due to the approximation errors, which is close to the theoretical limit of the expected code length using the ideal code length $\mathcal{L}_2[X|1]$, the one from Eq. (5.55) in the case of $\alpha = 2$:

$$\begin{aligned}E_X\left[\mathcal{L}_2[X|1]\right] &= \sum_{X=0}^{\infty}\left(X^2 + \log_2 Z_2(1)\right)p_2[X|1] \qquad (5.60)\\ &= \sum_{X=0}^{\infty}\left(X^2 + 0.6457\cdots\right)p_2[X|1].\end{aligned}$$

Fig. 5.13 shows an example of histograms to support qualitatively the approximation. One of the Gaussian distributions, $p_2[X|1]$, is shown Fig. 5.13 (a), which has an outline forming a quadratic curve on a logarithmic axis. It gets shaped by the 2-D integer nesting, as in Fig. 5.13 (b), which is given by the nested numbers. We can see that the outline of

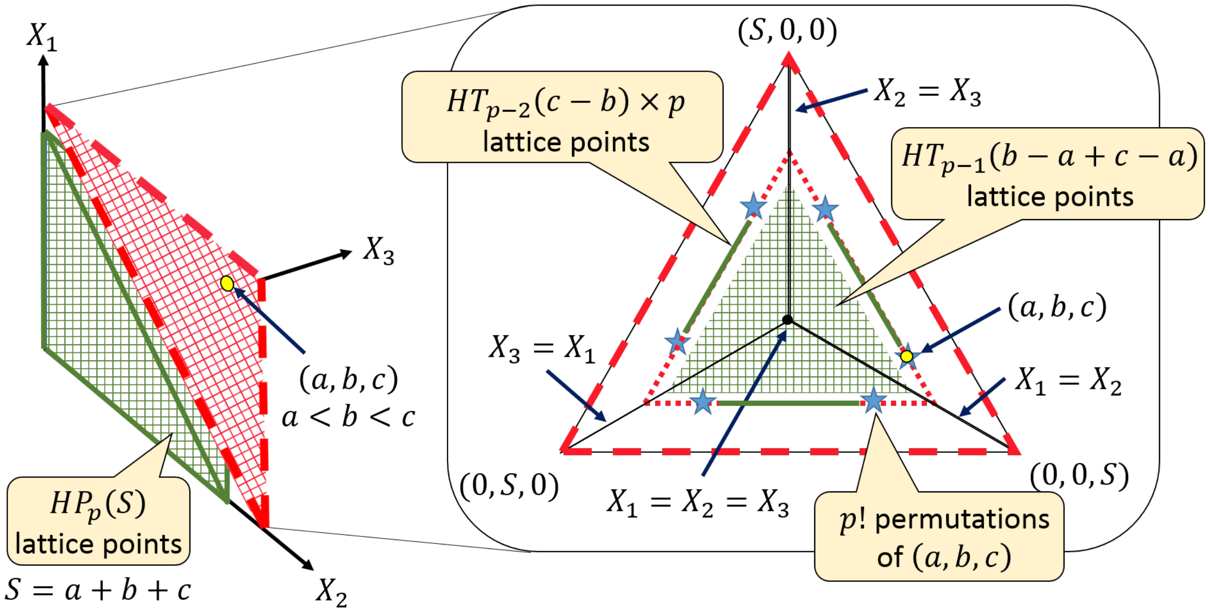


Fig. 5.14. Example of calculating the number for a given $p(=3)$-D lattice point (a,b,c) in case of $a<b<c$.

the bars for the values given from the integer pairs (X_1, X_2) of $|X_1 - X_2| \leq 1$ highlighted in red and yellow, which have dominant probability in Eq. (5.59), forms an almost straight line such as the one for Laplacian distributions does.

Case of p-D inputs

The proposed integer nesting can naturally be generalized for arbitrary natural number p by an iterative algorithm extended from the counting method in Fig. 5.12. An example is shown in Fig. 5.14, which interprets the generalization of the triangle and line in Fig. 5.12 as a hyper trigonal pyramid and hyper regular triangle. First, we count all the lattice points having lower ℓ^1 norms, which form a p-D hyper pyramid. Let us define the number of lattice points included in the hyper pyramid, including ones on the sides, as $HP_p(S)$, where S is the ℓ^1 norm of the input. It is identical to the number of possible integer combinations $(X_1, X_2, \cdots, X_p)$ satisfying

$$\sum_{i=1}^{p} X_i < S \text{ s.t. } X_i \geq 0 \ (i = 1, 2, \cdots, p), \tag{5.61}$$

which is written using the combination function $C(n,k) = n!/(n-k)!/k!$ as

$$HP_p(S) = C(S+p-1, p). \tag{5.62}$$

Then, focusing on the $(p-1)$-D hyper triangle, shown in the right-hand side of Fig. 5.14, formed by the lattice points having the same ℓ^1 norm, the ones included in the inner hyper triangle are counted. The inner hyper triangle, in this case where a is the minimum in the elements, is surrounded by a hyper triangle composed of vertices $(b+c-a,a,a)$, $(a,b+c-a,a)$, and $(a,a,b+c-a)$, and thus the number of included lattice points is identical that included in a hyper triangle composed of vertices $(b-a+c-a,0,0)$, $(0,b-a+c-a,0)$, and $(0,0,b-a+c-a)$, excluding the sides. By defining $HT_{p-1}(S')$ as the amount of lattice points included in a $(p-1)$-D hyper triangle composed of vertices $(S',0,0,\cdots,0)$, $(0,S',0,\cdots,0)$, $\cdots$, $(0,0,\cdots,0,S')$ excluding the sides, it can be counted as the number of possible integer combinations $(X_1,X_2,\cdots,X_p)$ satisfying

$$\sum_{i=1}^{p} X_i = S' \text{ s.t. } X_i > 0 \ (i=1,2,\cdots,p) \tag{5.63}$$

and written as

$$HT_{p-1}(S') = C(S'-1,p-1). \tag{5.64}$$

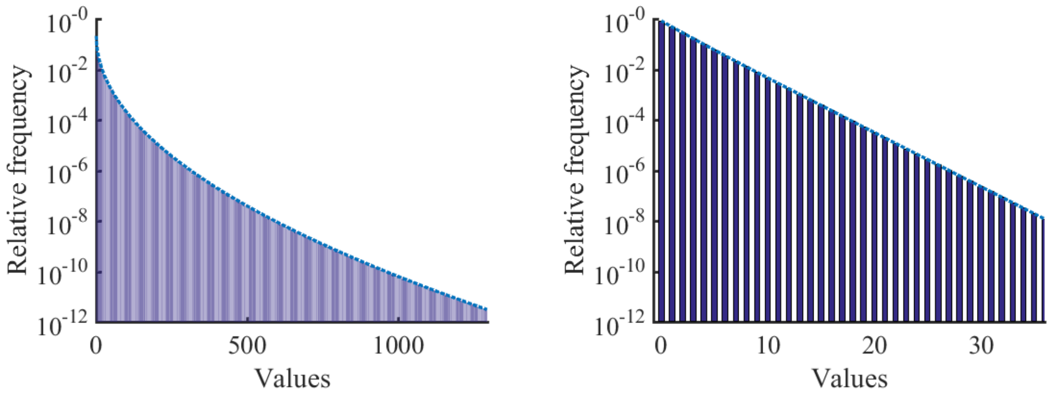
Therefore, the lattice points included in the inner hyper triangle amount to $HT_{p-1}(b-a+c-a)$.

Since the inner hyper triangle has $P(p,1)$ sides, where $P(n,k)=n!/(n-k)!$ is the permutation function, there neighbor $P(p,1)$ of $(p-2)$-D hyper triangles. The lattice points included in their inner hyper triangles are counted similarly, neglecting the smallest-valued element used in the previous iteration. After iteratively counting the ones in $(p-1)$-D to 1-D inner hyper triangles, the $p!$ permutations of the inputs, in the case where all the elements of the input have different values, are numbered to distinguish them from one another. Whenever there is more than one element taking the minimum value, the dimension of the hyper triangles and the permutations will degenerate.

The above method is summarized as a pseudo-code in Algorithm 5. To make it hold for every case, $C(n,k)$ and $P(n,k)$ are both defined to output 1 when $k=0$ and 0 when $n<k$ or when n or k is negative. For the same reason explained above, the pth powers of the inputs roughly show dominant effects on the outputs of p-D integer nesting, and thus the nesting is expected to shape the GGDs of shape parameter $\alpha=p$ into Laplacian distributions approximately in the sense of the expected code length.

Algorithm 5 p-D integer nesting integers $(X_1, X_2, \ldots, X_p)$ into a integer Y

```
Set $X$ as the copy of the input.
Set $I$ as a number of unique values in $X$.
Set $L_i = 0$ and $M_i = 0$ for $i = 1, 2, \ldots, p$.
Set $n = 0$, $i = 0$, $X_{\min} = -1$, and $\bar{M} = 1$.
Set $S$ as the sum of the input.
Set $Y = C(S + p - 1, p)$.
while $n < p$ do
  Find the minimum in the input and replace $X_{\min}$ with it. If there are more than one, choose the leftmost.
  Increment $Y$ by
  $P(p, n)C(S - X_{\min}(p - n) - 1, p - n - 1)/\bar{M}$.
  if $X_{\min}$ is equal to the previous one then
    Increment $M_i$ by 1.
    Multiply $\bar{M}$ by $M_i$.
  else
    Increment $i$ by 1.
    Set $M_i = 1$ and $L_i = X_{\min}$.
  end if
  Decrement $S$ by $X_{\min}$.
  Delete the chosen minimum from $X$.
  Increment $n$ by 1.
end while
Set $H = 0$, $i = 0$, and $N = p$.
while $i < I$ do
  Increment $i$ by 1.
  Set $n = 0$ and $m = M_i$.
  while $n < N$ do
    Increment $n$ by 1.
    if $n$-th element in the input is equal to $L_i$ then
      Increment $H$ by $C(N - n, m)$.
      Decrement $m$ by 1.
    end if
  end while
  Delete all $L_i$ from the input.
  Decrement $N$ by $M_i$.
  Multiply $H$ by $C(N, M_{i+1})$.
end while
Increment $Y$ by $H$ and output $Y$.
```



(a) Histogram of GGD of shape parameter $\alpha = 1/2$.

(b) Histogram of integers given by splitting integers belonging to GGD of shape parameter $\alpha = 1/2$.

Fig. 5.15. Changes in histogram of GGD of shape parameter $\alpha = 1/2$ by 2-D integer splitting. Blue dotted lines show the outlines of the histograms.

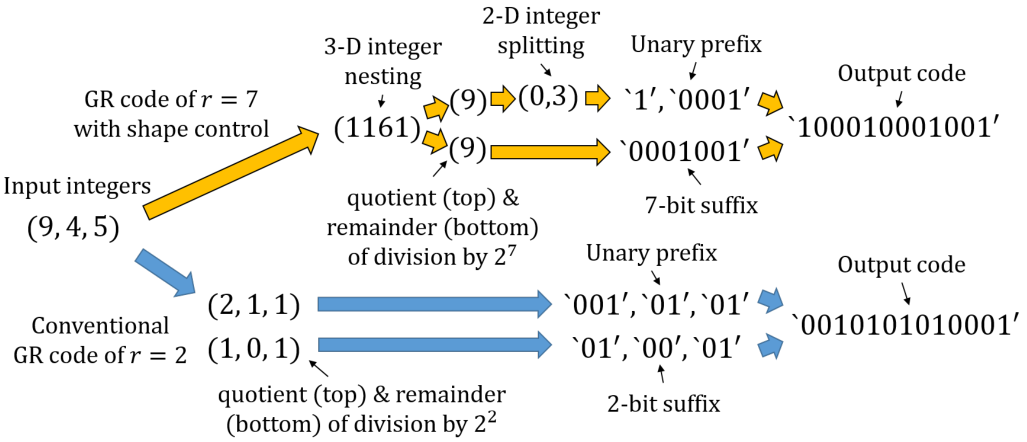


Fig. 5.16. Comparisons between the conventional GR encoding and GR encoding with the proposed shape control. The Rice parameters are set to the optimum value for each case.

5.3.3 Integer splitting

Case of 2-D outputs

The proposed integer nesting, as we can naturally understand from its concept, always

Algorithm 6 q-D integer splitting integer Y into integers $(X_1, X_2, \ldots, X_q)$

```
Set $S = 0$, $i = 0$, $n = 0$, $X_{\mathrm{min}} = -1$, and $\bar{M} = 1$.
Set $L_i = 0$ and $M_i = 0$ for $i = 1, 2, \ldots, q$.
Find the largest $S$ satisfying $Y \geq C(S + q - 1, q)$.
Decrement $Y$ by $C(S + q - 1, q)$.
while $n < q$ do
  Find the smallest $X_{\mathrm{min}}$ satisfying
  $Y \geq P(q, n)C(S - X_{\mathrm{min}}(q - n) - 1, q - n - 1)/\bar{M}$.
  Decrement $Y$ by
  $P(q, n)C(S - X_{\mathrm{min}}(q - n) - 1, q - n - 1)/\bar{M}$.
  if $X_{\mathrm{min}}$ is equal to the previous one then
    Increment $M_i$ by 1.
    Multiply $\bar{M}$ by $M_i$.
  else
    Increment $i$ by 1.
    Set $M_i = 1$ and $L_i = X_{\mathrm{min}}$.
  end if
  Decrement $S$ by $X_{\mathrm{min}}$.
  Increment $n$ by 1.
end while
Set $H = Y$, and $N = 0$.
Set $X$ as an empty set.
while $i > 0$ do
  Set $n = 0$ and $m = M_i$.
  Increment $N$ by $M_i$.
  Set $X$ as a set of $N$ elements $(X_1, X_2, \ldots, X_N)$.
  Get the quotient and remainder $h$ of $H$ divided by $C(N, M_i)$.
  Replace $H$ with the quotient.
  while $n < N$ do
    Increment $n$ by 1.
    if $h \geq C(N - n, m)$ then
      Set $X_n = L_i$.
      Decrement $h$ by $C(N - n, m)$.
      Decrement $m$ by 1.
    end if
  end while
  Replace the elements of $X$ where $L_i$ was not set with the previous $X$ in the same order.
  Decrement $i$ by 1.
end while
Output $X$ as $(X_1, X_2, \ldots, X_q)$.
```

has an inverse mapping uniquely giving a group of integers from a single integer, which we here present as integer splitting, Let us show an example of 2-D integer splitting, i.e., an inverse mapping of 2-D integer nesting mentioned in the previous section. The clue is that the input integer Y given by 2-D integer nesting (X_1, X_2) as in Eq. (5.56) satisfies the following inequalities in accordance with the sum of the nested integers $S = X_1 + X_2$:

$$\begin{aligned} &\frac{1}{2}S^2 < \frac{1}{2}S(S+1) \leq Y < \frac{1}{2}(S+1)(S+2) < \frac{1}{2}(S+2)^2 \\ &\iff \lfloor\sqrt{2Y}\rfloor - 1 \leq S \leq \lfloor\sqrt{2Y}\rfloor. \end{aligned} \tag{5.65}$$

Therefore, S is determined as

$$S = \lfloor\sqrt{2Y}\rfloor - f\left(Y < \frac{1}{2}\lfloor\sqrt{2Y}\rfloor\left(\lfloor\sqrt{2Y}\rfloor + 1\right)\right). \tag{5.66}$$

This corresponds to finding the maximum triangle in Fig. 5.12 that does not cover (X_1, X_2). Also, Eq. (5.56) can be rewritten as

$$|X_1 - X_2| - f(X_1 > X_2) = Y - \frac{1}{2}S(S+1). \tag{5.67}$$

The sum of $X_1 + X_2$ and $|X_1 - X_2|$ will always be an even number $2\max(X_1, X_2)$, which will lead to the 2-D integer splitting:

$$\begin{aligned} &\text{if } Y - S(S-1)/2 \text{ is even} \\ &\begin{cases} X_1 &= (S(S+3)/2 - Y)/2 \\ X_2 &= (Y - S(S-1)/2)/2 \end{cases} \\ &\text{otherwise} \\ &\begin{cases} X_1 &= (Y + 1 - S(S-1)/2)/2 \\ X_2 &= (S(S+3)/2 - Y - 1)/2 \end{cases} \end{aligned} \tag{5.68}$$

using S in Eq. (6.3).

As integer nesting, the proposed integer splitting can also shape the input distributions. This time, think of integers belonging to GGDs of shape parameter $\alpha = 1/q$ with a natural number q and splitting them into q integers each by q-D integer splitting. The splitting shapes the GGDs approximately into Laplacian distributions so that efficient compression can be achieved by GR encoding the split integers.

This fact can be easily understood by the expected code length when using 2-D integer splitting. Given an integer Y split into X_1 and X_2, their code length in GR code of Rice parameter $r = 0$ depends on S related to Y as in Eq. (6.3) and will be

$$\begin{aligned}\mathcal{L}^{\mathrm{GR}}_{1/2}[Y] &= (X_1 + 1) + (X_2 + 1) = S + 2 \\ &= \lfloor\sqrt{2Y}\rfloor - f\left(Y < \frac{1}{2}\lfloor\sqrt{2Y}\rfloor\left(\lfloor\sqrt{2Y}\rfloor + 1\right)\right) \\ &\quad +2. \end{aligned} \tag{5.69}$$

This code length is very close to the ideal one for the GGD $p_{1/2}[Y|0.5]$, i.e., $\mathcal{L}_{1/2}[Y|0.5]$ in Eq. (5.55) with $\alpha = 1/2$. In fact, the expected code length in this case is

$$E_Y\left[\mathcal{L}^{\mathrm{GR}}_{1/2}[Y]\right] \simeq \sum_{Y=0}^{\infty}\left(\sqrt{2Y} + \mathit{const.}\right)p_{1/2}[Y|0.5] \tag{5.70}$$

with *const.* a little larger than 1, due to $\sqrt{2Y} - 1 < \lfloor\sqrt{2Y}\rfloor \leq \sqrt{2Y}$ and $0 \leq f(\cdot) \leq 1$, while the theoretical limit is

$$E_Y\left[\mathcal{L}_{1/2}[Y|0.5]\right] = \sum_{Y=0}^{\infty}\left(\sqrt{2Y} + 1.4589\cdots\right)p_{1/2}[Y|0.5]. \tag{5.71}$$

Fig. 5.15 shows an example of histograms. One of GGDs, $p_{1/2}[X|0.5]$, is shown Fig. 5.15 (a), which has an outline forming an inverse quadratic curve in a logarithmic axis. It gets shaped by the 2-D integer splitting, as in Fig. 5.15 (b), which is given by the split numbers. We can see that the outline forms an almost straight line, such as the one for Laplacian distributions does, which is much more precise than in the case of the integer nesting shown in the previous section.

Case of q-dimensional outputs

As the integer nesting, the algorithm of the splitting can be generalized to q-D integer splitting. It consists of iteratively finding the largest hyper pyramid and inner hyper triangles that do not cover the output lattice point, as in Fig. 5.14. These processes determine the values in the outputs in ascending order, and the number for permutations is given after subtracting from the input integer the number of lattice points in the determined hyper pyramid and inner hyper triangles.

The procedure is summarized in Algorithm 6. Although the complexity becomes higher for larger q, finding the largest hyper pyramid, i.e., finding the largest S satisfying $Y \geq C(S+q-1, q)$ in the algorithm can be solved easily, by using the inequality corresponding to Eq. (5.65):

$$\lfloor\sqrt[q]{q!Y}\rfloor - (q-1) \leq S \leq \lfloor\sqrt[q]{q!Y}\rfloor. \tag{5.72}$$

For the same reason explained above, q-D integer nesting is expected to shape the GGDs of shape parameter $\alpha = 1/q$ into Laplacian distributions approximately in the sense of the expected code length.

5.3.4 Proposed shape control method

Summarizing the above discussion, for natural integers p and q, p-D integer nesting and q-D integer splitting integers enable us to approximately shape their distributions into Laplacian ones from the GGDs of shape parameters $\alpha = p$ and $1/q$, respectively. From these facts, we can expect to combine the nesting and splitting to deal with integers belonging to GGDs with shape parameters represented as $\alpha = p/q$, an arbitrary rational number, although it is a heuristic and non-trivial matter.

In the conventional GR encoding, the input integers are divided by 2^r using Rice parameter r, and the quotient and remainder are respectively represented as a unary prefix and an r-bit suffix. This separation can be interpreted as a control of the scale of the input distributions: The input integers are scaled by the division to be efficiently compressed by the unary code. In other words, their distribution, for example Laplacian distribution $p_1[X|2^r]$, changes to $p_1[X|1]$. From this perspective, the proposed integer nesting and splitting are the same kind of shaping operations as the division in the conventional GR encoding except that they change the shape instead of the scale.

An example of the proposed shape control in the GR encoding is shown in Fig. 5.16. The proposed encoding strategy performs the division by 2^r after integer nesting and performs integer splitting to the quotient, followed by the unary encoding. The remainder is represented by a r-bit suffix as in the conventional encoding. Practically, it is preferable to use the division between the nesting and splitting because we can use more precise Rice parameters. In the example, the proposed encoding can compress input $(9, 4, 5)$ into a total of 12 bits, while the conventional one cannot make it shorter than 13 bits of $r = 2$.

Given p and q, the Rice parameter to use can be estimated from the input integer sequence. Thinking of N integer-nested input integers $\{Y_k\}_{k=0}^{N-1}$, the length of the output code using Rice parameter r can be approximated in accordance with Eq. (5.72) as

$$\sum_{k=0}^{N-1} \left(\sqrt[q]{q! \frac{Y_k}{2^r}} + r + 1 \right), \tag{5.73}$$

Table 5.1. Numerically-calculated entropy, expected length of the proposed code, expected length of Huffman codes for the same input dimension, and the upper limit of expected code length generally guaranteed by Huffman coding scheme, respectively for input distribution $p_\alpha[X|\phi]$ in Eq. (5.54)

Shape parameter α of input	1/3	1/2	3/5	2/3	3/4	1	4/3	3/2	5/3	2	3
Scale ϕ of input	0.303	0.707	0.849	0.909	0.957	1.000	0.972	0.950	0.928	0.886	0.797
p for the proposed code (= Input dimension as a FV code)	1	1	3	2	3	1	4	3	5	2	3
q for the proposed code	3	2	5	3	4	1	3	2	3	1	1
Entropy (bit/sample)	6.414	4.161	3.425	3.062	2.703	2.000	1.489	1.323	1.192	1.001	0.728
Huffman code for $\mathcal{X}^p$ (bit/sample)	6.446	4.187	3.435	3.076	2.712	2.000	1.493	1.335	1.200	1.026	0.737
Proposed code (bit/sample)	**6.495**	**4.198**	**3.514**	**3.115**	**2.744**	**2.000**	**1.500**	**1.343**	**1.215**	**1.027**	**0.784**
General upper limit of Huffman coding scheme (bit/sample)	7.414	5.161	3.759	3.562	3.037	3.000	1.739	1.656	1.392	1.501	1.062

which is minimized by the stationary point of r:

$$r = q \log_2 \left(\frac{\ln 2}{qN} \sum_{k=0}^{N-1} \sqrt[q]{q! Y_k} \right), \tag{5.74}$$

depending on the $l^{1/q}$-norm of the nested integers.

5.3.5 Numerical calculation of theoretical expected code length

Actually, it is difficult to analyze theoretically the effects of the proposed shape control method on GR encoding, nevertheless we can calculate its theoretical expected code length. To compare the theoretical performance of the codes, we numerically calculated the expected length of the proposed codes with $r = 0$ and Huffman codes using the same input dimensions. Say the mapping by the combination of p-D integer nesting and q-D integer splitting is represented as $(X_1, X_2, \cdots, X_p) \to (Y_1, Y_2, \cdots, Y_q)$ the expected length of the proposed code with $r = 0$ for GGD $p_\alpha[X|\phi]$ is given as

$$\sum_{X_1, X_2, \cdots, X_p} \left[\frac{1}{p} \sum_{i=1}^{q} (Y_i + 1) \prod_{j=1}^{p} p_\alpha[X_j|\phi] \right]. \tag{5.75}$$

We truncated the GGDs at sufficiently large $(X_1, X_2, \cdots, X_p)$ of which corresponding term in Eq. (5.75) became lower than 10^{-10} to get the expected code length. The truncated distributions were also used for calculating the counterpart of Huffman codes for p-D integers, which corresponds to the theoretical limit of FV codes.

The numerical results for the expected code length are given in Table 5.1. It should be noted that the theoretical performance of the arithmetic coder matches the entropy because it can be regarded as optimal FV codes for large enough input dimensions. The shape parameters of the GGDs were set as $\alpha = p/q$ to match with the integer nesting and splitting dimensions. Since the scale ϕ, as we can see in the rough approximation in the integer splitting mentioned in the previous section, gets shifted by the proposed shape control, which makes the proposed code with $r = 0$ optimal for GGDs with its scale ϕ shifted from $\phi = 1$. Our preliminary test and some trials and errors revealed that the shift in scale ϕ actually slightly occurs also by the integer nesting and that the following heuristic formula gives us precise estimate: The integers X belonging to GGDs $p_\alpha[X|\phi]$ with $\alpha = p/q$ are approximately mapped by p-D integer nesting and q-D integer splitting to integers Y such as

$$Y \sim p_1 \left[Y|\phi/\Gamma(1 + \alpha^{-1})^{\alpha - 1} \right] \tag{5.76}$$

and thus we set ϕ for the numerical calculation using the gamma function $\Gamma(1 + N)$, a generalization of $N!$, as

$$\phi = \Gamma(1 + \alpha^{-1})^{\alpha - 1}. \tag{5.77}$$

This was motivated by the fact that the value for shifting the scale ϕ is exactly 1 when $\alpha = 1$, and is around 1 and $q!$ respectively when $\alpha = p$ and $\alpha = 1/q$, according to the discussions in the previous sections. Despite many heuristics and rough approximation, the expected code lengths shown in Table 5.1 are very close to one of Huffman codes

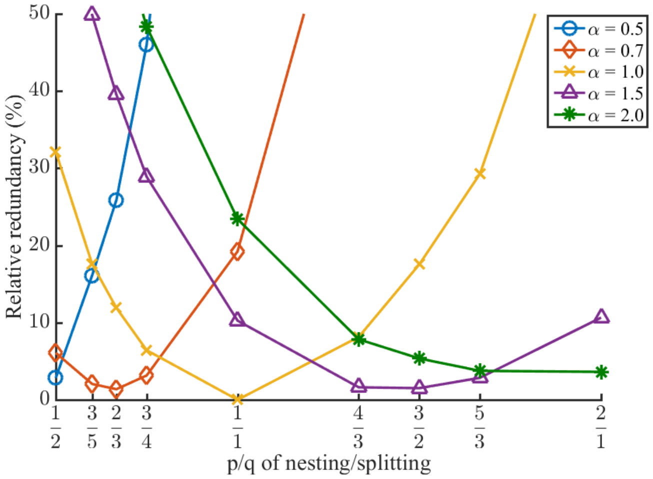


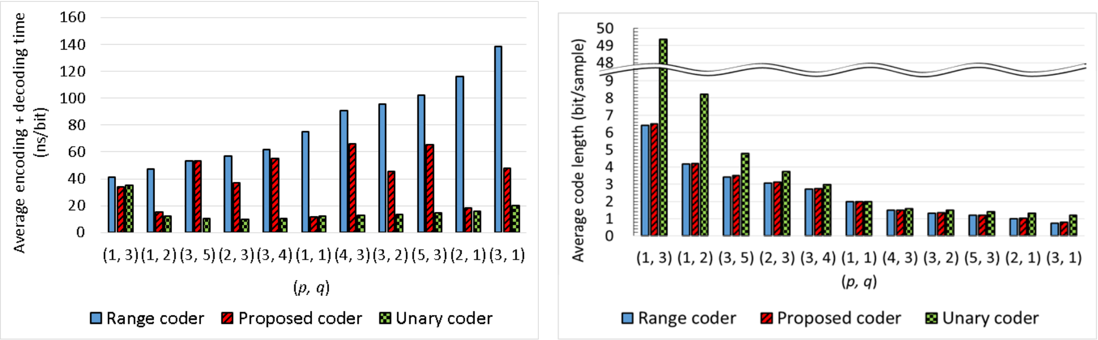
Fig. 5.17. Relative redundancy of the proposed codes, using each p and q for integer nesting and splitting, for random number sequences belonging to GGDs of respective shape parameter α with scale $\phi = 1$.

compared to the range generally guaranteed by the Huffman coding scheme, i.e., from the entropy H to $H + 1/p$. It can be said that the proposed code can theoretically achieve almost optimal expected code length as an FV code, at least under the demonstrated conditions.

It should be noted that we can estimate the scales of the nested or split integers and their optimal Rice parameter r as in Eq. (5.74) from the input integers. Therefore, the shift in the scales mentioned above will not be a problem in practice.

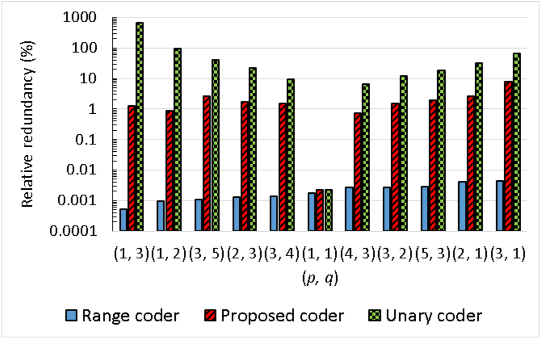
5.3.6 Simulation by random numbers

First, we focused on the code length for simulated random numbers of GR code combined with the proposed shape control with fixed parameters p and q. For each GGD $p_\alpha[X|1]$ of $\alpha = 0.5$, 0.7, 1.0, 1.5, and 2.0, $N = 10^6 + 80$ inputs of stationary memoryless non-negative random integer X were generated by the inversion method [91], shaped by p-D integer nesting and q-D integer splitting with $(p, q) = (1, 2)$, $(3, 5)$, $(2, 3)$, $(3, 4)$, $(1, 1)$, $(4, 3)$, $(3, 2)$, $(5, 3)$, and $(2, 1)$, and GR-encoded, where 1-D integer nesting/splitting do nothing. The sample number N was set to be dividable by 2,3,4, and 5. Fig. 5.17 compares the results for each distribution with p/q on the horizontal axis. The relative redundancy is defined here as the relative difference between the average code length and empirical



(a) Average total time for encoding and decoding normalized by the entropy of each input sequence.

(b) Average code length.



(c) Average relative redundancy.

Fig. 5.18. Comparison of trade-offs between compression speed and efficiency. The range coder was optimized for the empirical distribution of each input sequence. A machine with a 3.70-GHz Intel (R) Xeon (R) CPU E5-1630 v3, 16.0 GB memory and 64 bit windows 7 OS was used.

entropy. Note that $(p, q) = (1, 1)$ matches GR code with Rice parameter $r = 0$, or the unary code.

The code length for each distribution took a minimum at (p, q) that makes p/q nearest to shape parameter α. That supports the fact that the proposed shape control can shape GGDs of shape parameter $\alpha = p/q$ approximately into Laplacian distributions.

Next, to demonstrate the advantage of the proposed coder in computational complexity, we compared with the range coder [46], the fast implementation of arithmetic coding, and the unary coder. Computational complexity highly depends on the hardware to be used, and there are some standards for its evaluation, such as million instructions per second (MIPS), floating-point operations per second (FLOPS), and weighted million operations per second (WMOPS). However, analyzing those metrics for the coders requires to rewrite the source code and optimize it based on their specific basic operations. Therefore, for simplicity, we here estimated the complexity by the execution time of the coders using an Intel central processing unit (CPU) with the coders complied by the C-source compiler in Microsoft Visual Studio 2017. Since audio coders are recently used in such kind of conditions in many cases, we expect the execution time gives us a reasonable estimate of computational complexity in a practical sense.

For p-D integer nesting and q-D integer splitting, $(p, q) = (1, 3)$, $(1, 2)$, $(3, 5)$, $(2, 3)$, $(3, 4)$, $(1, 1)$, $(4, 3)$, $(3, 2)$, $(5, 3)$, $(2, 1)$, and $(3, 1)$ were used, and $N = 10^6 + 80$ inputs each were generated from GGDs $p_\alpha[X|\phi]$ with $\alpha = p/q$ and Eq. (5.77) for the corresponding conditions. The range coder used the empirical distribution gained from the inputs.

Fig. 5.18 describes their trade-offs between the coding time and compression efficiency. We took an average of encoding and decoding time over 100 trials and divided by N and the entropy to get the coding time. While the range coder showed very high compression efficiency because it was a VV code, the proposed coder performed faster in every case, especially when using smaller p and q. For example, the proposed coder showed about six times faster than the range coder for Gaussian-distributed integers, maintaining the increase in relative redundancy around 2.6%, which was much lower than that of the unary coder. Although the coding time in absolute has less meaning because it depends on hardware and compilers, its relative value gives us some reasonable estimate of complexity compared with other conditions based on the same hardware and compiler. Compared to the unary coder, the compression efficiency of the proposed code was much higher where the coding time was almost comparable at $(p, q) = (1, 2)$ and $(2, 1)$.

We also compared the relative redundancy with that of other codes—run-length GR (RLGR) code proposed in [84], Huffman codes, and the exp-Golomb code—using GGDs $p_\alpha[X|\phi]$ of $\alpha = 1/2,\ 2/3,\ 3/4,\ 4/3,\ 3/2$, and $2/1$ in a wider range of scales $\phi = 1.00, 1.01, 1.02, \cdots, 4.00$, for the same length of input sequences. RLGR code is the conventional method aimed at its use in compressing integers belonging to GGDs and includes the conventional GR code as its particular case. This time, for the proposed code, parameters (p, q) were given to match each shape parameter α and the Rice parameter was estimated using Eq. (5.74) for each ϕ.

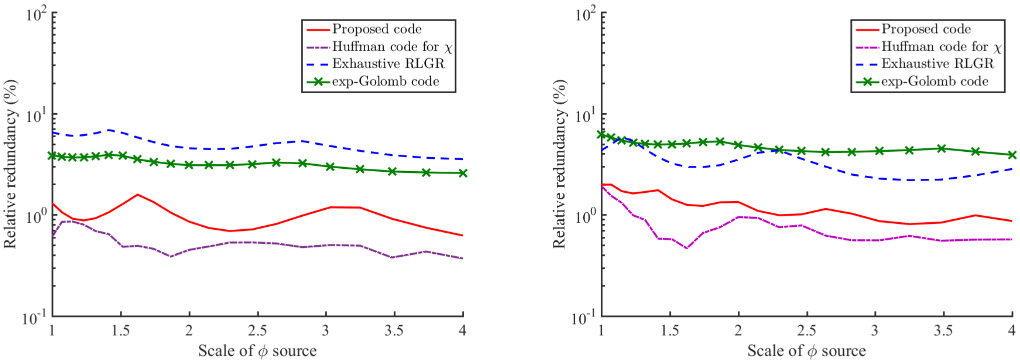
Fig. 5.19 shows the comparison results. The coding parameters for RLGR code and exp-Golomb code were searched exhaustively for each distribution. For Huffman codes, we took a histogram of the existing integers $\mathcal{X}$ in the input integer sequence for each distribution and made codebooks optimized for the histograms of the Cartesian products $\mathcal{X}^2$ of the integers. It can be seen from the curve that the proposed code outperformed the conventional exp-Golomb and RLGR codes. Additionally, this result also proves its higher performance than that of the conventional GR code because the RLGR code includes the conventional GR code representing by one of the coding parameters.

5.4 Conclusion of Chapter 5

This chapter presented two shaping methods related to the code modeling of speech and audio signals. Both of them shape the distributions to which the inputs belong into the ones optimally encoded by low-complexity structured codes, Golomb and Rice codes.

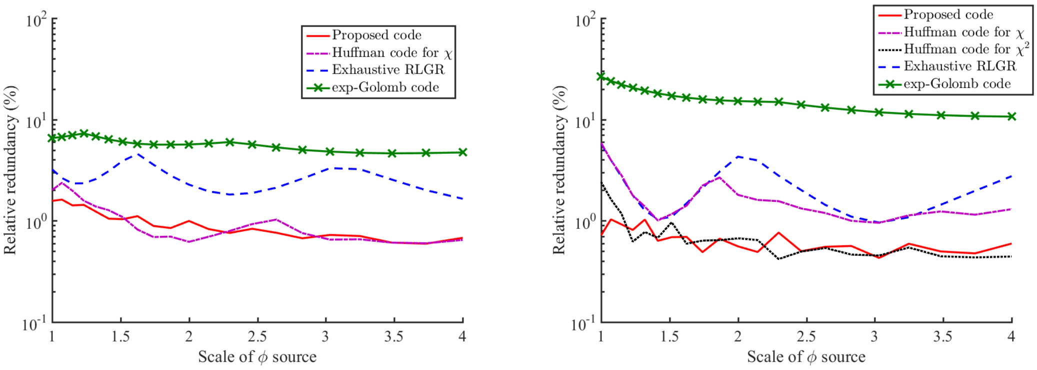
Bit inverting shapes the scale of the input distribution, realizing XDGR and XDG codes, which are optimal for integers belonging to sparse Laplacian distributions and have simple algorithms derived from the idea of AIFV codes. Shape control by integer nesting and splitting can approximately change generalized Gaussian distributions into Laplacian, which enables us to efficiently compress a wide range of sources belonging to distributions of the exponential family. Both methods were investigated in theory and in simulations that showed promising characteristics.

These proposed methods, in combination, are expected to fully make use of the assumptions of the speech and audio defined by the signal modeling even under conditions of low computational costs, which will be evaluated in the later chapter.



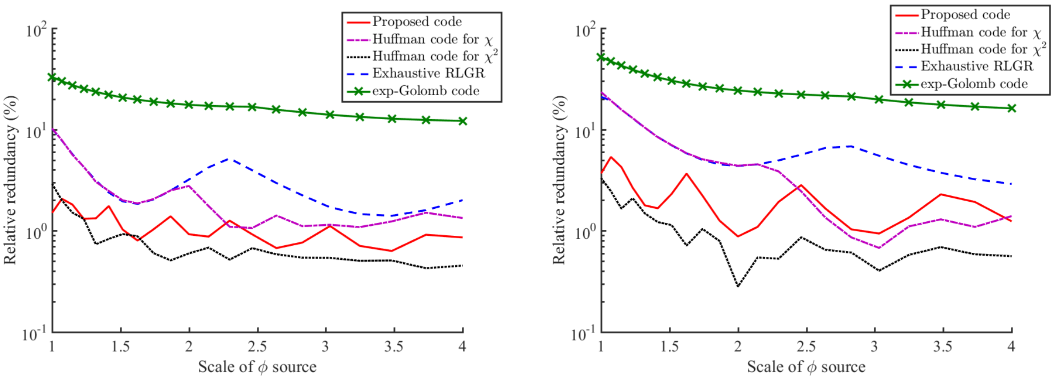
(a) $p = 1, q = 2$

(b) $p = 2, q = 3$



(c) $p = 3, q = 4$

(d) $p = 4, q = 3$



(e) $p = 3, q = 2$

(f) $p = 2, q = 1$

Fig. 5.19. Relative redundancy of each code. Red solid, purple long-dashed short-dashed, black dotted, blue dashed, and green crossed lines, respectively indicate the proposed code using each p and q for integer nesting and splitting, Huffman code, RLGR code, and exp-Golomb code.

Chapter 6

Application for low-delay and low-bit-rate mobile communication

6.1 Introduction of this chapter

The following chapters discuss the applications realized by the proposed shaping methods. Although each of the applications differs in the implementations, their fundamental structures are the same, which explained in Chap. 2 introducing the three types of modeling. In this chapter, we show an application on mobile communication, one of the greatest demands for speech-and-audio communication.

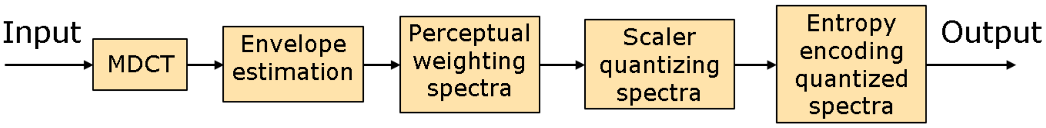
In mobile communications, the limitations of the radio resources and the computation resources of mobile phones require their codecs to operate at low bit rates with low computational costs. Although the networks are becoming broadband these days, speech communications, the communications that must not be interrupted, should be under the priority control, which still requires a low traffic volume. Also, the mutual communications demand low delay, low enough delay not to disturb the conversation. For the reasons above, the codecs for mobile communications have to find an exquisite trade-off of low bit rate, low delay, and low computational complexity while remaining high sound quality.

To achieve speech communications with high quality dealing with the trade-offs, codecs such as ITU-T G.729 [17] and 3GPP AMR [2,3] have overcome the difficulties by strongly assuming that the inputs are speech signals: They operate only on a narrow frequency band, utilizing statistical properties of speech and efficiently represent the spectral envelopes of voiced signals by LP. Consequently, the quality of the coding particularly increases when the input is a speech signal, and these codecs have made significant contributions to the prosperity of the speech communications. However, specializing in speech conversely leads to the weakness against the other signals, especially against signals having far different bandwidth and statistics from speech, and these kinds of signals significantly degrade the quality of the codecs. Realize communication with higher quality, establishing more comfortable conversations with higher presence requires codecs to deal with broader frequency bands and also with audio signals, music, for example, as well as the speech signals.

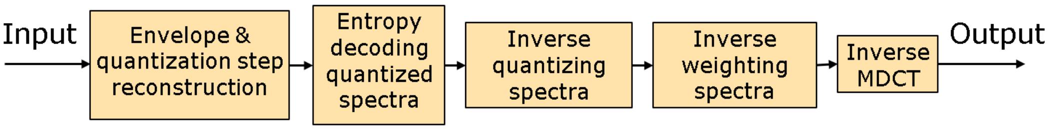
These days, as in 3GPP AMR-WB+ and MPEG-D USAC [18–21], a coding paradigm has been developed for coding signals including both speech and the other audio. These codecs switch their modes depending on the characteristics of the input signals. In most cases, speech signals are coded in the time domain, with the assumptions stated above. The other audio signals, especially signals with a variety of sounds, are coded in the frequency domain because it is easier to apply perception modeling, such as the model of masking effects. Although this paradigm succeeds in making up for the shortcomings of the time-domain coding, the frequency-domain coding needs relatively longer delay because of the need for buffering inputs in order to take sufficient frequency resolution for showing its merits of energy concentration and the perception modeling. For the sake of this fact, AMR-WB+ and USAC both permit more than 100 ms of delay only for their algorithms, while 100 ms is a detectable delay in conversations [25]. Therefore, to utilize the paradigm of unified speech and audio coding for mobile communications, it is necessary for the frequency-domain coding to represent the inputs efficiently in lower-delay conditions.

Recently, a promising frequency-domain coding scheme called transform coded excitation (TCX) has been invented [32], which is adopted into USAC. This coding scheme represents the input signals in the form of MDCT coefficients and applies a simple perceptual model that can be easily extended to low-delay conditions. Therefore, in this chapter, we consider designing a TCX-based frequency-domain codec that shows high quality at low-bit-rate and low-delay conditions, expecting to combine with the time-domain codecs.

We first describe the basic structure of the TCX based on the conventional scheme and then discuss enhancing it, focusing respectively on signal and code modeling, which are



(a) Encoder



(b) Decoder

Fig. 6.1. Breif flowchart of the baseline TCX.

the main factors related to its compression efficiency.

6.2 Structure of the baseline coding scheme

The TCX, the recently-proposed frequency-domain coding scheme based on MDCT, provides us with a simple perceptual model that can be easily realized by scalar quantization and entropy coding. To design a low-delay frequency-domain codec, we first implemented from scratch a baseline codec by extracting the essence of the TCX. After this, unless otherwise noted, we refer this TCX-based codec to TCX.

Figs. 1(a) (a) and (b) show the flowchart of the encoder and the decoder, respectively. First, the encoder transforms the $2N$-length input signal $\{x_n\}_{n=0}^{2N-1}$ into frequency spectra $\{X_k\}_{k=0}^{N-1}$ by MDCT after windowing. Then, the encoder extracts their envelope $\{H_k\}_{k=0}^{N-1}$ using LP from the spectra as explained in Chap. 2. The order of the LP is set to 16, as is usually done in similar conditions. The coefficients are coded in 20 bits using the vector quantization similar to [96] in the form of LSP. After the extraction, the encoder scalar quantizes the target spectra with the quantization noise perceptually controlled by multiplying perceptual weights $\{P_k\}_{k=0}^{N-1}$ derived from the smoothed envelope $\{\tilde{H}_k\}_{k=0}^{N-1}$ in Eq. (2.9). The quantized spectra are entropy coded by GR code applying a different Rice parameter for each frequency k in accordance with the value of the envelope of the weighted spectrum $P_k H_k$ as

$$\begin{aligned}
R_k &= \max([\log_2((\ln 2)\sigma P_k H_k/s)], 0) \\
&= \max([\log_2(P_k H_k) + \log_2((\ln 2)\sigma/s)], 0) \\
&\equiv \max([\log_2(P_k H_k) + \bar{R}], 0).
\end{aligned} \tag{6.1}$$

where σ is a prediction gain, and s indicates the quantization step. $\sigma P_k H_k/s$ in the equation indicates the envelope of the quantized spectra. On the other hand, $\bar{R}$ represents a base value of the Rice parameters in the frame and can be assumed to be a constant value that depends on the bit rate because the step size is decided to satisfy the given bit rate and the gain σ, which approximately indicates the standard deviation of the spectra, is related to the entropy of the target. Allocating the Rice parameters as in Eq. (6.1) results in fewer bits required for GR coding, as long as the envelope represents the values of the spectra with sufficient accuracy, compared with the case of using the same Rice parameter for all the frequencies.

Additionally, to enhance the performance of the entropy coding, the encoder codes the runs of zeros in the variable-length part of the code, i.e., the runs of zero-valued quotients of the target values divided by 2^{R_k}. This run-length code contains a fixed-length part and a variable-length part as GR code, and the information about how many bits to use for the fixed-length part is sent by a few bits at each frame. The step size of the scalar quantization, quantized in decibels, is found by a bisection search so that the result of the entropy coding meets the given bit rate.

The codec uses a window function shown in Fig. 6.2, which is based on a half-sine window and simply modified for its use in the low delay condition:

$$w_k = \begin{cases} 0 & (\text{if } 0 \le k < L) \\ \sin\left(\frac{\pi(k-L+0.5)}{2M}\right) & (\text{if } L \le k < M+L) \\ 1 & (\text{if } M+L \le k < N+L) \\ \sin\left(\frac{\pi(k-3L+0.5)}{2M}\right) & (\text{if } N+L \le k < 2N-L) \\ 0 & (\text{if } 2N-L \le k < 2N) \end{cases} \tag{6.2}$$

where $N = 320$, for the frame length, $M = 192$, for the lookahead, and $L = (N-M)/2$. Thus, with 20-ms overlapping, the encoder buffers 20 ms for each frame with 12 ms of lookahead, totally making 32 ms of algorithmic delay. This delay is much shorter compared

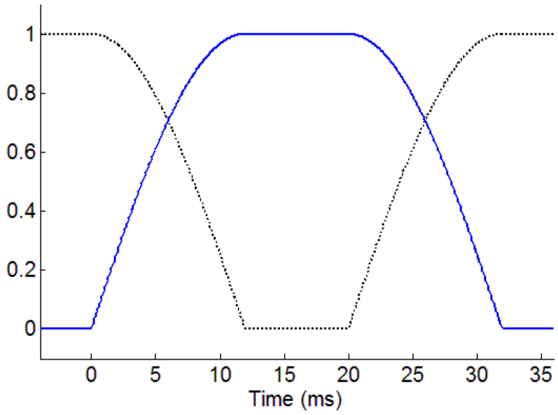


Fig. 6.2. Window function used in the codecs with half overlap. 640 samples at 16-kHz sampling rate.

to AMR-WB+, which takes the frame size of 128 ms with 16 ms of lookahead, totally permitting 144 ms of algorithmic delay [97].

Summarizing the above, this TCX is implemented by

- LP for the signal modeling
- smoothed envelope for the perception modeling
- scalar quantization and GR code for the code modeling

where the signal and code modelings are the main focus of this chapter for employing the proposed shaping methods.

In the following sections, the objective and subjective evaluations of the TCX with the proposed shaping methods are discussed, aiming at monaural signals at around 16-64 kbps at a 16-kHz sampling rate, which is the favored target for this use case. As mentioned in the previous section, we are expecting the codec to be used in a hybrid codec of time-domain and frequency-domain coding. Therefore, the evaluations of the quality were conducted for musical signals, for which the frequency-domain codec is typically responsible.

6.3 Enhancement in signal modeling

6.3.1 Evaluation of frequency warping

Integration into baseline codec

To integrate into the baseline TCX the frequency warping method presented in Chap. 3, which warps the frequency resolution of LP-based envelopes, we have only to insert matrix operations in the envelope estimation and the perceptual weighting. The modified envelope estimation works by applying the frequency-warping and inverse-warping matrix W and U, optimized in advance by Eqs. (3.14) and (3.15), respectively to the power spectra before estimating their envelopes and to the envelopes given as the conventional LP. As also mentioned in Chap. 3, the smoothed envelopes for the perceptual weighting are calculated from Eq. (2.9) followed by applying the inverse-warping matrix U with LP coefficients transformed in the LSP domain by the conversion matrix in Eq. (3.19).

For the W and U used in the codec, their initial values $W^{(0)}$ and $U^{(0)}$ are respectively given from $N \times N$ matrices of squared sinc interpolation:

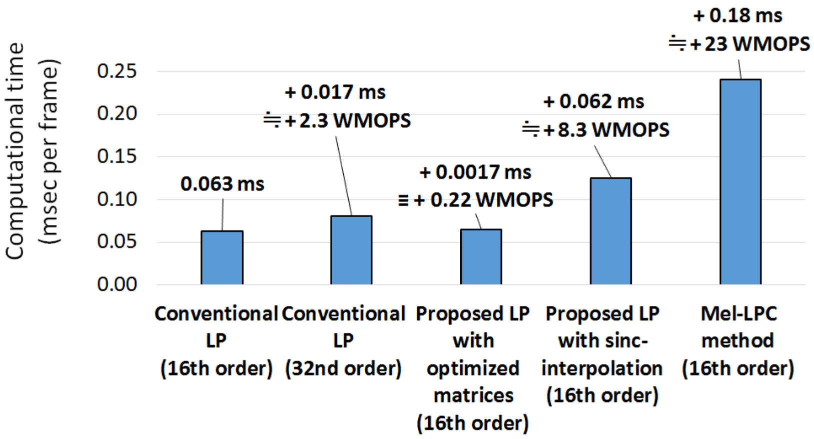
$$S_{mn} = \mathrm{sinc}^2(f(m) - n) \tag{6.3}$$

$$T_{mn} = \mathrm{sinc}^2(g(m) - n) \tag{6.4}$$

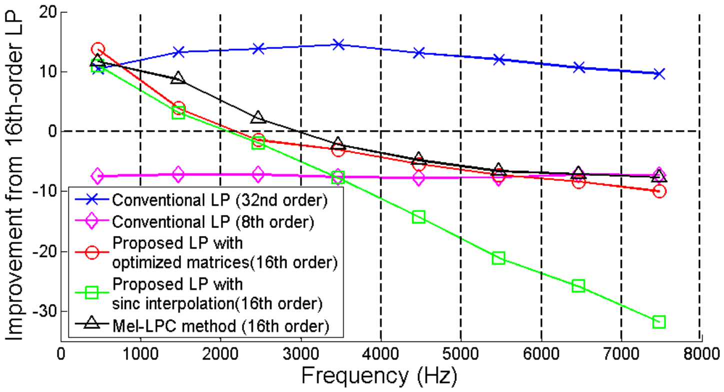
with the sinc function $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$, the mapping function $f(k')$ in Eq. (3.3) and its inverse function:

$$g(k) = \frac{N}{\ln(f_s/(2f_0) + 1)} \ln\left(\frac{f_s}{2f_0 N} k + 1\right), \tag{6.5}$$

which maps from the index k $(0 \leq k \leq N-1)$ of the frequency sampled uniformly on the linear frequency scale to the corresponding index on the Mel-frequency scale. Note that $g(0) = 0$, $g(N) = N$. Zero values are substituted for the elements in $\{S_{mn}\}$ and $\{T_{mn}\}$ except in the column with the maximum value of each row and the columns around it, resulting in sparse matrices having at most seven non-zero elements in each row. These sparse matrices become the initial values $W^{(0)}$ and $U^{(0)}$, resulting in the warping and the inverse warping matrices that cost only $7N$ operations of multiplication each to operate.



(a) Average calculation time per frame required to estimate parameters and calculate envelopes by each method. The difference of WMOPS [98] from the 16th-order LP was estimated on the basis of the computational time of warping-matrix operations.



(b) Average improvement from the conventional 16th-order LP in Itakura–Saito divergence of the envelopes extracted by each method.

Fig. 6.3. Comparison of the envelope models based on calculation time and accuracy.

Performance in calculation and accuracy

Before we go to the codec performance, let us focus on the effects of the proposed shaping method itself. We evaluated the proposed resolution-warped model of spectral envelopes at the 16th order based on its computational costs and the accuracy of the represented envelopes. The compared methods were the conventional LP, the higher order of the LP, the proposed LP, which was explained in the previous section, with optimized sparse matrices for warping, the proposed LP with sinc interpolation, which respectively used non-optimized full matrices in Eq. (6.3) and Eq. (6.4) for warping and inverse warping, and the Mel-LPC method in [52].

First, we measured the calculation times to estimate the parameters for each model from the power spectra of test data and calculate the envelopes. For a fair comparison, we used for each method C program files in the speech signal processing toolkit (SPTK) ver. 3.6 [99]. The proposed LP was implemented by just adding the matrix operations in the conventional LP. Since the programs in SPTK work only for the frame length with the power of two, the computational times were calculated for 256 samples per frame at a 16-kHz sampling rate, although the TCX processes the target signals at 320 samples per frame. We tested 31,301 frames, each containing 256 samples of audio signals at a 16-kHz sampling rate, and used a machine with a 2.40-GHz Intel (R) Core (TM) i7-3630QM CPU, 16.0 GB memory and 64 bit windows 7 OS.

The results are shown in Fig. 3(a). The value written for each bar is the additional time spent from the 16th-order conventional LP. Sparsely designed matrices lowered the computational cost for the warping to the level of minor complexity compared with the other methods.

Secondly, with the same test data, the accuracy of the envelopes extracted by each method was measured. We evaluated it by the IS divergence since the envelopes are based on the IS divergence from the power spectra, as mentioned in the previous section. For evenly divided frequency bands $\{B_i\}_{i=1}^{10}$, we calculated the improvements of the accuracy from the 16th-order conventional LP defined with IS divergence $D_{\mathrm{IS}}(x|y) = y/x - \ln(y/x) - 1$ as

$$\mathrm{Improvement}_i \equiv \sum_{k \in B_i} \left(D_{\mathrm{IS}}(\hat{H}_k^2 | X_k) - D_{\mathrm{IS}}(H_k^2 | X_k) \right) \tag{6.6}$$

where $\{\hat{H}_k\}$, $\{H_k\}$ and $\{X_k\}$ are the envelope of 16th-order conventional LP, the envelope of each method, and the power spectra of the test data.

The average improvements are plotted in Fig. 3(b). The envelopes of the proposed LP with the optimized matrices, despite their significantly low costs for calculation, showed almost the same accuracy as the Mel-LPC method, approximately equivalent to the 32nd-

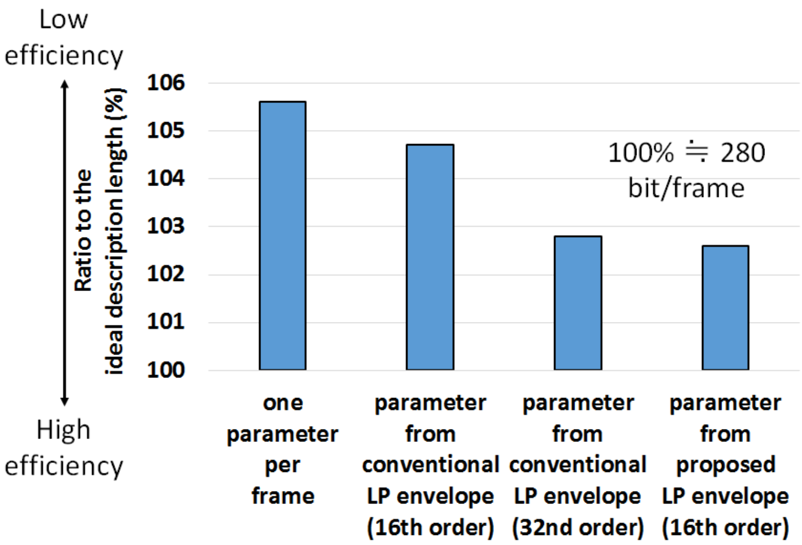


Fig. 6.4. Bit-per-frame comparison of GR coding with each Rice-parameter decision. The average bit length at each frame. Frame length was 20 ms at 16-kHz sampling rate.

order conventional LP in the lower band and the 8th-order counterpart in the higher band. Meanwhile, the envelopes that used non-optimized full matrices appeared to be degraded in the higher band, which was caused by the irreversible warping and inverse warping operations. This result proves that the proposed optimization provided the warping operations with sufficient consistency.

Performance in Golomb–Rice coding

To evaluate the effects of the resolution-warped envelopes on the TCX, we compared the performance in GR coding. For this comparison, quantized spectra were prepared by coding the test data mentioned above with the TCX integrated with the frequency warping at 16 kbps. Then, we coded the quantized spectra with GR code using the Rice parameters decided by the following methods. The first method is to allocate the optimal Rice parameter for each frequency, which provides the ideal description length. The second method is to use the same Rice parameter for every frequency in each frame that makes the shortest code length. The third is to calculate the Rice parameters by Eq. (6.1) respectively with the envelopes $\{H_k\}$ extracted by

1. the 16th-order conventional LP

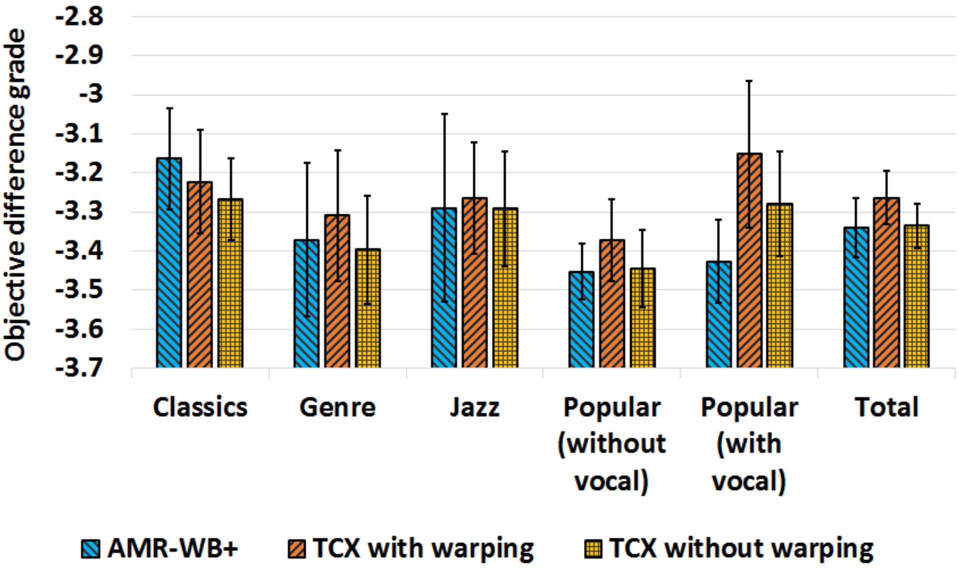


Fig. 6.5. Database-wise average PEAQ scores. Error bars indicate the 95% confidence intervals.

2. the 32th-order conventional LP
3. the 16th-order proposed LP with optimized matrices

without the quantization for the envelopes.

Fig. 6.4 presents the average frame-wise ratio of each bit length to the ideal description length. The result suggests the validity of changing Rice parameters for each bin following the envelope. The resolution warping of the envelopes enhances the efficiency even more, with equivalence to the conventional LP of twice the order. This enhancement occurred because most of the values of the target spectra had a more dynamic range in the lower band, which required a more precise allocation of Rice parameters than in the higher band.

Performance in objective sound quality

Objective quality was measured by the method of the perceptual evaluation of audio quality (PEAQ) in AFsp [100]. The test data were randomly selected from the four databases in the RWC Music Database [101]: Ten items each from the Classical Music, Jazz Music, and Music Genre Databases; twenty items from the Popular Music Database,

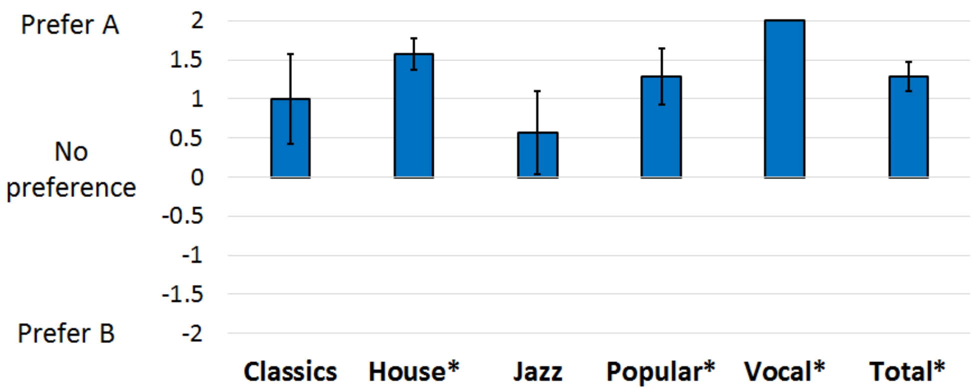


Fig. 6.6. Item-wise scores of AB test, where A is the TCX with warping and B is the TCX without warping. Average and standard error. Asterisk indicates the existence of significant difference at 5% in a t-test.

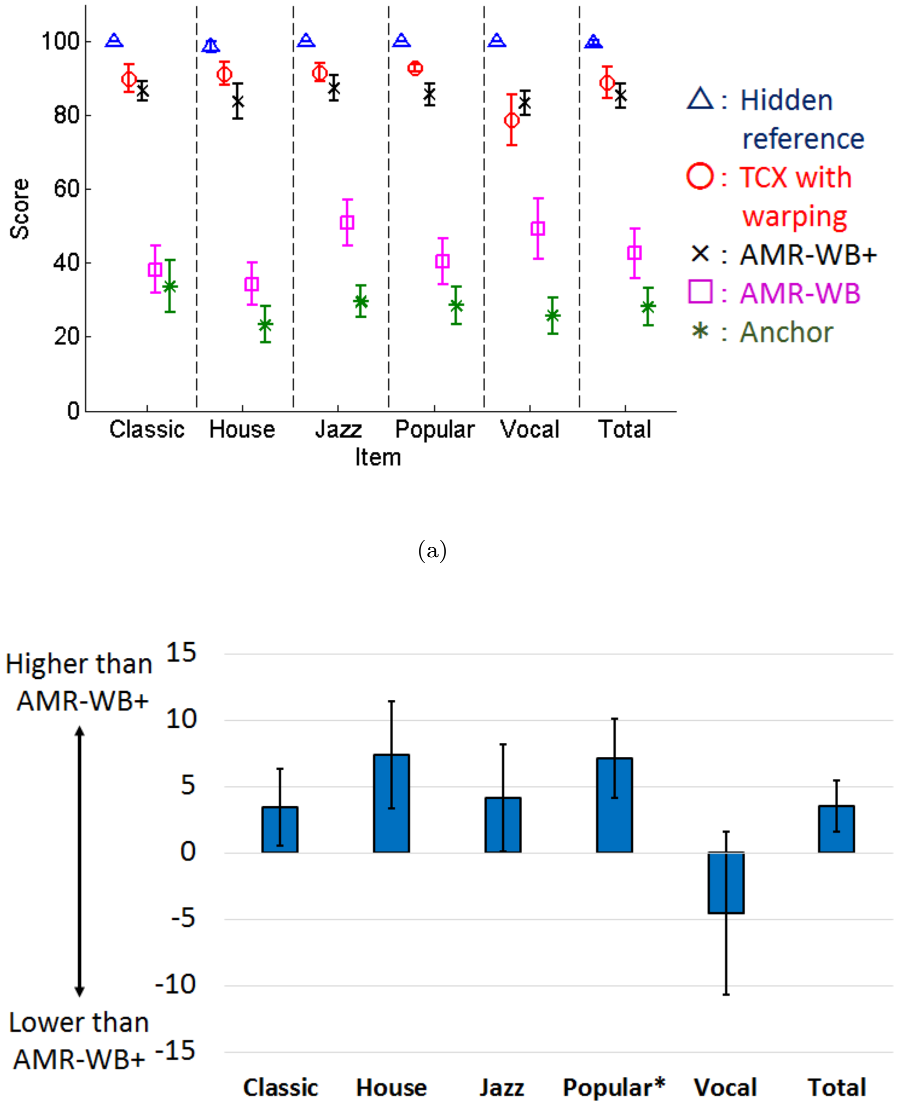
ten without vocals, and ten with vocals. All items contained ten seconds of signals down-sampled into 16 kHz.

Fig. 6.5 displays the results. Although the shortage of the frame length caused disadvantage against AMR-WB+, especially in classical music items, which often contain stationary signals, the TCX with the frequency warping showed higher performance on average owing to the enhancement provided by the resolution-warped representation.

Performance in subjective sound quality

The codecs were also evaluated subjectively. Five audio items in the RWC Music Database, ten seconds each down-sampled into 16 kHz, were used for the evaluations. The test items were labeled as follows: "Classic," for a violin piece from the Classical Music Database, "House," for a guitar piece from the Music Genre Database, "Jazz," for a trumpet piece from the Jazz Music Database, "Popular," for a synthesizer piece from the Popular Music Database, and "Vocal," for a female vocal piece from the Popular Music Database.

Two experiments were conducted with seven participants, all engaged in audio signal processing research. First, the effects of the resolution warped representation were evaluated by an informal AB test. Participants blindly listened to the items, each coded in 16 kbps by respective TCX with and without the frequency warping. After listening to the



(a)

(b)

Fig. 6.7. Item-wise MUSHRA scores. Average and standard error. (a) Absolute scores. (b) Relative scores of the TCX compared to AMR-WB+. Asterisk indicates the existence of significant difference at 5% in a paired t-test.

references, the participant chose the method they preferred by scoring from -2 to 2 points for each item.

The results in Fig. 6.6 prove the resolution-warped representation of the envelope enhances the subjective quality of the codec.

Secondly, the TCX with the warping was compared with the other codecs by ITU-R BS.1534-1 Multiple stimuli with hidden reference and anchor (MUSHRA) [102]. The items were each coded at 16 kbps by the TCX with the warping, AMR-WB+, and AMR-WB. This time, the participants were blindly provided with these coded items with the references and 3.5-kHz band-limited anchors, and they graded the quality of each codec from 0 to 100 points.

The scores are presented in Fig. 6.7. The results for AMR-WB confirm the weakness of time-domain coding of audio signals other than speech and the need for frequency-domain coding. As stated in the introduction, AMR-WB+ uses frequency-domain coding in addition to time-domain coding, which made the quality higher than AMR-WB. Meanwhile, due to the efficient representation, the TCX with frequency warping showed comparable performance, although it has much lower delay compared with AMR-WB+.

6.3.2 Evaluation of amplitude warping

Integration into baseline codec

We can easily introduce the amplitude-warping method and the GGD assumptions into the baseline codec if we modify its entropy coding part to use arithmetic coding. Therefore, we also implemented arithmetic coding mode for the baseline codec using the range-coder-based arithmetic coding [46].

The arithmetic coding can allocate real-valued bits by encoding the whole spectra in the frame at once and is adopted in, for example, JPEG [103] and USAC. Based on the conventional assumptions as in the case of GR coding, the target of the arithmetic coding is written as

$$Y_k = P_k X_k / s, \tag{6.7}$$

ignoring the rounding operation, and its envelope can be represented by

$$\hat{H}_k = \sigma P_k H_k / s = \frac{\sigma P_k}{s|1 + \sum_{n=1}^{p} a_n e^{-\mathrm{j}\frac{\pi k}{N} n}|}. \tag{6.8}$$

The encoder allocates the bit length for each $\{Y_k\}_{k=0}^{N-1}$ in accordance with its envelope

$\{\hat{H}_k\}_{k=0}^{N-1}$ as

$$b_k = \begin{cases} -\log_2(\tilde{q}_1(\frac{1}{2} \mid \hat{H}_k)) & (\text{if } Y_k = 0) \\ \\ -\log_2(\tilde{q}_1(|Y_k| + 0.5 \mid \hat{H}_k) - \tilde{q}_1(|Y_k| - 0.5 \mid \hat{H}_k)) + 1 & (\text{otherwise}) \end{cases} \tag{6.9}$$

where $\tilde{q}_1(Y_k \mid \hat{H}_k)$ is a cumulative frequency function of Laplacian defined as

$$\begin{aligned} \tilde{q}_1(Y_k \mid \hat{H}_k) &= \int_0^{Y_k} 2\tilde{p}_1(x \mid \hat{H}_k)\,\mathrm{d}x \\ &= \int_0^{Y_k} \frac{1}{\hat{H}_k} \exp\left(-\left|\frac{x}{\hat{H}_k}\right|\right) \mathrm{d}x, \end{aligned} \tag{6.10}$$

a one-sided cumulative distribution function of the Laplacian with its scale depending on the value of the envelope, and +1 in Eq. (6.9) is for the sign of each MDCT coefficient.

The entropy coding assuming GGD can be implemented in this entropy coding mode. Thus, we consider here to apply to the above arithmetic coding the proposed amplitude-warping method combined with this assumption. First, the relationship is shown between the code length of the arithmetic coding, which assumes the GGD, and the negative log-likelihood of Eq. (3.31).

If we assume the scale of each frequency k as $\{\phi_k P_k/s\}_{k=0}^{N-1}$ for the GGD with fixed weights $\{P_k\}_{k=0}^{N-1}$ and quantization step s, the bit allocation for $\{Y_k(= P_k X_k/s)\}_{k=0}^{N-1}$ of the arithmetic coding will be

$$b_k = \begin{cases} -\log_2\left(\tilde{q}_\alpha\left(\frac{1}{2} \mid \phi_k P_k/s\right)\right) & (\text{if } Y_k = 0) \\ \\ -\log_2\left(\tilde{q}_\alpha\left(|Y_k| + \frac{1}{2} \mid \phi_k P_k/s\right) - \tilde{q}_\alpha\left(|Y_k| - \frac{1}{2} \mid \phi_k P_k/s\right)\right) + 1 & \\ & (\text{otherwise}) \end{cases} \tag{6.11}$$

where $\tilde{q}_\alpha(Y_k \mid \phi)$ is defined as

$$\begin{aligned} \tilde{q}_\alpha\left(Y_k \mid\mid \phi\right) &= \int_0^{Y_k} 2\tilde{p}_\alpha\left(x \mid\mid \phi\right)\,\mathrm{d}x \qquad (6.12) \\ &= \frac{1}{\Gamma(1/\alpha)} \Gamma_{\text{inc}}\left(\frac{1}{\alpha}, \left|\frac{B(\alpha)Y_k}{\phi}\right|^\alpha\right) \end{aligned}$$

with $B(\alpha)$ of Eq. (3.29) and the incomplete gamma function:

$$\Gamma_{\text{inc}}(x, y) = \int_0^y e^{-t} t^{x-1} \mathrm{d}t. \tag{6.13}$$

In the case of $Y_k \neq 0$, using the fact that $\tilde{q}_\alpha(Y_k \mid \phi)$ depends only on the ratio for Y_k and ϕ, the bit allocation b_k of Eq. (6.11) can be rewritten as

$$\begin{aligned} b_k &= -\log_2 \left(\tilde{q}_\alpha \left(\frac{s|Y_k|}{P_k} + \frac{s}{2P_k} \;\middle|\; \phi_k \right) - \tilde{q}_\alpha \left(\frac{s|Y_k|}{P_k} - \frac{s}{2P_k} \;\middle|\; \phi_k \right) \right) + 1 \\ &= -\log_2 \left(\tilde{q}_\alpha \left(|X_k| + \frac{s}{2P_k} \;\middle|\; \phi_k \right) - \tilde{q}_\alpha \left(|X_k| - \frac{s}{2P_k} \;\middle|\; \phi_k \right) \right) + 1 \end{aligned} \tag{6.14}$$

and when the quantization is fine enough, or $s \ll 1/P_k$, the first-order approximation of b_k for s becomes

$$\begin{aligned} b_k &\simeq -\log_2 \left(\frac{s}{P_k} \frac{\partial}{\partial x} \tilde{q}_\alpha(x \mid \phi_k) \Big|_{x=|X_k|} \right) + 1 \\ &= -\log_2 \tilde{p}_\alpha(|X_k| \mid \phi_k) + \log_2 P_k/s, \end{aligned} \tag{6.15}$$

and so as in the case of $Y_k = 0$. This results in the total bit length:

$$\begin{aligned} \sum_{k=0}^{N-1} b_k &\simeq \sum_{k=0}^{N-1} [-\log_2 \tilde{p}_\alpha(|X_k| \mid \phi_k) + \log_2 P_k/s] \\ &\propto \mathcal{L}_\alpha^{\mathrm{LH}}(\{X_k\} \mid \{\phi_k\}), \end{aligned} \tag{6.16}$$

which approximately proportionate to the negative log-likelihood presented in Eq. (3.31). In the end, the minimization of the total bit length comes down, when the quantization is sufficiently fine, to the minimization of the negative log-likelihood, which can be optimized by the proposed PAPSE by using Eq. (3.33) for $\{\phi_k\}_{k=0}^{N-1}$. Moreover, for the approximation of the perceptual weights $\{P_k\}_{k=0}^{N-1}$, we can use the smoothed version of the envelope in Eq. (3.36) as

$$P_k = \left| 1 + \sum_{n=1}^{p} \gamma^n a_n e^{-\mathrm{j}\frac{\pi n k}{N}} \right|^{2/\alpha} \tag{6.17}$$

because the envelope in Eq. (3.36) shows the rough shape of the magnitude spectra as well as the conventional envelope does.

Summarizing the discussion above, the algorithm to encode the frequency spectra goes as follows:

1. Calculate the model parameters σ^2 and $\{a_n\}_{n=1}^{p}$ in Eq. (3.36) by the Levinson–Durbin algorithm using for $\{R_n\}_{n=0}^{p}$ in Eq. (2.5) the inverse cosine transform of the α-th power of the magnitude spectra $\{|X_k|\}_{k=0}^{N-1}$ as in Eq. (3.35) instead of using the one of the power (Method of PAPSE).

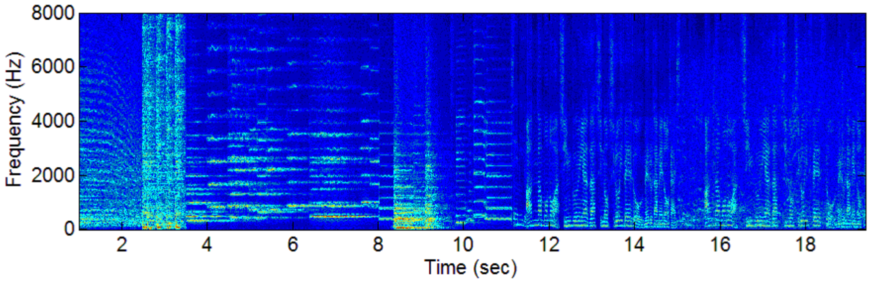
2. Approximate the perceptual weights by Eq. (6.17) instead of the ones in Eq. (2.9).
3. Scaler quantize the weighted spectra $\{P_k X_k\}_{k=0}^{N-1}$ by step size s.
4. Allocate bit length to the quantized spectra by arithmetic coding in accordance with Eq. (6.11) with Eq. (3.33) for the scale $\{\phi_k\}_{k=0}^{N-1}$ instead of allocating in accordance with Eq. (6.9).

These minor changes from the TCX provide the codec with the globally-optimized compression of the scaler-quantized spectra belonging to the GGD of shape parameter α the scale of which is constrained by the model of Eq. (3.33). It should be noted that, since the proposed PAPSE is based on the all-pole model as the conventional LP, the model parameters $\{a_n\}_{n=1}^{p}$ can be quantized by the conventional quantization algorithms for the LP coefficients such as in [96], the vector quantization of the coefficients in the form of LSP.

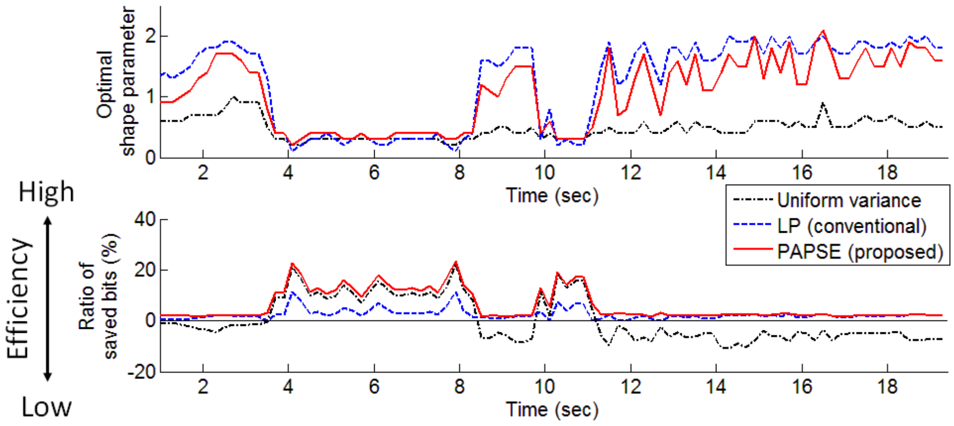
The model parameters $\{a_n\}_{n=1}^{p}$ defined in Eq. (3.33) and the quantization step s can be quantized respectively by the method of [96] and by simple logarithmic scaler quantization. For the other model parameter σ^2, we assume here its ratio to the quantization step s to be a constant depending only on the conditions of the bit rate and the shape parameter α because it is strongly related to the entropy of the generalized-Gaussian source. In other words, the σ^2 is 0-bit quantized. The decision of this constant is performed by taking the α-power mean of $\sigma^{2/\alpha}/s$ given from the training data.

Optimal shape parameter in each acoustic situation

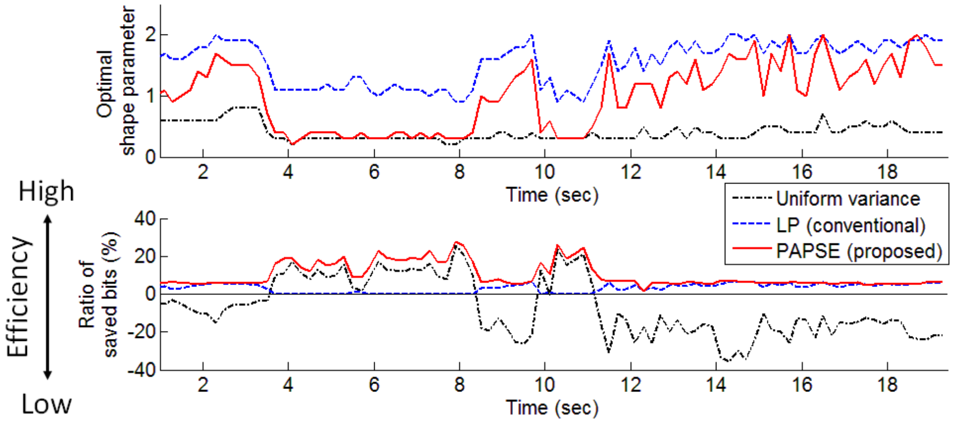
Before we go on to the evaluation of the codec integration, the assumption of the GGD and the use of the proposed PAPSE should be justified. Therefore, we first compared the compression efficiency for fixed quantized spectra to see how the distribution of the frequency spectra varies depending on the acoustic properties of the signal and how the dependence fluctuates due to the difference in the model of the scales: the uniform-scale model, which is the assumption in [59,60], the conventional LP model in Eq. (3.39), which is used in the baseline TCX, and the proposed PAPSE model in Eq. (3.33) presented in the previous chapter. The fixed spectra were prepared by quantizing the MDCT coefficients of a test signal using the TCX at 64 kbps and 16 kbps and were compressed by the arithmetic coding assuming the GGD. The following three methods calculated the scales for the GGD:



(a)



(b) Result for 64 kbps.



(c) Result for 16 kbps.

Fig. 6.8. Spectrogram of the test signal (a), transitions of optimal parameter α (the upper half of (b) and (c)), and ratios of the saved bits from the conventional setting (the lower half of (b) and (c)).

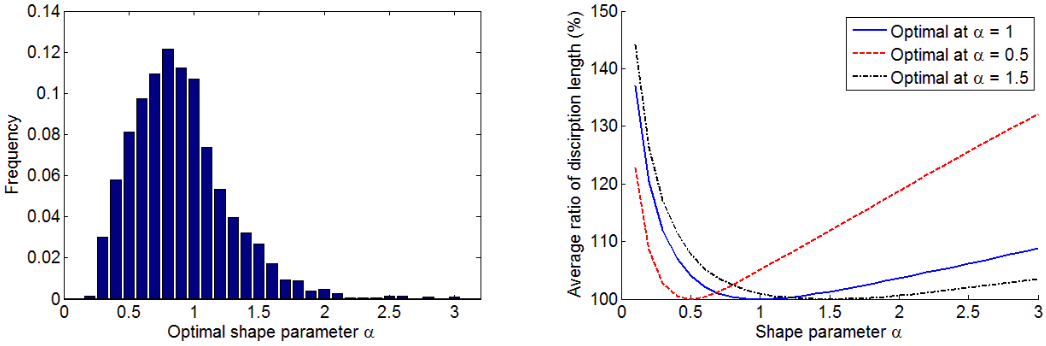
- For the uniform-scale model, the scales were given by the zeroth-order of the proposed PAPSE, which means using the maximum-likelihood uniform scales for each frame;
- for the conventional LP model, the scales were given by Eq. (3.39) with the 16th-

order LP coefficients for each frame;

- for the proposed PAPSE model, the scales were given by Eq. (3.33) with the model parameters obtained by the 16th-order of the proposed PAPSE for each frame.

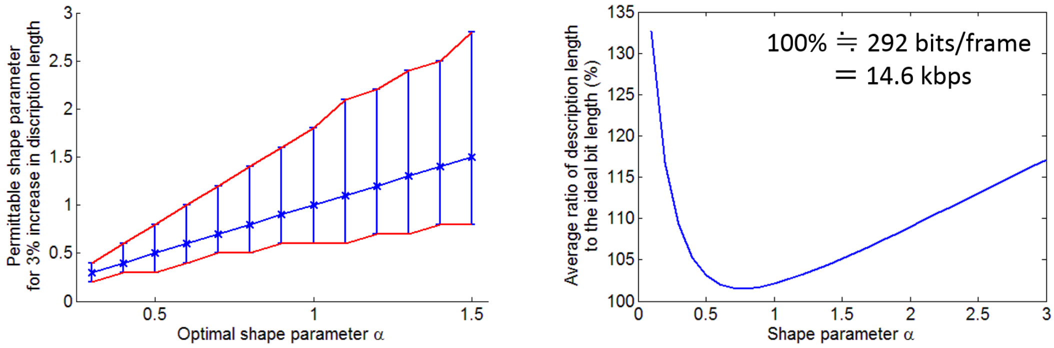
For each case, without quantizing the model parameters, we compressed the fixed spectra by using for the shape parameter α of the GGD from $\alpha = 0.1$ to $\alpha = 3$ in increments of 0.1 and found the α for each consecutive ten frames that makes the minimum description length in bits. Note that dealing with each consecutive ten frames was intended just for avoiding meaningless outliers. In all the three cases, to fairly compare the effects of the models, we also fixed the perceptual weights $\{P_k\}_{k=0}^{N-1}$ defined in Eq. (6.7) to the one used to quantize the fixed spectra. The test signal, the spectrogram of which is shown in Fig. 6.8 (a), was composed of, in order to see different acoustic cases, four seconds each of a popular music (synthesizer), a classical music (violin), a jazz music (trumpet), a male speech (clean), and the same male speech with pink noise (signal-to-noise ratio was 10 dB). We used the musical signals in the RWC music database [101] and the speech signals in the ATR speech database [104] all down-sampled into a 16-kHz sampling rate.

The results are described in Fig. 6.8 (b) and (c). The upper halves of the figures show the transition of the optimal shape parameters α for each consecutive ten frames in each case, and the lower halves show the ratio of the saved bits compared with the arithmetic coding assuming the Laplacian with its scales modeled by the conventional LP, or the conventional setting. Although the result of the optimal α for the uniform-scale model agreed with the ones of the previous studies in [59, 60], it is obvious that the optimal α reflected the acoustic properties much more when assuming non-uniform scales for the GGDs: higher values of α for noisy or temporal sounds and lower values for clean or tonal sounds. In the sense of compression efficiency, there were some situations that the tuning of α makes an improvement from the conventional setting even if the scales were assumed to be uniform, but this uniform assumption failed in the other situations and worsened the efficiency. Besides, comparing the conventional and proposed PAPSE models, the difference in the transition of the optimal α was affected by the bit rate more in the case of the conventional LP model. Furthermore, in the case of the proposed PAPSE model, the compression efficiency was much improved, especially in tonal sounds, while the improvement was much smaller in the case of the conventional LP model despite the



(a) (b)

Fig. 6.9. Histogram of the optimal shape parameter α (a), transitions of average bit length by changing α from the optimal one (b). Total 24720 frames of MDCT coefficients were tested.



(a) (b)

Fig. 6.10. Ranges of α that makes the bit length increase less then 3 % from the optimal length (a), and average bit length when using the same α for all the data (b). 100% in (b) was given by the average bit length when using the optimal α for each consecutive ten frames. Total 24720 frames of MDCT coefficients were tested.

tuning of α.

Effects of shape parameter on compression efficiency

From now on, we focus on the musical signals because the TCX is mainly expected to deal with these kinds of input, on which it is hard, in most cases, for the time-domain codec to exert its advantages. The second experiment was aimed at discussing the following factors:

- Which shape parameter α the optimal α tends to be;
- how much bits the coding consumes when the α used in the coding is not the optimal one;
- how the optimal α ranges when we permit to some extent the increase in the bit length;
- which α is the best if we have to choose one α to code all the spectra.

With the motivation to answer these questions, we prepared quantized MDCT coefficients from ten seconds each of fifty musical signals by the same method in the previous experiment at 16 kbps and compressed the quantized coefficients by the arithmetic coding for the GGD with the proposed PAPSE using the shape parameter α from $\alpha = 0.1$ to $\alpha = 3$ in increments of 0.1. The fifty musical signals were randomly selected from the four databases in the RWC Music Database [101]: Ten items each from the Classical Music, Jazz Music, and Music Genre Databases; twenty items from the Popular Music Database, ten without vocals, and ten with vocals. Ten seconds of signals were extracted from each item and down-sampled into 16 kHz.

Fig. 6.9 (a) is the histogram of the optimal shape parameter α that attained the shortest bit length for each consecutive ten frames. As in the histogram, the optimal α tends to be near $\alpha = 1$, which does not conflict with the previous assumptions for the distribution of frequency spectra. The curves in Fig. 6.9 (b) were obtained by gathering the frames that attained the shortest bit length when $\alpha = 0.5$, $\alpha = 1$, and $\alpha = 1.5$, respectively, and plotting the average of the bit length for each case using the other α. It can be seen that all three curves have asymmetric forms, steeper in the left side, and the steepness gets weaker as the optimal α becomes high. Related to the steepness, the ranges of α are also shown in Fig. 6.10 (a) that makes at most three percent of the increase in the bit length from the case of the optimal α. In other words, each range in Fig. 6.10 (a) corresponds to the range of α that makes each of the curves in Fig. 6.9 (b) lower than 103 %. We can see that the range gets larger as the optimal α becomes high, that is to say, the spectra belonging to the GGD with higher α are more tolerant, in terms of a bit length, to the difference in α used in the coding. Fig. 6.10 (b) is, as to speak, an integration of Figs. 6.9 (a), (b), and Fig. 6.10 (c). This curve denoted a plot of the bit length when we used a single α for compressing all the test spectra compared with the optimal bit length when we were able to tune the α for each consecutive ten frames. The asymmetry of the curve remains, and the suitable α for compressing all the test spectra seems to be between 0.5 and 1 with $\alpha = 0.8$ being optimum for this test.

Since the TCX, as mentioned above, chooses quantization steps that make the compression results satisfy the given bit rate, it can be said that the higher compression efficiency the arithmetic coding has, the more precise the encoder is allowed to quantize the input frequency spectra. In order to check if the quantization gets more precise by the proposed PAPSE, we also compared the SNR of the quantized spectra in Eq. (6.7), which is considerably related to the sound quality, fixing the bit rate at 16 kbps by finding the scale s with a bisection search. The same test data as above were compressed by the arithmetic coding with one shape parameter α for every frame. To make a fair comparison, we used, in every condition, for the perceptual weights $\{P_k\}_{k=0}^{N-1}$ the one approximated from the conventional LP as in Eq. (2.9).

The results are plotted in Fig. 6.11. The peak of the curve for the proposed PAPSE seems to be in between 0.5 and 1 with $\alpha = 0.7$ being optimum for this test, which almost agrees with the previous experiment. On the other hand, with the conventional LP, tuning the shape parameter α was less effective, and this result agrees with the last experiment. The SNR in the conditions near $\alpha = 2$ was close to one of the conventional LP. That is because the proposed PAPSE identifies with the conventional LP when $\alpha = 2$, namely when assuming Gaussian for spectra.

Performance in sound quality

This time, in order to confirm the benefit of both the proposed PAPSE and the GGD, we applied the coding scheme to the baseline TCX, which used range-coder-based arithmetic coding instead of GR coding, at 16 kbps and evaluated objectively and subjectively the sound quality of musical signals.

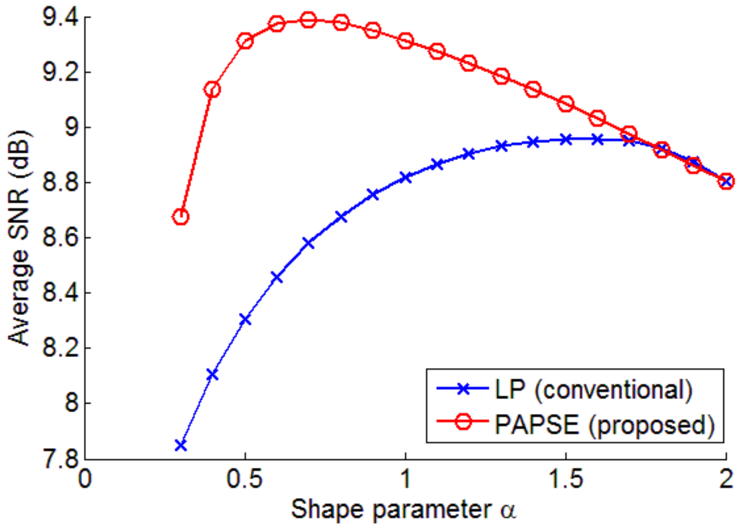


Fig. 6.11. SNR of the quantized spectra by the shape parameter α for the arithmetic coding using each LP. Total 24720 frames of MDCT coefficients were tested at 16 kbps.

For the objective evaluation, the method of PEAQ in AFsp [100] was used. The test data for this objective evaluation were the same as the last experiment, and the following conditions of TCX were compared:

- Assuming the Laplacian (the GGD with shape parameter $\alpha = 1$), modeling its scales with the conventional LP model, and approximating the perceptual weight by $\{P_k\}_{k=0}^{N-1}$ in Eq. (2.9), hereinafter called the conventional setting;
- assuming the GGD with $\alpha = 0.7$, modeling its scales with the proposed PAPSE model, and approximating the perceptual weight by $\{P_k\}_{k=0}^{N-1}$ in Eq. (6.17), hereinafter called the proposed setting.

One may seem it unfair to use shape parameter α that proved in the last experiment to be the best parameter for representing the test data. However, our objective for this experiment is to confirm the existence of the distribution that makes better performance than the Laplacian and to see the effects of the optimization in that case. Therefore, we are not insisting on the superiority of the shape parameter $\alpha = 0.7$.

Fig. 6.12 displays the database-wise PEAQ scores for the all conditions. The performance of the TCX appeared to be enhanced by introducing the GGD and the proposed

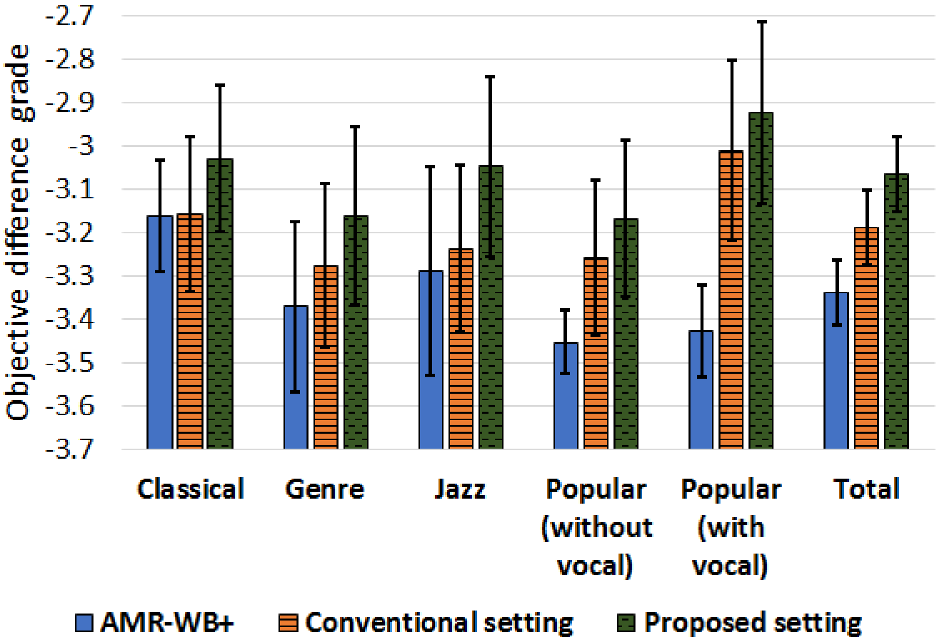
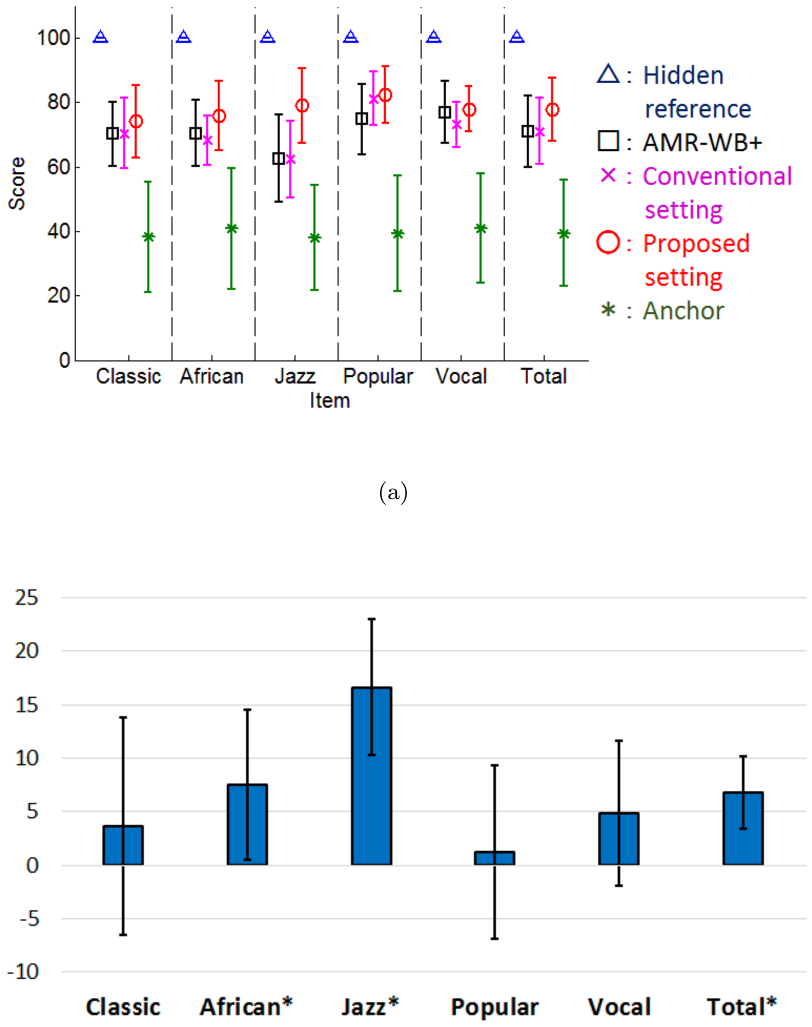


Fig. 6.12. Database-wise PEAQ scores. Error bars indicate the 95 % confidence intervals.

PAPSE. Additionally, the paired t-test proved there was a significant difference at five percent, totally and for each database, between the conventional and proposed setting.

For the subjective evaluation, the three conditions were compared by MUSHRA. Five audio items in the RWC Music Database, ten seconds each down-sampled into 16 kHz, were used for the evaluations. The test items were labeled as follows: "Classic," for a harpsichord piece from the Classical Music Database, "African," for a guitar piece from the Music Genre Database, "Jazz," for a piano piece from the Jazz Music Database, "Popular," for a synthesizer piece from the Popular Music Database, and "Vocal," for a female vocal piece from the Popular Music Database. Eight participants, all engaged in audio signal processing research, were blindly provided with the coded items with the references and 3.5-kHz band-limited anchors, and they graded the quality of each codec from 0 to 100 points.

The absolute scores and the scores increased by the proposed setting are presented in Fig. 6.13. The codec with the conventional setting, which showed almost the same performance as AMR-WB+, was enhanced in average by introducing the GGD and the proposed PAPSE optimized to it, with three out of five items and total score having a significant difference at five percent.



(a)

(b)

Fig. 6.13. Item-wise MUSHRA scores. Average and 95 % confidential interval of t-distribution. (a) Absolute scores. (b) Relative scores of the proposed setting compared to the conventional setting. Asterisk indicates the existence of significant difference at 5 % in a paired t-test.

Discussions

The results in the previous experiments insist on the importance of the following three factors: selecting the distributions for the target spectra, assuming non-uniform scales over frequencies, and parameterizing the scales, or spectral envelope, optimally. The most interesting part of the results is that only in the case of parameterizing the scales by the conventional LP the transition of optimal shape parameters greatly fluctuated comparing Fig. 6.8 (b) and (c). Since the conditions in Fig. 6.8 (b) and (c) differed solely in bit rate, in other words, in the step size of the quantization, one may expect only the scale, instead of the shape, of the distribution to have been influenced by the quantization. However, this expectation merely holds as long as the scales of the target is appropriately estimated. We conjecture that the scales, parameterized improperly by the conventional LP in the sense of likelihood, required the shape parameter α to compensate for the scales in order to describe the target spectra. Therefore, tuning the shapes of the assumed distributions may be vain efforts unless we parameterize the scales correctly. Surprisingly, the proposed PAPSE, which approximately attains the optimal parameterization of the scales for the GGDs, seems to have preserved its approximation in the low-bit-rate situation such as 16 kbps at a 16-kHz sampling rate, and significantly reduced the description length by properly parameterizing the scales.

The remained discussions are about computational complexity and the effects of the slight modifications from the conventional scale model in Eq. (3.39) to the proposed scale model in Eq. (3.33). Owing to the slight modifications in the model, which were aimed to make the optimization solved by the Levinson–Durbin algorithm, the additional costs of computation for the proposed PAPSE only appear in the powering operations in Eq. (3.33), (3.35), and (6.17) for the encoder. According to BASOP [98], the total increase of the computational costs in the encoder, in the condition of the codec stated in the evaluations, amount to approximately 1.2 WMOPS, about only four percent of the increase when the α is not an integer. It should be noted that this computation still has room for improvement. If we permit the decrease in the precision of the envelope, we can reduce the costs by calculating the envelope only in several frequency intervals. Moreover, when $\alpha = 1$, the process of the proposed PAPSE corresponds to the one in [40], and it is known that the computational complexity gets reduced from the case of applying the conventional LP in that case. By the way, we have to emphasize that the proposed PAPSE is *not* guaranteed to always make the bit length shorter compared with the case of the conventional LP because the model is slightly changed to simplify the algorithm. Nevertheless, the results of all the experiments proved that the difference in the model had a little influence on the superiority of the proposed PAPSE model.

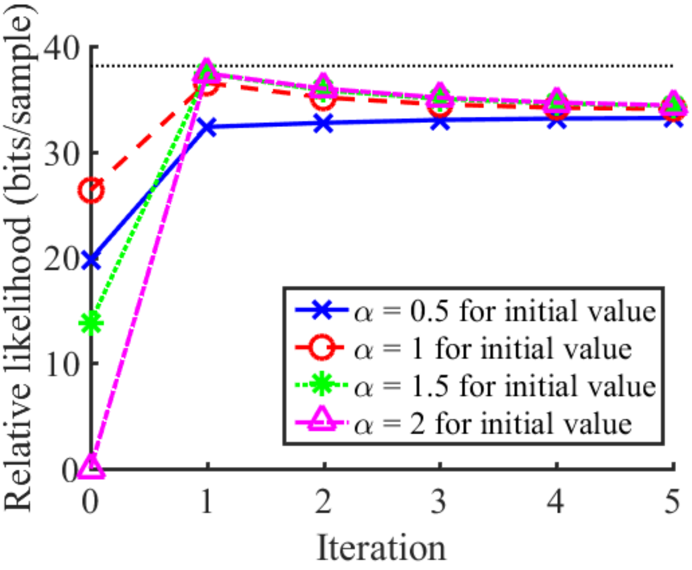


Fig. 6.14. Relative average log-likelihood compared to LP (bits/sample) by each iteration and initial values. Black dotted horizontal line shows the limit found by an exhaustive search. 17232 frames of 16-kHz audio signals (about 6 minutes) were tested.

6.3.3 Further extensions

Frame-wise adaptive amplitude warping

Using the shape parameter estimation presented in Sec. 3.3.3, PAPSE has the potential to enhance the baseline TCX further: Estimating frame-by-frame α makes higher likelihood, using different distributions in the arithmetic coding under the estimated α and transmitting α to the decoder by allocating bits to it. Here, we evaluate its possibility of enhancement.

At first, to check the characteristics of the proposed method, its effect on likelihood, initial dependence, convergence, the required number of iteration, we estimated shape parameters for some speech and audio signals and calculated their likelihood. For the shape parameter estimation step in the proposed method, we prepared for the α candidates $\alpha = 0.1$ to $\alpha = 3$ in increments of 0.1.

Fig. 6.14 plots the average log-likelihood. The likelihood was calculated as relative

Table 6.1. Relative average log-likelihood compared to LP (bits/sample) with 17232 frames of 16-kHz audio signals (about 6 minutes) tested

Without PAPSE	With PAPSE		
Simple moment-based estimation	Fixed ($\alpha = 0.7$)	Proposed estimation	Exhaustive search
-60.1111	27.9153	37.4118	38.1298

log-likelihood in bits compared by its counterpart of LP:

$$L = \frac{1}{NM} \sum_{k,l} \log_2 \frac{\tilde{p}_{\alpha_l^*}(|X_{k,l}| \mid H_{\alpha_l^*,k,l})}{\tilde{p}_2(|X_{k,l}| \,||\, H_{k,l})} \tag{6.18}$$

where l, M, and α_l^* respectively stands for the frame number, the total frames, and the estimated α for the l-th frame. $H_{\alpha_l^*,k,l}$ and $H_{k,l}$ were given by 16-th-order PAPSE and 16-th-order LP, respectively. It can be seen that there is some convergence, and it depends on the initial values. Practically, it seems sufficient to use a large value for the initial α and the results of the first iteration.

Table 6.1 compares other estimation methods with the proposed method (the result of the first iteration with initial value $\alpha = 2$). The difference between the proposed method and the exhaustive search was less than 1 bit, and changing the shape parameter frame by frame showed a higher likelihood compared to the fixed PAPSE, which calculated the log-likelihood with a constant α for every frame. A simple moment-based method, which estimated α using the moment-based method without normalizing the spectra by its scale $H_{k,\alpha}$, seems to have given inaccurate estimates resulting in the decrease of the likelihood from LP.

Next, we compared the transition of the estimated shape parameters. The test signal was composed of four seconds each of a popular music (synthesizer), a classical music (violin), a jazz music (trumpet), a male speech (clean), and the same male speech with pink noise (signal-to-noise ratio was 10 dB), which was used in [105]. Fig. 6.15 depicts the result. The shape parameters estimated by the proposed method roughly tracked the optimal shape parameters found by the exhaustive search, showing the correspondence between the shape parameter and some acoustic features. On the other hand, the simple moment-based method estimated continuously as around $\alpha = 0.5$, which revealed to be inaccurate by the previous experiment.

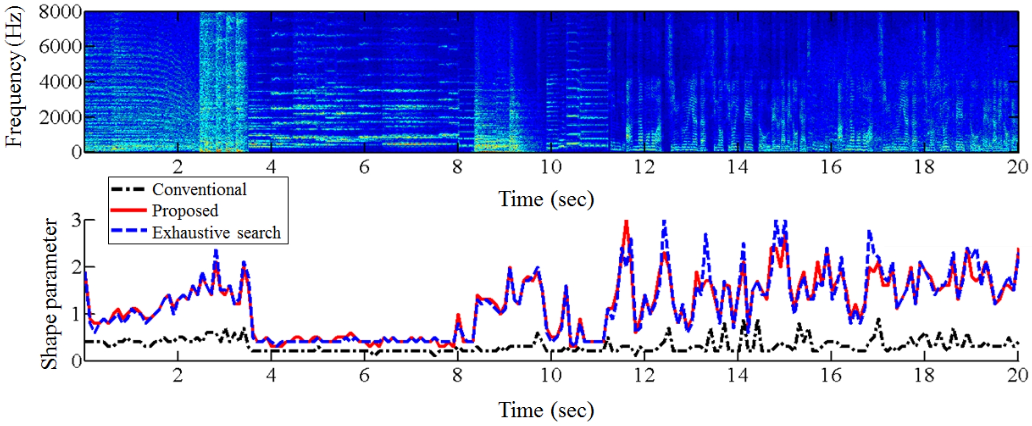


Fig. 6.15. Spectrogram and its shape parameters estimated for every five frames. Simple moment-based method for conventional method (black chained line), proposed method (red solid line), and exhaustive search (blue slashed line).

At 16 kbps, we used two conditions of codec integration for the comparison:

1. (Constant PAPSE) Using GGD of $\alpha = 0.7$ for arithmetic coding with its scales represented by fixed PAPSE of 16-th order, of which coefficients $\{a_n\}_{n=1}^{16}$ were vector quantized in the form of LSP as in [96] with 20 bits;
2. (Adaptive PAPSE) adding shape parameter estimation part before PAPSE. The arithmetic coding used different GGD for each frame based on the estimated α. The variance was also represented by 16-th-order PAPSE. Its coefficients were quantized by the method stated above with different codebooks for each α. The α was represented by 1 bit, as mentioned below.

For the estimated shape parameter, we used the results from the first iteration of the proposed shape parameter estimation with an initial value of $\alpha = 1$. The quantization of the estimated shape parameter α was designed heuristically by trial and error, with the optimal α expected to change smoothly in audio signals: Representing the quantized shape parameter in τ-th frame $\hat{\alpha_\tau}$ with fourth-order moving average by 1 bit, in other words, selecting $\hat{\beta}_\tau$ that satisfies

$$\hat{\alpha_\tau} = \mu^{-1}(\hat{\beta}_\tau + 0.7\hat{\beta}_{\tau-1} + 0.6\hat{\beta}_{\tau-2} + 0.5\hat{\beta}_{\tau-3} + 0.4\hat{\beta}_{\tau-4}) \tag{6.19}$$

where $\mu(\alpha)$ indicates the μ-law algorithm of ITU-T G. 711 [1]. The values for $\hat{\beta}_\tau$ was defined to make $\hat{\alpha_\tau}$ be in [0.5 1]. The other bit allocations were evenly set.

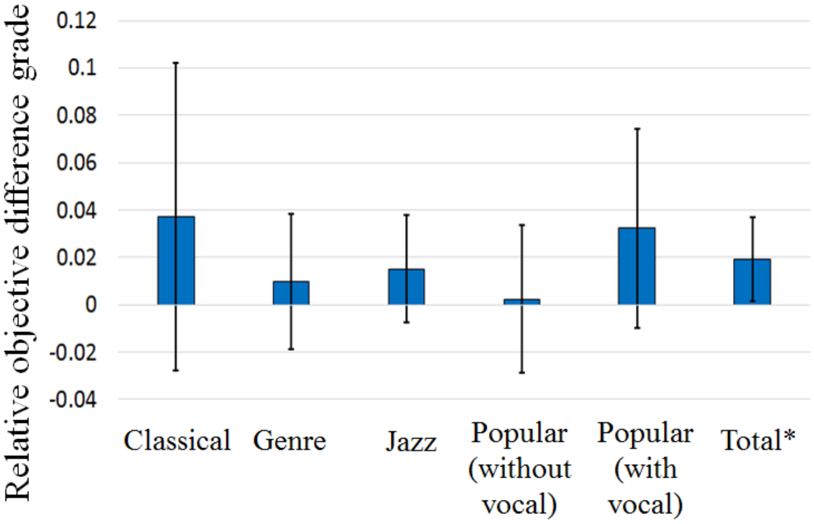


Fig. 6.16. Database-wise relative objective difference grades by PEAQ compared to the TCX with constant PAPSE. Average and 95 % confidence intervals. Asterisk indicates there was a significant difference at 5 % in a paired t-test.

The objective sound quality of the reconstructed signals in PEAQ, graded from -4 to 0 points, was calculated by McGill University's AFsp PQevalAudio [100]. Fifty items were randomly selected from the four databases in the RWC Music Database [101]: Ten items each from the Classical Music, Jazz Music, and Music Genre Databases; twenty items from the Popular Music Database, ten without vocals and ten with vocals. Ten seconds of signals were extracted from each item and down-sampled into 16 kHz.

The relative scores of the TCX with adaptive PAPSE compared to the one with constant PAPSE are shown in Fig. 6.16. The shape parameter estimation made the average objective quality higher, giving a significant difference in the total score. As for the complexity, the additional computational costs for applying the shape parameter estimation in this condition were about 0.6 WMOPS [98], about 2 % of the total costs of the TCX-based coder.

To evaluate whether the difference in objective quality shown by the previous experiment is actually audible, an informal subjective evaluation was held. Five audio items in the RWC Music Database, ten seconds each down-sampled into 16 kHz, were respectively coded in the two conditions, presented to seven participants with the references and 3.5-kHz band-limited anchors, and graded from 0 to 100 points, as is done in ITU-R

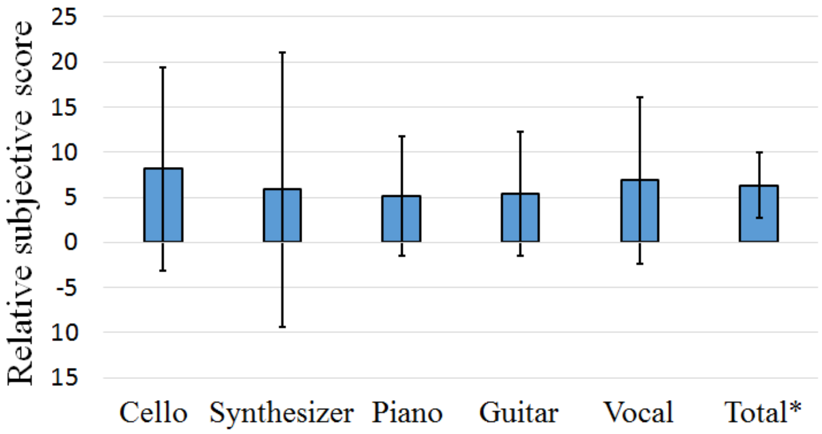


Fig. 6.17. Item-wise relative subjective scores by MUSHRA compared to the TCX with constant PAPSE. Average and 95 % confidence intervals. Asterisk indicates there was a significant difference at 5 % in a paired t-test.

BS.1534-1 MUSHRA [102]. The test items were labeled as follows: "Cello," for a cello piece from the Classical Music Database, "Synthesizer," for a synthesizer piece from the Music Genre Database, "Piano," for a piano piece from the Jazz Music Database, "Guitar," for a Guitar piece from the Popular Music Database, and "Vocal," for a female vocal piece from the Popular Music Database.

Fig. 6.17 describes the item-wise relative scores of the TCX with adaptive PAPSE compared to the one with constant PAPSE. It can be seen that the shape parameter estimation enhanced the average subjective quality with a significant difference in the total score, which resembles the results of the objective evaluation.

Combining two types of warping

The ideas proposed in Chap. 3 have placed high importance on the use of the all-pole-based model and the Levinson–Durbin algorithm for representing the spectral envelopes because they lead to algorithms with low computational complexity and enable the codec to quantize the model parameters efficiently by LSP representation and to guarantee the stability of the parameters simply. Therefore, in order not to spoil these merits, the integration of frequency and amplitude warping also has to be implemented based on the all-pole model and the Levinson–Durbin algorithm.

Heuristic combination of the envelope models Eqs. (3.2) and (3.36) leads to the following envelope and scale models, respectively:

$$H_{\alpha,k} = \left(\sum_{i=0}^{N-1} U_{ki} \left| 1 + \sum_{n=1}^{p} a_n e^{-\mathrm{j}\frac{\pi i}{N} n} \right|^{-2} \right)^{\frac{1}{\alpha}}, \tag{6.20}$$

a powered all-pole model of which frequency resolution is warped by the inverse warping matrix U, and

$$\phi_k = \alpha^{1/\alpha} B(\alpha) \sigma^{2/\alpha} H_{\alpha,k} \tag{6.21}$$

as in Eq. (3.33). This envelope model becomes equivalent to Eq. (3.2) when $\alpha = 2$ and to Eq. (3.36) when U is identity. According to the motivation stated above and the proposed methods, it is natural to think that the extraction of the envelopes based on Eq. (6.20) can be done by regarding warped and powered spectra as power spectra, in other words using

$$R_n = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} W_{ki} |X_i|^{\alpha} \cos\left(\frac{\pi n k}{N}\right) \tag{6.22}$$

for auto-correlation function in the Levinson–Durbin algorithm.

However, it should be noted that for the sake of the frequency-warping operation, the criterion of this extraction, or the estimation of $\{a_n\}_{n=1}^{p}$, gets different from the actual bit length of the entropy coding; the criterion becomes the bit length added up over the *warped*-frequency axis. On the other hand, in regard to the other model parameter σ^2, the power of the prediction residual , which is usually given from the Levinson–Durbin algorithm and is used in calculating the variance $\{\phi_k\}_{k=0}^{N-1}$ as in Eq. (6.21), the parameter can be easily optimized on the criterion of the actual bit length. Provided that the variance of the spectra is given by Eq. (6.21) with the extracted envelope $\{H_{\alpha,k}\}_{k=0}^{N-1}$, the negative log-likelihood of the spectra $\{X_k\}_{k=0}^{N-1}$, assumed to be generalized-Gaussian distributed, becomes

$$\begin{aligned}
&\mathcal{L}_{\alpha}^{\mathrm{LH}}(\{X_k\} \mid \{\phi_k\}) \\
&= \sum_{k=0}^{N-1} (\log_2 e) \left[\left| \frac{B(\alpha) X_k}{\phi_k} \right|^{\alpha} + \ln \phi_k - \ln A(\alpha) \right] \\
&= \frac{1}{\ln 2} \sum_{k} \left[\frac{1}{\alpha \sigma^2} \left| \frac{X_k}{H_{\alpha,k}} \right|^{\alpha} + \frac{1}{\alpha} \ln \sigma^2 + \ln \alpha^{1/\alpha} B(\alpha) H_{\alpha,k} / A(\alpha) \right].
\end{aligned} \tag{6.23}$$

This likelihood is convex for σ^2 so that the optimal σ^2 that minimizes $\mathcal{L}_{\alpha}^{\mathrm{LH}}(\{X_k\} \mid \{\phi_k\})$

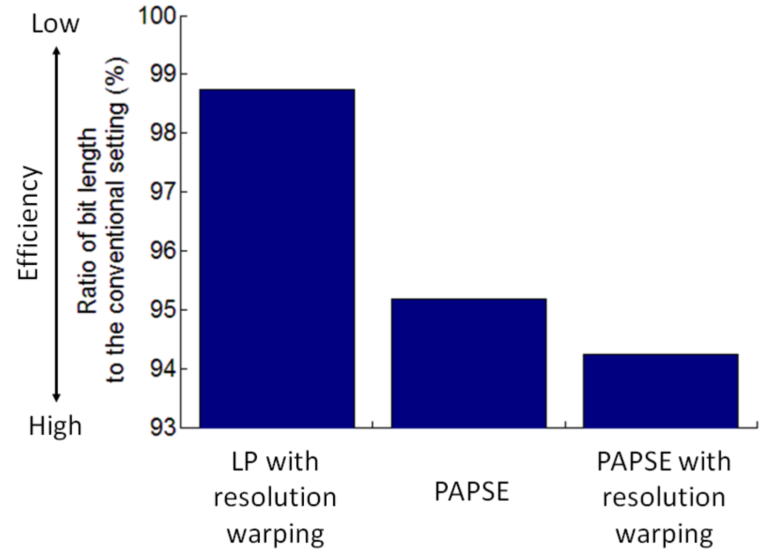


Fig. 6.18. Comparison of the average bit length by each representation of envelopes. 100 % indicates the average bit length in the case of the conventional setting. Total 24720 frames of MDCT coefficients were tested.

is given by

$$\frac{\partial}{\partial \sigma^2}\mathcal{L}_\alpha^{\mathrm{LH}}(\{X_k\} \mid \{\phi_k\}) = \frac{1}{\alpha \ln 2}\sum_k \left[-\frac{1}{(\sigma^2)^2}\left|\frac{X_k}{H_{\alpha,k}}\right|^\alpha + \frac{1}{\sigma^2}\right] = 0$$
$$\iff \sigma^2 = \frac{1}{N}\sum_k \left|\frac{X_k}{H_{\alpha,k}}\right|^\alpha, \tag{6.24}$$

which is the mean of α-th-powered residual $\{|X_k/H_{\alpha,k}|^\alpha\}_{k=0}^{N-1}$.

We evaluated the model by compressing the quantized spectra using the range-coder-based arithmetic coding, as in the last section. For this evaluation, ten seconds each from fifty musical signals in the RWC Music Database [101] were down-sampled into 16 kHz and tested. The quantized spectra were prepared by quantizing the MDCT coefficients of a test signal using the baseline TCX at 16 kbps and were compressed by the arithmetic coding with the following conditions, and the code lengths were compared:

- Extracting envelope by the conventional LP with the Laplacian assumed for the arithmetic coding (the conventional setting);
- extracting envelope by the resolution-warped model in Eq. (3.2) switching by 1 bit the warping matrices between the identity ones and the logarithmic ones presented in Chap. 3 with the Laplacian assumed for the arithmetic coding;

- extracting envelope by the PAPSE model in Eq. (3.36) with the GGD of $\alpha = 0.7$ assumed for the arithmetic coding;
- extracting envelope by the integrated model in Eq. (6.20) switching by 1 bit the warping matrices between the identity ones and the logarithmic ones with the GGD of $\alpha = 0.7$ assumed for the arithmetic coding.

To make a fair comparison, we used, in every condition, for the perceptual weights $\{P_k\}_{k=0}^{N-1}$ the one approximated from the conventional LP as in Eq. (2.9). A closed-loop search decided the switching of the warping.

The result is depicted in Fig. 6.18. The bit length of each condition is represented by the ratio to one of the conventional setting. It can be seen that both ideas of the frequency- and amplitude- warping contributed to enhancing the efficiency of the entropy coding, and the integration of the ideas made further enhancement. Since the perceptual weights $\{P_k\}_{k=0}^{N-1}$ were the same in every condition, higher compression efficiency can be expected to make higher sound quality at the same bit rate allowing us to quantize spectra more precisely.

6.4 Enhancement in code modeling

6.4.1 Integration into baseline codec

In the previous section, the integration of the proposed PAPSE was realized by using arithmetic coding in the entropy coding part of the TCX because GR coding cannot optimally encode under the assumption of GGDs. It is expected that introducing the methods for the code modeling proposed in Chap. 5 will enable us to integrate PAPSE into the GR-coding-based scheme. However, the conventional adaptation of the Rice parameter as in Eq. (6.1) restricts each Rice parameter to be non-negative, which will limit the freedom of the bit assignment of GR encoding, which is the base of the proposed integer-mapping methods. Here, the bit-plane rearrangement, LSBM proposed in Sec. 3.4, can be applied to avoid this restriction. LSBM can, in an invertible way, whiten the quantized spectra to be uniform scales making the XDGR code, the GR code extension achieved by the proposed bit inversion, available to be performed.

To summarize the above explanations, the proposed coding scheme works as depicted

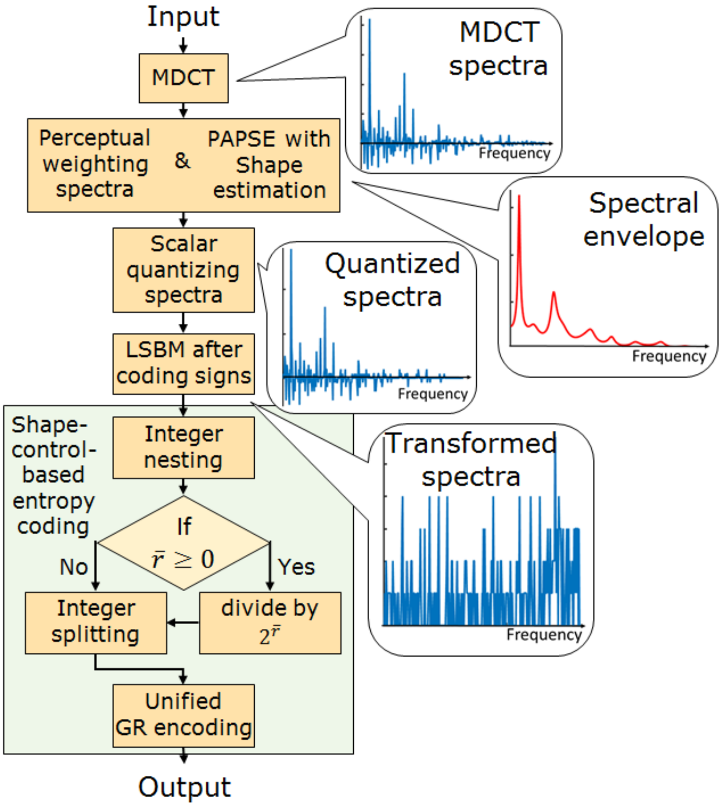


Fig. 6.19. Brief flowchart of the encoding processes in the proposed scheme.

in Fig. 6.19: In the integrated encoder, the input is first transformed into MDCT spectra with their shape parameter and envelope estimated; the spectra get perceptually weighted based on the smoothed envelope followed by scaler quantization; the signs of the non-zero quantized spectra are coded into one bit each, and the LSBM transforms the absolutes into smoothed ones following the envelope; finally, the transformed spectra go through the proposed shape control, using the p and q in a predetermined range that makes p/q nearest to the estimated shape parameter, and then encoded by unified GR code, the combination of the proposed XDGR and conventional GR codes.

It should be noted that the model parameters required to be sent to the decoder for each frame are GR-coded spectra, the LP coefficients for the envelope, the step size for the quantization, and the shape parameter, of which only difference to the baseline coding scheme is the frame-wise shape parameter. Therefore, to focus on the effect of the proposed shape control, we do not discuss here the quantization of parameters instead of MDCT spectra and shape parameters.

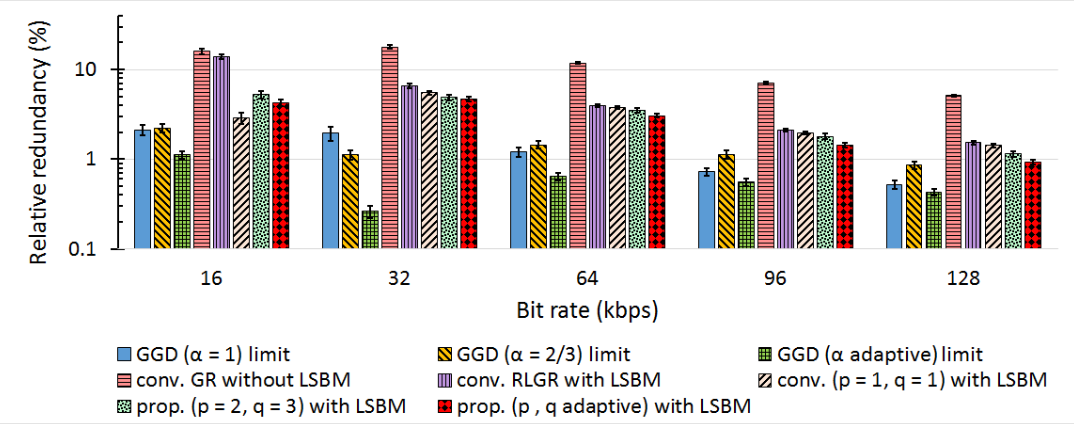


Fig. 6.20. Relative redundancy of each coding scheme for real audio frequency spectra. The item-wise average and 95% confidence interval.

6.4.2 Performance in compression efficiency

We evaluated the compression performance by applying it to the baseline TCX, as explained above. A hundred-and-forty items of audio in the RWC Music Database [101] and super-wide-band speech, eight seconds each down-sampled into 16 kHz, were quantized, at 16, 32, 64, 96, and 128 kbps, by an oracle encoder based on [106]: The encoder can use PAPSE switching shape parameter α frame by frame with no limitation and simulates the bit rate of quantized spectra $\hat{X}_k = \lfloor W_k X_k/s + 0.5\rfloor$ by the log likelihood for the GGD assumption $\hat{X}_k \sim \tilde{p}_\alpha(x|W_k\phi_k/s)$ where $\{X_k\}_{k=0}^{N-1}$, $\{\phi_k\}_{k=0}^{N-1}$, $\{W_k\}_{k=0}^{N-1}$, and s are respectively the MDCT spectra of input signals, their envelope, perceptual weights, and the step size searched for each bit rates. We calculated the code length of the quantized spectra in the following eight conditions using the same perceptual weights $\{W_k\}_{k=0}^{N-1}$, step size s, and frame size $N = 320$:

1. The theoretical limit using the Laplacian assumption $\hat{X}_k \sim \tilde{p}_1(x|W_k\phi_k/s)$ with the envelope $\{\phi_k\}_{k=0}^{N-1}$ estimated by 16-th-order PAPSE of $\alpha = 1$ [GGD ($\alpha = 1$) limit];
2. the theoretical limit using the constant GGD assumption $\hat{X}_k \sim \tilde{p}_{2/3}(x|W_k\phi_k/s)$ with the envelope $\{\phi_k\}_{k=0}^{N-1}$ estimated by 16-th-order PAPSE of $\alpha = 2/3$ [GGD ($\alpha = 2/3$) limit];
3. the theoretical limit using the adaptive GGD assumption $\hat{X}_k \sim \tilde{p}_\alpha(x|W_k\phi_k/s)$ with shape parameter α estimated frame by frame using the same estimation conditions

in [106] and the envelope $\{\phi_k\}_{k=0}^{N-1}$ estimated by 16-th-order PAPSE of the estimated α [GGD (α adaptive) limit];

4. the baseline coding scheme introduced in the previous section using GR encoding with the envelope $\{\phi_k\}_{k=0}^{N-1}$ estimated by 16-th-order PAPSE of $\alpha = 1$ [conv. GR without LSBM];
5. the proposed coding scheme using the conventional RLGR encoding instead of the unified GR encoding with its coding parameter searched exhaustively after transforming the quantized spectra by LSBM based on the envelope $\{\phi_k\}_{k=0}^{N-1}$ estimated by 16-th-order PAPSE of α estimated frame by frame as in condition 3 [conv. RLGR with LSBM];
6. the proposed coding scheme using the proposed shape control of $(p, q) = (1, 1)$ and with the envelope $\{\phi_k\}_{k=0}^{N-1}$ estimated by 16-th-order PAPSE of $\alpha = 1$ [prop. ($p = 1$, $q = 1$) with LSBM];
7. the proposed coding scheme using the proposed shape control of $(p, q) = (2, 3)$ with the envelope $\{\phi_k\}_{k=0}^{N-1}$ estimated by 16-th-order PAPSE of $\alpha = 2/3$ [prop. ($p = 1$, $q = 1$) with LSBM];
8. the proposed coding scheme using the proposed shape control of adaptive (p, q) in accordance with the shape parameter α estimated frame by frame as in 3) with the envelope $\{\phi_k\}_{k=0}^{N-1}$ estimated by 16-th-order PAPSE of $\alpha = p/q$ [prop. (p, q adaptive) with LSBM].

Condition 6 is equivalent to the proposed coding scheme without the proposed shape control. Note that the shape parameter $\alpha = 2/3$ was chosen because it was close to the parameter that showed high performance in [105] for a constant shape parameter. The results for conditions 3, 5, and 8 include 1 bit for representing the frame-wise shape parameter α. Since every condition shares the same perceptual weights, the lower code length means there is room for more precise quantization, and thus higher sound quality can be expected at the same bit rate compared to the conventional coding schemes, which have shown subjective quality compatible with the state-of-the-art standards in [105–107].

The relative redundancy of each condition, compared to the bit rate of the oracle encoder, is described in Fig. 6.20. The proposed coding scheme showed higher compression efficiency than the pure GR and RLGR encodings. The approximation of the shape control became harder to hold at lower bit rates, resulting in a little worse performance at 16 kbps. That was due to the course decorrelation, compared with the quantized values of spectra, by the bit-wise operations of LSBM, where a room remains for further enhancement. However, the proposed shape control generally enhanced the efficiency even when used with constant parameters (p, q). Also, using frame-wise (p, q) further improved the performance, especially at higher bit rates.

Of course, the arithmetic coding can achieve very close to the theoretical limits, so we do not insist on the superiority of the proposed coding to it. However, the shape control, combined with GR coding, has lower requirements for computation resources and will remain a reasonable choice when designing a codec.

6.5 Conclusion of Chapter 6

In this chapter, we focused on frequency-domain audio codecs, codecs encoding audio signals in the frequency domain, in a low-bit-rate and low-delay condition aiming at their use in mobile communications. These frequency-domain codecs are mainly expected to deal with signals, including a variety of sounds that time-domain codecs, the conventional speech codecs, fail to exert high performance. Generally, codecs have intrinsic trade-offs between the compression efficiency and the delay they permit, and especially in frequency-domain codecs, the restriction of low delay significantly affects their reproduction quality. Therefore, this chapter presented independent approaches for integrating the proposed shaping methods to modify TCX, the state-of-the-art frequency-domain codec, from the perspectives of signal and code modelings.

In the signal-modeling approach, we respectively evaluated the effects of the proposed frequency and amplitude warping. The frequency warping showed some enhancement in envelope estimation by warping its frequency resolution, improving the baseline TCX based on GR coding. On the other hand, amplitude warping enabled us to use the GGD assumption for frequency spectra in entropy coding, which revealed to be more reasonable compared to the conventional Laplacian assumptions and enhance the compression efficiency of the baseline TCX based on arithmetic coding. Additionally, the shape parameter estimation and the combination of the frequency and amplitude warping are investigated as promising possibilities for further extensions of these methods.

In the communication-modeling approach, we realized to introduce the GGD assumption into the baseline TCX based on GR-coding by applying the proposed bit inverting and integer mapping. Supported by the proposed bit-plane rearrangement, the codes extended from GR code were combined with the proposed amplitude warping, enhancing the compression efficiency of the baseline TCX based on GR-coding. The compression efficiency was yet far from that of the theoretical limits, but the low-complexity algorithms derived from the proposed methods will give more flexible choices for codec design.

Both approaches were appeared to be useful for the low-bit-rate and low-delay frequency-domain codec. All of the methods proposed here, including the extensions, are designed independently and available for combinational use. Therefore, they are expected to be integrated into the speech and audio codecs, the hybrid codecs of time and frequency domain, realizing communications with higher quality, and a part of these methods are already adopted in the recent standard 3GPP EVS [34].

Chapter 7

Application for low-delay and high-quality sound-data transmission

7.1 Introduction of this chapter

The previous chapter presented the use case at rather low bit rates. In this chapter, we go on to another low-delay coding application requiring higher bit rates. Recent progresses in audio codecs, MPEG-4 AAC family [9, 24, 46], ALS [11, 12], and 3GPP EVS [6–8] for example, have enabled us to transmit more freely high quality music and speech. Moreover, owing to the development of communication networks, there is a rising demand for real-time transmission of high-quality audio.

In radio broadcasting, sound data of live coverage have to be transmitted to the radio station and mixed before broadcasted. Even if the mixed audio stream gets compressed and degraded when delivered on the radio, sound data should have as little degradation as possible when mixed or processed. Therefore, in those cases, recorded sounds are preferred to be transmitted in real-time with high fidelity.

As in the case of speech communication, real-time audio transmission critically requires low transmission delay, the delay caused by the codecs, networks, computation, for example. The transmission delay is one of the critical factors for presence, and trained listeners are said to detect it from 100 ms [25]. Since typical IP network delivery takes around 100 ms, the algorithmic delay of the codecs should be negligibly short. For instance, in radio broadcasting of sports, the listeners tend to receive the radio at the relay site to listen to

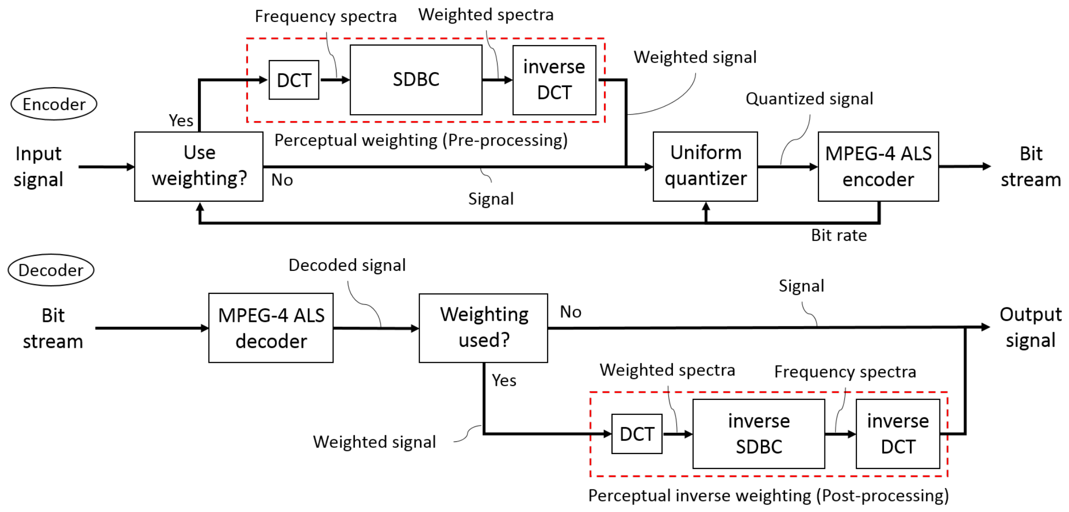


Fig. 7.1. Structure of the CLEAR encoder (upper half) and decoder (lower half).

the commentary in real-time, which strictly requires low-delay delivery of sounds.

In Japan, the radio broadcasters have used for years Integrated services digital network (ISDN) services [108], metal-network services of low-cost and high reliability. However, the deterioration of the metal lines required their providers to migrate to new services: Data connect [109] is one of them, which is a high-quality optical-network service with low cost at 1-Mbps limit, about three-quarters of the bit rate of compact-disc quality sound data. For this reason, a new codec fully making use of the new network services is needed recently. This kind of network services, to guarantee real-time transmission, require high priority and stability, which usually have restrict the limit of their bit rates following their fee. Therefore, it is preferable to use lossy codecs of constant bit rates instead of lossless ones of variable bit rates such as MPEG-4 ALS. Note that indeed, the scalable codec standard MPEG-4 Scalable to lossless (SLS) [33] can control the bit rate, but its frame size limits the flexibility of the codec delay, and thus it does not support short delay conditions.

Enhanced apt-X [110] and NICAM [39] are the codecs capable of compressing audio signals with little degradation and low delay. These kinds of codecs are designed to quantize signals to make the quantization noise inaudible, which sometimes requires a different criterion than SNR. Awing to their designs, these codecs achieve compression with quantization noise that non-professionals can hardly detect. However, in cases such as sound data transmission in radio broadcasting, where high bit rates are allowed, many digital sounds can be transmitted losslessly by invertible compression. Even in such cases, these codecs apply the quantization, which degrades the fidelity of the signals in SNR. Taking into account that the sounds must pass through the mixing process, they should be compressed losslessly whenever possible.

For the above reasons, we present in this chapter Conditionally lossless encoding under allowed rates (CLEAR), a near-lossless full-band stereo compression scheme enhancing the fidelity of sound data with high perceptual quality. It focuses on compressing the 48-kHz 16-bit stereo speech-and-audio data into about one to three quarters with the algorithmic delay of around 2 ms. As stated in the following section, from the business point of view, the designed codec is required to be used in low running and maintenance costs. Therefore, it is very reasonable if we can reuse the existing international standards. Here, by applying the perception model proposed in Chap. 4, it is shown that we can design the lossy codec from an existing international standard without changing its format.

We first show the overall scheme of the proposed CLEAR in this chapter. Then, it is evaluated by some objective and subjective measurements.

7.2 Proposed low-delay near-lossless coding scheme

7.2.1 Bit-rate-controlled compression

CLEAR is based on the idea of [72], which combines pre- and post-processed uniform quantizer with invertible compression. Instead of just lowering the bit rates of the invertible compression by the quantization as in [72], CLEAR controls the bit rate to be under the target rate and leaves the input signals lossless if possible.

From the perspective of the costs related to the codec, the costs for the development, including the audio transmission services, and maintenance, if available, it is better to use international standards than to make up from scratch because there is official support usually, and sometimes the transfer protocols are prepared for their formats.

From the above facts, the CLEAR encoder and decoder are designed as in Fig. 7.1. The CLEAR encoder compresses 48-kHz 16-bit stereo signals, about 1.5 Mbps of pulse code modulation (PCM) data, by 1.3-ms frame without overlaps. Therefore, the algorithmic delay of the CLEAR encoder is 1.3 ms. For the invertible compression, MPEG-4 ALS simple profile [111] is used, which has the following characteristics:

1. It supports low delay conditions (delay from one sample).
2. It represents signals with LP filters and their entropy-coded prediction residuals.
3. It has an option of *losslessly* omitting LSBs of the input signals when their LSBs are all zero in one frame.
4. It deals with bitstreams that can be sent by a standard protocol such as IETF RTP 3640 and 6416 [112, 113].

The first one enables us to control the algorithmic delay of the codec easily. The second one makes efficient entropy coding for signals with time correlation. The third one aims at dealing with cases in broadcasting where the devices sometimes represent 20-bit audio signals as 24-bit amplitude by padding 4 bits of LSBs for each sample. This option enables the CLEAR encoder to quantize the input signals by altering their LSBs to zeros without additional bits for representing quantization steps: Quantizing *outside* the MPEG-4 ALS encoder, the CLEAR encoder invokes the option that automatically makes use of the altered LSBs to reduce more the data size.

With the option, the CLEAR encoder iteratively compresses and quantizes the input signal, altering its LSBs to make the MPEG-4 ALS encoding bit rate under the target rate. Fig. 7.2 illustrates the frame-wise rates of altered LSBs during this process. The more LSBs altered to zero, the coarser the quantization, and no bit altered means lossless compression.

7.2.2 Adaptive pre- and post-processing

Coarse uniform quantization makes quantization noise with flat frequency spectra, which sometimes audible at higher frequencies. Therefore, the CLEAR encoder applies perceptual weighting as pre-processing when more than 4 bits are altered to zero by the quantizer. The CLEAR decoder judges whether the weighting is done by checking the number of LSBs altered and performs inverse weighting as post-processing if necessary.

The perceptual weighting in CLEAR comprises block companding in the DCT domain. SDBC, introduced in Chap. 4, approximates the masking thresholds and weights DCT spectra following them without any auxiliary parameters required for the psycho-acoustic

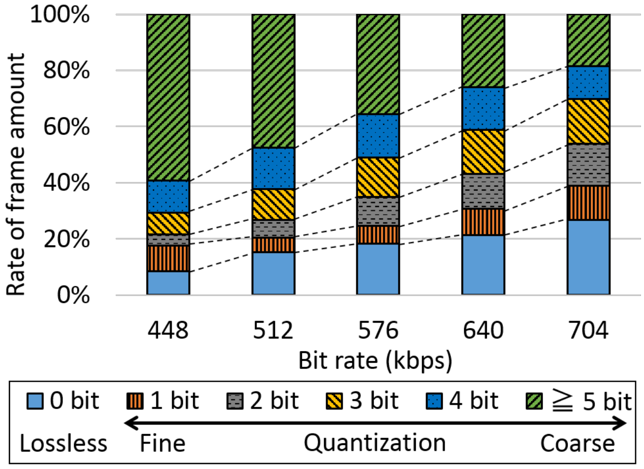


Fig. 7.2. Rates of altered LSBs by the CLEAR encoder at each bit rate. 42 speech and audio items, 8 seconds each of 48-kHz 16-bit stereo files.

model as in [72].

Since only when the quantization is coarse enough is this weighting performed, the CLEAR encoder can losslessly compress more than a quarter of frames at 704 kbps, according to Fig. 7.2. Moreover, CLEAR is backward compatible with MPEG-4 ALS because both the quantization and the perceptual weighting do not require additional information to decode.

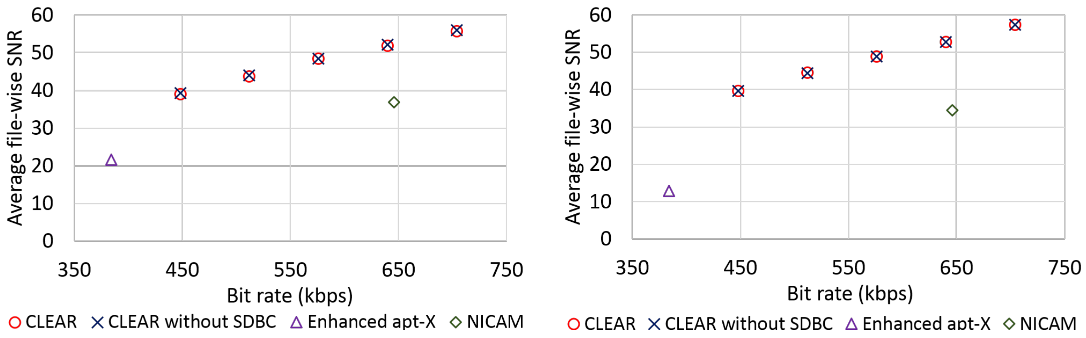
To summarize the above, this CLEAR is implemented by

- LP, inside MPEG-4, ALS for the signal modeling
- Adaptive pre- and post-processing by SDBC for the perception modeling
- the uniform quantization outside MPEG-4 ALS and GR code inside MPEG-4 ALS for the code modeling

where perception modeling is the main focus of this chapter for employing the proposed shaping methods.

7.3 Performance in objective sound quality

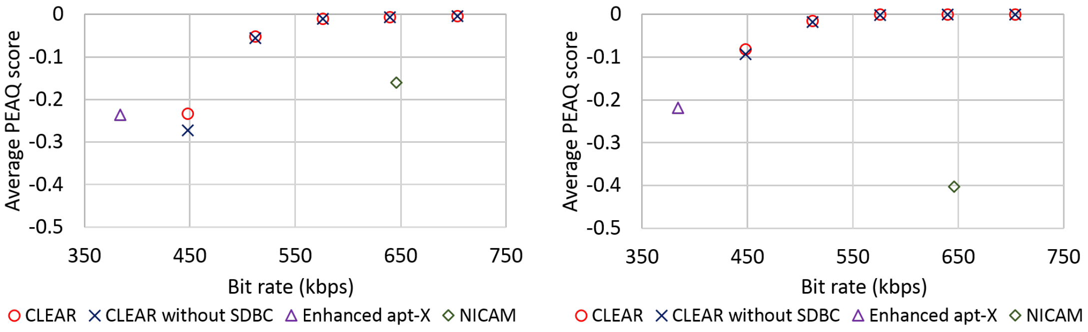
To evaluate the performance of CLEAR codec, we compared with the benchmark codecs, Enhanced apt-X, and NICAM. The algorithmic delays of the codecs were respectively 1.3, 1.9, 1.0 ms. The fidelity and the objective quality of the reconstructed signals were respec-



(a)

(b)

Fig. 7.3. Average SNR of the item-wise reconstructed signals. (a) Original items from the database. (b) Items whose volume turned down to -26 dBov.



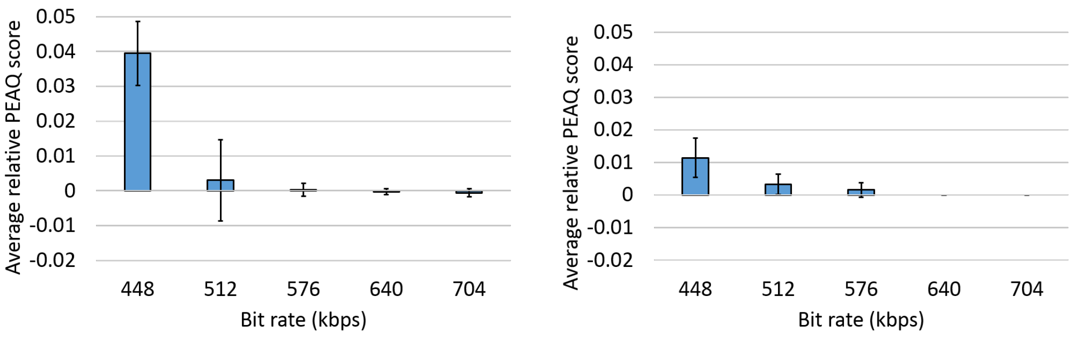
(a)

(b)

Fig. 7.4. Average PEAQ scores of the item-wise reconstructed signals. (a) Original items from the database. (b) Items whose volume turned down to -26 dBov.

tively evaluated as file-wise SNR and PEAQ scored from -4 to 0 by McGill University's AFsp PQevalAudio [100]. The test items included 140 speech and audio files, 8 seconds each of 48-kHz 16-bit stereo recording. These items were tested in two ways: Inputting to the codecs directly and inputting to the codecs after turning down the volume to -26 dBov to evaluate the performance for quiet sounds. Since NICAM supports only a 32-kHz sampling rate, the items were downsampled, only in evaluating NICAM, and upsampled to 48 kHz in calculating SNR and PEAQ scores. We also compared the CLEAR codec without using SDBC to evaluate its effects.

Fig. 7.3 plots the average SNR of each codec. The bit rates were for the total of the



(a) (b)

Fig. 7.5. Increase in PEAQ scores by SDBC, average and 95% confidence intervals by item-wise paired t test. (a) Original items from the database. (b) Items whose volume turned down to -26 dBov.

two channels. CLEAR showed high reconstruction fidelity compared to the other codecs. It can also be seen that the performance of CLEAR slightly increased for quiet sounds because CLEAR compresses signals losslessly if they are statistically easy to compress. The other codecs, on the other hand, failed to take advantage of the quietness. Note that Enhanced apt-X has about 50% longer algorithmic delay than CLEAR. The high fidelity of CLEAR will make it easier to mix or process transmitted sound data.

Figs. 7.4 and 7.5 respectively plot the average PEAQ scores of each codec, and the scores of CLEAR increased by the proposed SDBC. Note that the centers of each symbol show each PEAQ score in Fig. 7.4, and all the scores are under zero. The small but statistically significant improvements in PEAQ scores by the proposed SDBC at lower bit rates indicate that the adaptive pre- and post-processing by SDBC will reduce the risk of annoying noise made by quantization without affecting the fidelity of the reconstructed signals.

7.4 Performance in subjective sound quality

Next, we held two subjective experiments. One was ITU-R BS.1116 [114], aimed at evaluating the effects of the proposed SDBC. Sixteen listeners were blindly provided pairs of original (reference) and coded (test) speech and audio items and scored each sound quality in degradation mean opinion score (DMOS) from 1 to 5. Twelve items were used in

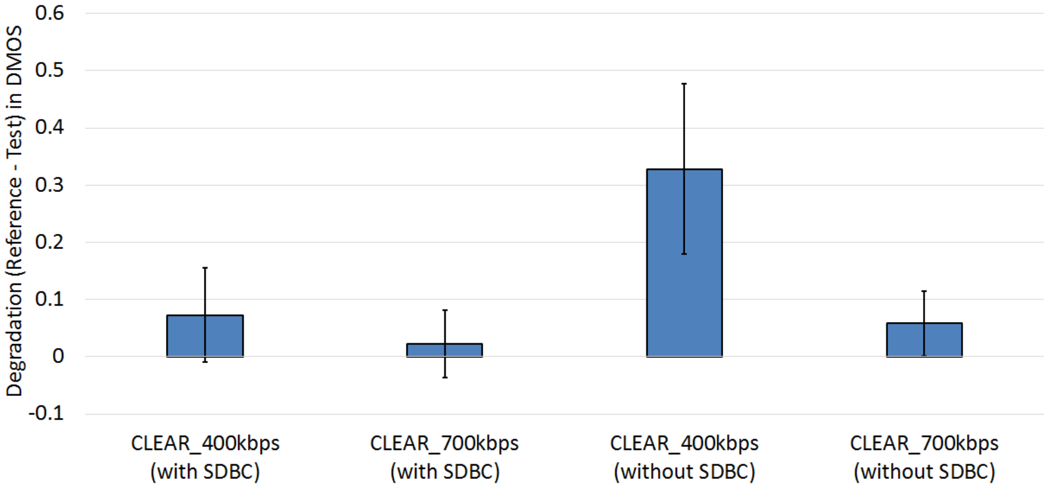


Fig. 7.6. Relative degradation by each condition in ITU-R BS.1116. Average and 95%

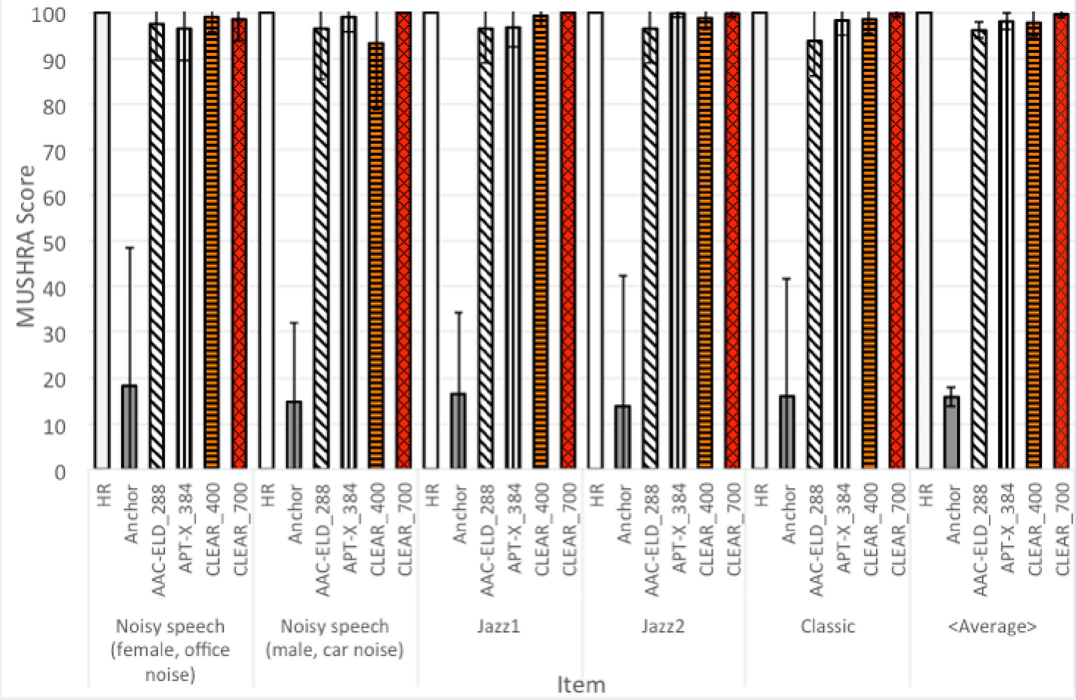


Fig. 7.7. Item-wise MUSHRA scores by each condition. Average and 95% confidence interval in t-test.

total, and four conditions, CLEAR at 400 and 700 kbps with and without SDBC. Fig. 7.6 shows the result, which plots the difference of the DMOS, or the average degradations. It can be seen that the proposed SDBC concealed the significant degradations shown in the conditions without SDBC.

Another experiment was TU-R BS.1534-1 MUSHRA [102], aimed at evaluating the sound quality of the proposed CLEAR. Four trained participants were blindly provided with the following conditions and graded the quality of each one from 0 to 100 points:

- Original items as hidden reference (HR);
- 3.5-kHz band-limited anchors (Anchor);
- ones coded by AAC-ELD at 288 kbps with 7.5-ms algorithmic delay (AAC-ELD 288);
- ones coded by Enhanced APT-X at 384 kbps with 1.9-ms algorithmic delay (APT-X 384);
- ones coded by CLEAR at 400 kbps with 1.3-ms algorithmic delay (CLEAR 400);
- ones coded by CLEAR at 700 kbps with 1.3-ms algorithmic delay (CLEAR 700).

The average scores are presented in Fig. 7.7. All the codecs showed a high sound quality that normal listeners without training often could not tell the difference. Despite its stricter low-delay condition, the proposed CLEAR performed compatible quality to those reference codecs, succeeding in maintaining both objective and subjective quality.

7.5 Conclusion of Chapter 7

We introduced in this chapter a low-delay and high quality audio coding scheme, CLEAR, as an application of the proposed shaping method in the perception modeling. A combination in the CLEAR codec of a uniform quantizer, MPEG-4 ALS, and adaptive pre- and post-processing, realized by the proposed SDBC, achieved effective compression of audio signals with high fidelity. Awing to the option of MPEG-4 ALS and the headerless process of SDBC, the CLEAR encoder can output bitstreams based on the MPEG standard. Objective and subjective experiments revealed its high performance in reconstruction fidelity with the compatible perceptual quality compared with benchmark codecs. CLEAR is expected to be useful in real-time sound data transmission, where sounds are mixed after transmitted, and is adopted in a commercial codec, HDIP-3000V, now on sale from RF Design [115].

Chapter 8

Application for low-delay and packet-independent inter-device communication

8.1 Introduction of this chapter

The development and spread of IP have made a considerable influence on the trend of speech compression. Notably, the concept of packet freed us from the risks of bit errors, allowing us to use powerful variable-length coding tools such as arithmetic coding even in real-time speech communication. These tools have enhanced the performance of many codecs and contributed to the state-of-the-art speech codec standards such as 3GPP EVS. The codecs presented in the previous chapters also depend on this kind of protocol.

However, there remains a demand for speech communication without IP in conditions where bit errors will occur. Among various Internet of things (IoT) systems, although they are mainly expected to depend on IP, some use cases may need local communication or observation, which require total optimization with a more simple protocol having smaller overhead at the expense of bit-error correction capacity. As a well-known conventional example, wireless microphones need low-delay codecs dealing with bit errors. With variable-length coding tools used in these cases, bit errors will make the decoder mistake the sample length to decode, with its influence lasting for a long period and collapsing the whole communication in the worst case. This possibility of the mistakes will not be zero even if the code is protected by some error-correction codes. For this reason, ADPCM [26] and its derivatives such as Enhanced APT-X [110], codecs used for compressing high-quality speech in the above conditions, have fixed bit assignments and tend to use high bit rate, which limits the number of devices working simultaneously at limited communication bandwidth. Besides, the high bit rate makes the bit-error problem harder, even for low bit error rates (BERs), compared with the case of low-bit-rate radio communication.

Therefore, in this chapter, we present Bit-plane rearrangement for audio and voice encoding (BRAVE), an efficient low-delay and bit-error-robust codec with lower bit rates than the conventional ones to transmit more sounds simultaneously with high sound quality. It focuses on 32-kHz 16-bit monaural inputs compressing into about one fifth with around 3-ms algorithmic delay, which is a stricter delay condition than mobile communication because it may be used in load speakers, for example. To deal not only with speech but also music items, we designed based on MDCT, which can easily control the quantization noise, instead of codecs based on strict speech models, for example, ITU-T Rec. G.728, G.729, and Pitch synchronous innovation code excited linear prediction (PSI-CELP) [17, 116, 117] although they are robust against bit errors and some of them have contributed to non-IP mobile communication.

Guaranteeing the decoded frame-wise sample length under bit-error conditions requires strictly fixing the total bits used for each frame. At higher bit rates, vector quantization, one of the well-used fixed-length coding techniques, has difficulties in avoiding its exponentially increasing computation costs, although there are many extensions to reduce them [40, 118, 119]. Therefore, codecs working at the bit rates focused here are preferable to use sample-wise bit assignment with scalar quantization. In this case, the compression performance of the codec greatly depends on how to assign the bits to each frequency spectra and represent the bit assignment itself.

Due to this fact, we adopt the shaping method introduced in Sec. 3.4, LSBM, which realizes an error-robust realization of bit assignment based on spectral envelopes and reduces the effects of the bit errors occurred in the bit assignment on the degradation of the reconstructed frequency spectra. The paper first quickly reviews the conventional idea of bit assignment for frequency spectra, providing the relationship with the spectral

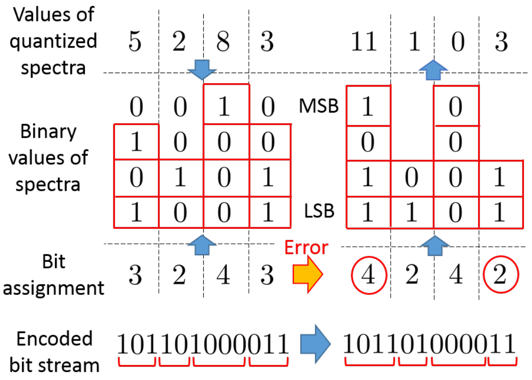


Fig. 8.1. Simple coding of quantized spectra with bits assigned to each frequency. Bit assignment indicates the number of the red blocks used for packing each spectrum into the bit stream. If the bit error, in the yellow arrow, changes the information of the bit assignment as in the red circles, the bit steam read by the decoder will be shifted, resulting in large errors even in the frequency where the bit assignment is correct.

envelopes. Then, the structure of the proposed BRAVE is described, followed by some objective and subjective evaluations.

8.2 Proposed low-delay bit-error-robust coding scheme

8.2.1 Envelope-based bit assignment

Many methods of bit assignment for frequency spectra are based on the idea of adaptive transform coding in [120], which insists on the optimality of bit assignment in the sense of rate-distortion trade-offs. The following discussions focus on assigning bits to positive values by omitting polarity of the spectra, which can be coded independently or combined with the positive values by folding and interleaving as in [83]. Assuming real-valued MDCT spectra $\{X_k \in [0, \infty)\}_{k=0}^{N-1}$, each distributed in accordance with independent probability density functions (pdfs) having a same shape but different variances $\{\sigma_k^2\}_{k=0}^{N-1}$, the probability distribution and the optimal expected code length $\{b_k\}_{k=0}^{N-1}$ of each spectra scalar-quantized by quantization steps $\{d_k\}_{k=0}^{N-1}$, can be approximately written with a pdf

$p_X(X|1)$ of scale 1 as $\left\{\frac{d_k}{\sigma_k} p_X\left(\left.\frac{d_k X}{\sigma_k}\right| 1\right)\right\}_{k=0}^{N-1}$ and

$$
\begin{aligned}
b_k &\simeq \int_0^\infty -\frac{d_k}{\sigma_k} p_X\left(\left.\frac{d_k X}{\sigma_k}\right|\right) \log_2\left(\frac{d_k}{\sigma_k} p_X\left(\left.\frac{d_k X}{\sigma_k}\right|\right)\right) \mathrm{d}X \\
&= \log_2 \frac{\sigma_k}{d_k} + C(p_X(X|1)),
\end{aligned} \tag{8.1}
$$

respectively, where $C(p_X(X|1))$ is a constant depending only on the shape $p_X(X|1)$ of the distributions. If the quantization steps are small enough, the expected energy of the quantization noise will proportionate d_k^2, and thus the rate-distortion optimizing problem of a given bit rate B becomes

$$
\min_{\{d_k\}} \sum_{k=0}^{N-1} d_k^2 \text{ s.t. } B = \sum_{k=0}^{N-1} b_k. \tag{8.2}
$$

This leads to the optimal quantization steps

$$
d_k = 2^{-\frac{B}{N}} \left(\prod_{j=0}^{N-1} \sigma_j\right)^{1/N}, \tag{8.3}
$$

uniform among the frequencies n, and the bit assignment

$$
b_k = \log_2 H_k + C(p_X(X|1)), \ \ H_k \equiv \frac{\sigma_k}{\left(\prod_{j=0}^{N-1} \sigma_j\right)^{1/N}}, \tag{8.4}
$$

of which differences among the frequencies k depend only on the normalized standard deviation $\{H_k\}_{k=0}^{N-1}$.

The signal modeling often approximates this normalized standard deviation as, for example, LP coefficients or LSPs [30, 32, 80, 105, 107] because the spectral envelope derived from the model has a property that its geometric average among the frequencies becomes 1, just as $\{H_k\}_{k=0}^{N-1}$ has. Therefore, the encoder can optimally code the spectra by the combination of scalar quantization and bit assignment based on the parameterized spectral envelope. The decoder reconstructs the spectra by first decoding the envelope and then reading the bitstream following the bit assignment the envelope suggests. More concretely speaking, parameterizing the bits to be assigned is equivalent to parameterizing the logarithmic spectral envelope.

As mentioned in the introduction, we have to use a strictly fixed bit assignment. Namely, the bit assignment $\{b_k\}_{k=0}^{N-1}$ should be integer as is the case of the codec in [120]. However, in cases where the encoder naively packs the code of spectra sequentially into the bitstream

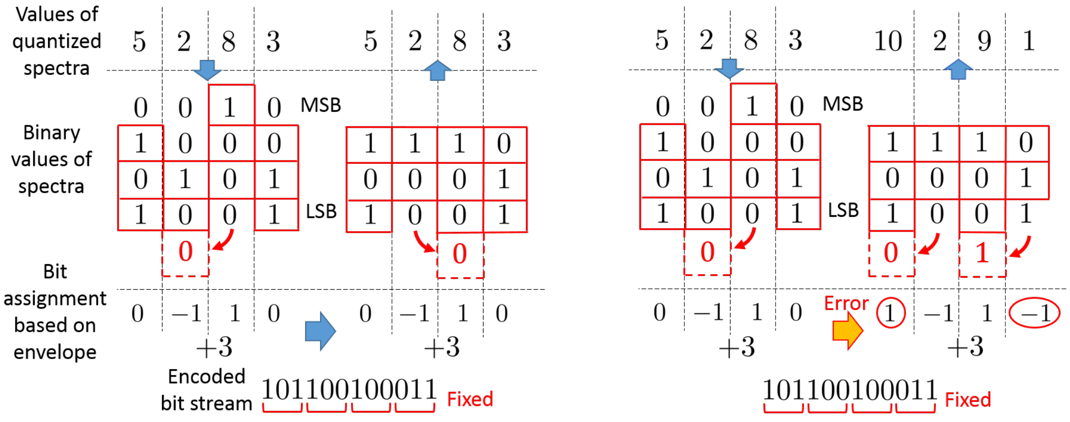


Fig. 8.2. Coding of quantized spectra with LSBM. If the bit assignment is sent correctly, the decoder can reconstruct the quantized spectra correctly (left half). Even if errors change the bit assignment as in the red circles, the order of the bit stream read by the decoder will not change (right half).

as in Fig. 8.1, even a little error in the bit assignment disturbs the whole order of the bits read by the decoder, making unexpected decoding errors. Indeed, it may be effective to modify the scanning order of the bit planes, such as in [121], but it still has a risk of unexpected collapse.

8.2.2 Least significant bit management based on spectral envelope

Here, LSBM proposed in Sec. 3.4 can be used, which enables us to realize invertible 'division' in the integer region by managing the LSBs of the quantized values of the spectra. Using the proposed LSBM, the decoder can read out the bitstream by fixed bit-length intervals so that even if the bit assignment changed by errors, the influence of the errors do not spread to the whole order of the bits, which guarantees that the most significant bits (MSBs) of the spectra will be preserved where their corresponding envelopes are correct. For the example in Fig. 8.2, 3 bits of MSBs of the spectra are guaranteed in this sense. Therefore, there are fewer chances where the errors crucially distort the outline of the spectra, as in Fig. 8.1, and the decoder can avoid substantial perceptual degradation. Of course, the spectral envelop $\{H_k\}_{k=0}^{N-1}$ can be coded using all-pole models represented by LP coefficients or LSPs. However, since the bit assignment needs only the integer precision for the LSBM scheme, in low-delay conditions where the

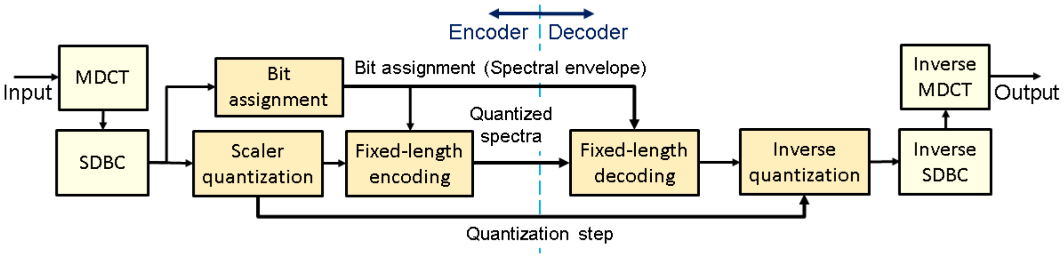


Fig. 8.3. Brief outline of the proposed BRAVE coding scheme.

frame length and the envelope length are short, it may be efficient to code the envelope-dependent terms $\{\log_2 H_k\}_{k=0}^{N-1}$ directly by vector quantization, for instance. In these cases, the distortion can be minimized by preparing the patterns of envelope-dependent terms $\{\log_2 H_k^{(m)}\}_{k=0}^{N-1}$ for $j = 0, \cdots, M-1$, which are organized to sum up to zero in each pattern, and searching the pattern minimizing the spectral energy $\sum_{k=0}^{N-1} \left| X_k / H_k^{(m)} \right|^2$, with bit-shift operations, for example.

In fact, taking into account that $\sum_k \log_2 H_k = 0$ for every patterns, this minimization problem is rewritten as

$$\min_m \sum_{k=0}^{N-1} \left(\left| X_k / H_k^{(m)} \right|^2 - \ln \left| X_k / H_k^{(m)} \right|^2 - 1 \right), \tag{8.5}$$

in other words, equivalent to the IS divergence minimization, with which LP analysis also deals [28]. Therefore, the vector quantization mentioned above has the same characteristics in the sense of envelope fitting in LP.

8.2.3 Codec design

Based on the above ideas, the proposed BRAVE works as in Fig. 8.3. The codec compresses 32-kHz sampling rate 16-bit depth monaural signals into around 96 kbps, working for 32- or 64-sample frames with quarter overlaps, which makes 1.5- or 3-ms algorithmic delay. It first transforms the input signals into MDCT spectra and performs perceptual weighting by SDBC presented in Chap. 4. Then, the envelope-dependent terms of the bit assignment are determined by vector quantization mentioned above based on the weighted spectra. The weighted spectra are scalar quantized by a single quantization step, followed by LSBM. The envelope-dependent terms used for LSBM are represented in 4 bits at 32-sample-frame mode and 12 bits at 64-sample frame mode by the vector quantization. For

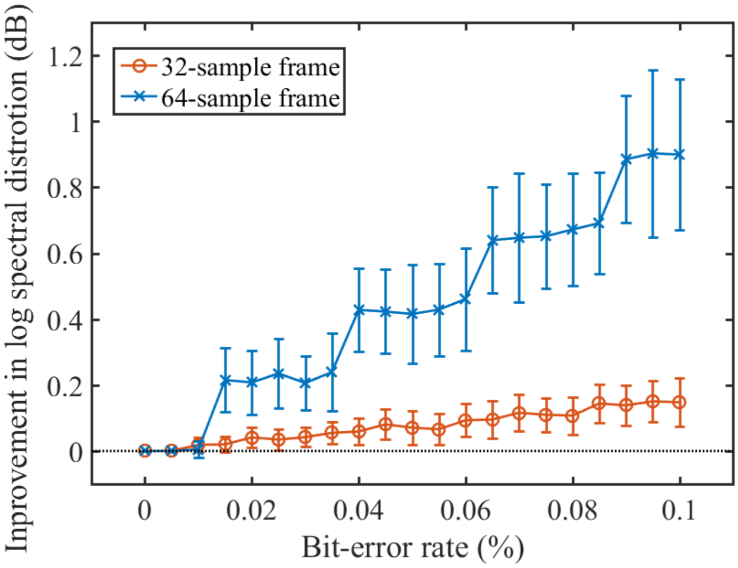


Fig. 8.4. Item-wise improvement in log spectral distortion for each BER. Average and standard deviation. Circles and crosses indicate 32- and 64-sample-frame conditions, respectively.

reconstructing speech signals, the decoder performs pitch enhancement [27] by a post-filter using only the decoded signals. It should be noted that the SDBC allows us to use only the envelopes of integer precision because it does not require LP coefficients such as the perception modeling by smoothed envelopes.

To summarize the above, this BRAVE is implemented by

- integer-precision envelopes with LSBM for the signal modeling
- SDBC for the perception modeling
- the uniform quantization and fixed-bit-length coding for the code modeling

where the signal modeling is the main focus in this chapter for employing the proposed shaping methods.

8.3 Performance in bit-error robustness

At first, to check the robustness of the proposed LSBM, we compared the log spectral distortion [122] of the reconstructed MDCT spectra by artificially making bit errors on the encoded bitstreams. Two conditions were prepared for BRAVE: one used proposed LSBM

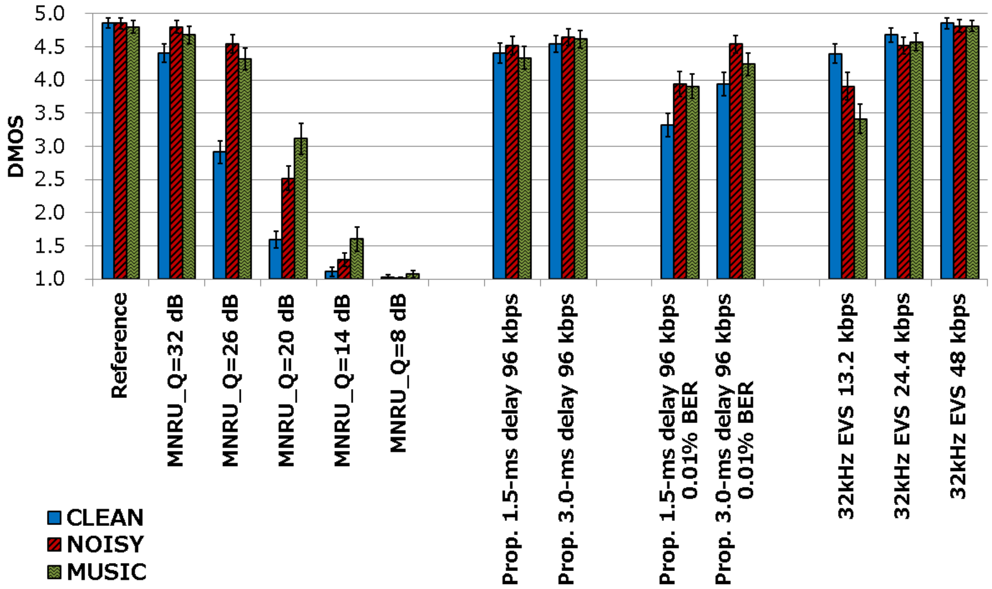


Fig. 8.5. Condition-wise DMOS of ITU-T Rec. P.800. Error bars indicate 95 % confidence intervals.

for bit assignment, and the other used conventional simple sequential bit assignment. We encoded and decoded by these codecs 84 items, about 8 seconds each of speech and audio with a 32-kHz sampling rate and 16-bit depth. Among the encoded bitstreams, the bits representing the spectral envelopes were randomly altered respectively under each BER. Note that the commercial use in this condition usually requires under around 0.01% BER for safety.

Fig. 8.4 shows the improvement, the difference of the proposed LSBM from the conventional bit assignment, in log spectral distortion at each frame-length condition. The difference between the proposed and conventional bit assignments became more evident as the BER raised. However, the preliminary experiment on the reproduction SNR showed little difference between the conditions. That means the small values such as spectra in higher frequencies were damaged by bit errors more at the conventional bit assignment, and the errors should be more annoying for listeners compared to them in the case of using the proposed LSBM.

8.4 Performance in sound quality

To evaluate the quality of the proposed BRAVE, we held a subjective experiment for degradation category rating (DCR) based on ITU-T Rec. P.800 [123]. 24 participants rated, in five-point degradation, four items each of clean speech, noisy speech and music for the respective conditions. For a reference, we prepared EVS, known to perform very high quality at mobile communication using more than ten times longer algorithmic delay (32 ms) compared to the proposed BRAVE. Some conditions had 0.01% BER with errors randomly generated based on uniform probability.

Fig. 8.5 describes the item-wise DMOS based on the average over the 96 votes for each condition. Due to the limitations of space, we omitted from the graph the results of Modulated noise reference units (MNRUs) at 14- and 8-dB signal-to-modulated-noise ratio. It can be seen that the proposed codec at 96 kbps displayed high DMOS around 4.5 on average. Also, the DMOS of the proposed codec with 0.01% bit error was around 4.0, showing little degradation, especially in noisy conditions.

8.5 Conclusion of Chapter 8

In this chapter, we presented a low-delay and bit-error-robust speech-and-audio codec, BRAVE, focusing on its use in real-time packet-free communication. Adopting LSBM, the proposed shaping method for the signal modeling, the bit assignment in the proposed BRAVE for frequency spectra was realized by the addition and subtraction of LSBs following the spectral envelopes. This method enables the decoder to guarantee the range of the damage by the bit errors occurring in the codes for spectral envelopes, which may reduce the annoying noise given by the errors.

The objective and subjective evaluations showed its high performance in sound quality even in the bit-error conditions. Therefore, the proposed BRAVE is expected to be useful in such as real-time inter-device communication and is in preparation for some commercial use.

Chapter 9

Conclusion

In this thesis, we presented shaping methods, methods extending basic speech-and-audio coding models by only inserting simple invertible transformation, aiming at their use in low-delay codecs. Generally, codecs have intrinsic trade-offs between the compression efficiency and the delay they permit, and the restriction of low delay, which required in many use cases related to real-time communication, greatly affects their reproduction quality. Therefore, this thesis interpreted in Chap. 2 speech-and-audio coding into three types of essential modeling that much reflect on the codec strategies and quality: The signal modeling determines the statistical assumptions for speech-and-audio signals, the distributions to which each sample or frequency spectrum belongs; the perception modeling defines the sensitivity of the listeners or receivers, which gives us some objective functions; the code modeling designs the optimization for the objective function given by the perception modeling subjecting to the assumptions given by the signal modeling. With each importance pointed out, the following chapters introduced the shaping methods from several points of view.

Chap. 3 presented the sample-based shaping methods for the signal modeling. Here, we focused on LP, one of the essential techniques for the signal modeling, and extended it by using two shaping methods. One was frequency warping that warps the frequency resolution of LP-based spectral envelope estimation using non-negative sparse matrices. Another was amplitude warping that realizes maximum-likelihood estimation of envelopes based on generalized-Gaussian-distributed frequency spectra with slight changes in estimation algorithms. The changes of envelope model derived from the shaping methods enabled the LP-based envelopes to be estimated under more reasonable assumptions. Additionally, LSBM was introduced, the shaping method realizing the invertible division, or whitening, of integer spectra. Owing to this shaping method, we can more flexibly use the envelopes even if we deal with quantized integer inputs, as in the case of contentious situations.

Chap. 4 presented the block-based shaping method for perception modeling. Since the codec design heavily depends on perception modeling, it is crucial to implement the psycho-acoustic characteristics more flexibly. The usage of the companding, one of the simplest models, was extended to make the codec design more flexible. The proposed SDBC enabled us, for the sake of requiring no auxiliary parameters for reconstruction, to realize the perceptual control of quantization noise as weighting pre- and post-processing outside of the compression optimization.

Chap. 5 presented the distribution-based shaping methods for the code modeling. GR code, the structured code used in many applications, was focused in this chapter. Without ruining its low complexity of coding algorithms, we aimed at extending it to optimally encode integers belonging to generalized Gaussian distributions, a more general class of distributions including Laplacian assumed in the conventional GR code. The proposed shaping by bit inversion gave us the extension for sparse, or low-scaled, Laplacian. XDGR and XDG codes, given from the inversion, can optimally encode sparse Laplacian sources, which GR code does not support, showing some symmetry between them in the compression performance. Another proposed shaping by integer mapping was proven to approximately shape the generalized-Gaussian-distributed sources into Laplacian by invertible mappings between a group of integers and a single integer. Therefore, combining the two shaping methods enables us to encode a wide range of generalized-Gaussian-distributed sources by GR code.

We also presented the low-delay applications realized by the above basic shaping methods. Chap. 6 was aimed at mobile communication, which requires low delay and low bit rate. Focusing on the state-of-the-art frequency-domain coding scheme used in the unified speech and audio coding in such conditions, we evaluated the proposed shaping methods for the signal modeling. Due to the enhanced compression efficiency given by the frequency and amplitude warping, the subjective quality of the codec was enhanced by the respective method, showing some possibilities of further extensions by using adaptive shapes of GGDs and by combining the warping methods. Moreover, we also evaluated the

GR code extensions given by the shaping methods for the code modeling. It was shown that we could enhance the GR-based codec by introducing the proposed bit inverting and integer mapping by realizing GR coding under the GGD assumptions.

Chap. 7 was aimed at real-time audio data transmission for broadcasting, which requires a low delay and high fidelity for reconstructed waveforms. The flexible feature of the proposed SDBC enabled us to design a low-delay and nearly-lossless codec based on lossless audio standard MPEG-4 ALS without changing its compression format, which much saved the development and maintenance costs of services. The objective and subjective evaluation proved the perceptual effects of SDBC and the high quality of the proposed codec from both perspectives of waveform fidelity and sound quality.

Chap. 8 was aimed at real-time inter-device communication for IoT devices, which requires a low delay and bit error robustness. The proposed LSBM, for the sake of its discrete characteristics, enabled us to make use of the spectral envelopes for compressing in fixed-length coding, robust against bit errors. The objective and subjective evaluation showed its bit-error robustness and high sound quality despite its very low algorithmic delay.

To summarize the above studies, the proposed shaping methods were confirmed to be useful for the low-delay speech-and-audio codec. It should be noted that some of these methods are already adopted for commercial use. Besides, the methods proposed were independently designed so that it is possible to combine each other, remaining as future issues. Therefore, the methods not adopted yet are also expected to be used in practical codecs combined with other technologies.

Chapter 10

Acknowledgments

This thesis summarized the results of the collaborative research between Nippon Telegraph and Telephone Corporation (NTT) and the University of Tokyo (from April 2013 to March 2015) and the continuing research in Moriya research laboratory after joining NTT Communication Science Laboratories (CS Labs.) in April 2015. Many people have supported the research of these years. So, I would like to list them here as possible and offer my sincere thanks to them.

First, I am very much grateful to Prof. Hiroshi Saruwatari for accepting to take his very precious time to review my work as my supervisor. He kindly gave me an opportunity of applying for the degree. He also gave me practical advices for summarizing the whole research and many comments for improving the thesis.

Secondly, I would like to thank the committee members (in alphabetical order), Prof. Ayumu Matani, Prof. Hiroshi Nakamura, Prof. Keisuke Hasegawa, Prof. Shoichi Koyama, and Prof. Takaaki Nara for spending their time examining the defense.

I would like to express my most profound appreciation to Dr. Takehiro Moriya, the NTT fellow being my substantial supervisor of the master thesis and also my superior in NTT CS Labs.. For years, Dr. Moriya has taught me about the audio coding from the basics to practice and given me many chances of study and work. Without his instruction, this research would not have been possible. It was my great honor to be able to work under the world-wide succeeding person like him.

I would also like to appreciate the other coauthors of all related papers, who belong to the NTT CS Labs.. Dr. Yutaka Kamamoto has kindly supported my research in many ways; discussing the studies, correcting the papers and presentations, advising me what to do in research, and so on. He gave me many valuable comments and encouragement during my research not only for the study but also for working life in NTT. Dr. Hirokazu Kameoka, a guest associate professor of the Graduate School of Information Science and Technology in the master course, has made enormous efforts for the management of Moriya-Kameoka laboratory at that time and provided me with fruitful comments on my works and presentations. Dr. Noboru Harada also contributed to the research, joining the discussions and giving insightful suggestions.

I would like to thank as well the co-developers of the CLEAR codec in Chap. 7: Mr. Toshiharu Morizumi, Mr. Tomoyuki Kanekiyo, Mr. Junichi Nakajima, Mr. Keisuke Hasegawa (now with NTT Service Evolution Laboratories (EV Labs.)), Mr. Takahito Kawanishi (now with NTT CS Labs.), Mr. Koichi Furukado (now with NTT InfraNet), Mr. Kenichi Noguchi (now with NTT Media Intelligence Laboratories (MD Labs.)), and Mr. Jouji Nakayama (now with NTT Intellectual Property Center). They made enormous efforts to integrate my implementation into hardware and service. Without them, the CLEAR codec would not have been commercially used.

Let me thank the co-developers of the BRAVE codec in Chap. 8: Mr. Shigeaki Sasaki, Mr. Yoshiaki Kozaki, Mr. Hiroki Ohba, and Mr. Yuki Fukuhara of NTT TechnoCross. They arranged my source code to make use of it in digital signal processors. Their implementation made the BRAVE codec reach the level of commercial use.

I am very grateful to those who helped me conducting listening experiments. Mr. Takaaki Kurita and Mr. Kazumi Kato of NTT Advanced Technology helped us carry out ITU-T P.800 in Chaps. 7 and 8. They were very experienced in formal standardized listening tests, and their help was indispensable for extensive evaluation. Also, I appreciate Ms. Kanako Sato of NTT Network Technology Laboratories (NT Labs.) for letting me use the laboratory equipment for the experiments.

I have to express my gratitude to those who have contributed to ensuring the intellectual properties of this study. Mr. Kazunaga Ikeda and Ms. Rie Yamada of NTT Intellectual Property Center have made great efforts to support building the patents. They have been inter-mediators between the law firm and me, with a deep understanding of my ideas. Especially, Mr. Ikeda has been extremely patient, interviewing me to develop the strategies for the patents. Mr. Takahiro Yoshimura, Mr. Yasuyuki Taya, Mr. Kazuhiro Kiwata, and Mr. Masaki Hayashi of Nakao & Associates Patent & Trademark Law Firm also helped me in writing the patents. Their fast and thorough works were very constructive. For the office procedures at the University of Tokyo, Ms. Kana Taguchi of Toudai Technology Licensing Organization (TLO) made great efforts even after I graduated from the master's course.

I have greatly benefited from the people related to the Moriya-Kameoka laboratory in my master's course. Ms. Naoko Tanji, the research care worker (now with Saruwatari & Koyama laboratory), has been extraordinary tolerant and supportive, dealing with the management of the laboratory and many office works. Even after graduation, she kindly helped me applying the degree in clerical procedures. Mr. Tomohiko Nakamura (now with Saruwatari & Koyama laboratory) gave me many advices as a senior student about school life. Mr. Norihiro Takamune (now with Saruwatari & Koyama laboratory), Mr. Takuya Higuchi (now with Apple), and Mr. Kento Kadowaki (now with Nomura Research Institute) have made my school life worthwhile, having discussions and sometimes competing with each other.

I have also greatly benefited from the members of Moriya research laboratory. Dr. Takashi G. Sato, Dr. Yoshifumi Shiraki, and Dr. Kenji Ishikawa have encouraged my research by discussing and giving many advices. Besides, they, and of course Dr. Moriya, Dr. Harada, and Dr. Kamamoto, have been very supportive in routine business in NTT. Owing to their help, I have been able to concentrate on my research.

Finally, this work has also been supported by many people not listed above: Many participants of the listening tests evaluated the sound quality of the codecs; reviewers of the papers gave me many technical suggestions related to the work. Their feedbacks certainly improved the quality of the study and greatly encouraged the development of the techniques. So, I would like to thank everyone concerned with this work again.

Chapter 11

Bibliography

[1] ITU-T Rec. G. 711, “Pulse code modulation (PCM) of voice frequencies,” 1988.

[2] “3GPP TS 26.090 Release 15 - Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Transcoding functions,” 3GPP, 2018.

[3] E. Paksoy, J. Carlos de Martin, A. McCree, C. G. Gerlach, A. Anandakumar, , and V. Viswanathan, “An adaptive multi-rate speech coder for digital cellular telephony,” in *Proc. ICASSP*, vol. 1, Mar. 1999, pp. 193–196.

[4] “3GPP TS 26.171 Release 15 - Speech codec speech processing functions; Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; General description,” 3GPP, 2018.

[5] R. Salami, B. Bessette, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola, and K. Jarvinen, “The adaptive multi-rate wideband codec: history and performance,” in *Speech Coding, 2002, IEEE Workshop Proceedings.*, Oct 2002, pp. 144–146.

[6] “3GPP TS 26.441 Release 12,” 3GPP, 2014.

[7] M. Dietz, M. Multrus, V. Eksler, V. Malenovsky, E. Norvell, H. Pobloth, L. Miao, Z. Wang, L. Laaksonen, A. Vasilache, Y. Kamamoto, K. Kikuiri, S. Ragot, J. Faure, H. Ehara, V. Rajendran, V. Atti, H. Sung, E. Oh, H. Yuan, and C. Zhu, “Overview of the EVS codec architecture,” in *Proc. ICASSP*, Apr. 2015, pp. 5698–5702.

[8] S. Bruhn, H. Pobloth, M. Schnell, B. Grill, J. Gibbs, L. Miao, K. Jarvinen, L. Laaksonen, N. Harada, N. Naka, S. Ragot, S. Proust, T. Sanda, I. Varga, C. Greer, M. Jelinek, M. Xie, and P. Usai, “Standardization of the new 3GPP EVS codec,”

in *Proc. ICASSP*, Apr. 2015, pp. 5703–5707.

[9] ISO/IEC, 13818-7, "Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC)," 1997.

[10] J. Herre and B. Grill, "Overview of MPEG-4 audio and its applications in mobile communications," in *WCC 2000 - ICCT 2000*, vol. 1, Aug 2000, pp. 604–613.

[11] ISO/IEC, 14496-3, "Information technology – Coding of audio-visual objects – Part 3: Audio," 2009.

[12] T. Liebchen and Y. Reznik, "MPEG-4 ALS: an emerging standard for lossless audio coding," in *Proc. DCC*, 2004, pp. 439–448.

[13] Y. Kanai, S. J. Greaves, H. Muraoka, and I. Tagawa, "Shingled Perpendicular Magnetic Recording and Suitable Write Heads (in Japanese)," *Magnetics Japan*, vol. 5, no. 7, pp. 320–331, Jul. 2010.

[14] *NE Handbook Sensor Networks.* GSM Association, 2017.

[15] *An Introduction to Network Slicing.* GSM Association, 2017.

[16] ITU-T Rec. G. 726, "40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)," 1990.

[17] ITU-T Rec. G. 729, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)," 1996.

[18] "3GPP TS 26.290 Release 15 - Audio codec processing functions; Extended Adaptive Multi-Rate - Wideband (AMR-WB+) codec; Transcoding functions," 3GPP, 2018.

[19] ISO/IEC, 23003-3, "Information technology – Information technology – MPEG audio technologies – Part 3: Unified speech and audio coding," 2012.

[20] S. Quackenbush, "MPEG Unified Speech and Audio Coding," *IEEE MultiMedia*, vol. 20, no. 2, pp. 72–78, Apr. 2013.

[21] M. Neuendorf, M. Multrus, N. Rettelbach, G. Fuchs, J. Robilliard, J. Lecomte, S. Wilde, S. Bayer, S.and Disch, C. Helmrich, R. Lefebvre, P. Gournay, B. Bessette, J. Lapierre, K. Kjörling, H. Purnhagen, L. Villemoes, W. Oomen, E. Schuijers, K. Kikuiri, T. Chinen, T. Norimatsu, C. K. Seng, E. Oh, M. Kim, S. Quackenbush, and B. Grill, "MPEG Unified Speech and Audio Coding - The ISO/MPEG Standard for High-Efficiency Audio Coding of All Content Types," in *Proc. AES 132nd Convention, #8654*, Apr. 2012.

[22] ISO/IEC, 11172-3, "Information technology – Coding of moving pictures and asso-

ciated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio," 1993.

[23] ISO/IEC, 13818-3, "Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio," 1998.

[24] M. Schnell, R. Geiger, M. Schmidt, M. Jander, M. Multrus, G. Schuller, and J. Herre, "Enhanced MPEG-4 Low Delay AAC-Low Bitrate High Quality Communication," in *Proc. AES 122nd Convention, #6998*, 2007.

[25] A. Raake, *Speech quality of VOIP Assesment and prediction.* John Wiley & Sons, Ltd, 2006.

[26] K. Sayood, *Introduction to Data Compression.* Elsevier, ch. 11, year =2006.

[27] D. Malah and R. Cox, "A generalized comb filtering technique for speech enhancement," in *Proc. ICASSP*, vol. 7, May 1982, pp. 160–163.

[28] F. Itakura and S. Saito, "A statistical method for estimation of speech spectral density and formant frequencies," *Electron. Commun.*, vol. 53-A, no. 1, pp. 36–43, 1970.

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

[30] T. Bäckström, *Speech Coding with Code Excited Linear Prediction.* Springer, 2017.

[31] T. Berger, *Rate distortion theory : a mathematical basis for data compression.* Springer Netherlands, 1999.

[32] G. Fuchs, C. Helmrich, G. Markovic, M. Neusinger, E. Ravelli, and T. Moriya, "Low delay LPC and MDCT-based audio coding in the EVS codec," in *Proc. ICASSP*, Apr. 2015, pp. 5723–5727.

[33] R. Geiger, J. Herre, H. Huang, X. Lin, S. Rahardja, and R. Yu, "MPEG-4 Scalable to Lossless Audio Coding," in *Proc. AES 117th Convention, #6183*, 2004.

[34] "3GPP TS 26.445 Release 12," 3GPP, 2014.

[35] M. Schroeder and B. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," in *Proc. ICASSP*, vol. 10, Apr. 1985, pp. 937–940.

[36] C. Laflamme, J. . Adoul, H. Y. Su, and S. Morissette, "On reducing computational complexity of codebook search in CELP coder through the use of algebraic codes," in *Proc. ICASSP*, vol. 1, Apr. 1990, pp. 177–180.

[37] F. Itakura, "Line spectrum representation of linear predictor coefficients of speech signals," *The Journal of the Acoustical Society of America*, vol. 57, no. 6, p. S35, 1975.

[38] N. Sugamura and F. Itakura, "Speech analysis and synthesis methods developed at ECL in NTT -From LPC to LSP-," *Speech Communication*, vol. 5, no. 2, pp. 199–215, 1986.

[39] A. J. Bower, "Digital two-channel sound for terrestrial television," *IEEE Trans. on Consumer Electronics,*, vol. CE-33, no. 3, pp. 286–296, 1987.

[40] N. Iwakami, T. Moriya, and S. Miki, "High-quality audio-coding at less than 64 kbit/s by using transform-domain weighted interleave vector quantization (TwinVQ)," in *Proc. ICASSP*, vol. 5, May 1995, pp. 3095–3098.

[41] S. W. Golomb, "Run-length encodings," *IEEE Trans. on Information Theory*, vol. 12, pp. 399–401, Jun. 1966.

[42] R. F. Rice, "Some practical universal noiseless coding techniques - part I-III," *Jet Propulsion Laboratory Technical Report*, vol. JPL-79-22, JPL-83-17, JPL-91-3, pp. 1509–1520, 1979, 1983, 1991.

[43] H. Yokoo, "An improved Ziv-Lempel coding scheme for universal source coding," *Electronics and Communications in Japan (Part I: Communications)*, vol. 69, pp. 12–20, 1986.

[44] P. Tischer, "A Modified Lempel-Ziv-Welch Data Compression Scheme," *Australian Computer Science Communications*, vol. 9, no. 1, pp. 262–272, 1987.

[45] E. R. Fiala and D. H. Greene, "Data Compression with Finite Windows," *Comm. of the ACM*, vol. 32, no. 4, pp. 490–505, Apr. 1989.

[46] Salomon, S. and Motta, G., *Handbook of Data Compression.* Springer, 2010.

[47] H. Yamamoto, "Data Compression with Finite Windows," *J. of Institute of Inform. Processing of Japan*, vol. 35, no. 7, pp. 600–608, Jul. 1994 (in Japanese).

[48] T. Robinson, "SHORTEN: Simple lossless and near-lossless waveform compression," *Cambridge Univ. Eng. Dept., Cambridg, UK, Tech. Rep. 156,*, 1994.

[49] T. Liebchen, Y. Reznik, T. Moriya, and D. T. Yang, "MPEG-4 audio lossless coding," in *Proc. AES 116th Convention, #6047*, Jun. 2004.

[50] N. Harada, Y. Kamamoto, T. Moriya, Y. Hiwasaki, M. A. Ramalho, L. Netsch, J. Stachurski, L. Miao, H. Taddei, and F. Qi, "Emerging ITU-T standard G.711.0 -

lossless compression of G.711 pulse code modulation," in *Proc. ICASSP*, 2010, pp. 4658–4661.

[51] J. Harrington and S. Cassidy, *Techniques in Speech Acoustics.* Prentice-Hall, 1971.

[52] H. W. Strube, "Linear prediction on a warped frequency scale," vol. 68, no. 4, pp. 1071–1076, 1980.

[53] K. Tokuda, T. Kobayashi, S. Imai, and T. Chiba, "Spectral estimation of speech by mel-generalized cepstral analysis," *Electron. Commun. Japan*, vol. 3, no. 2, pp. 30–431, 1993.

[54] S. Wabnik, G. Schuller, U. Kramer, and J. Hirschfeld, "Frequency warping in low delay audio coding," in *Proc. ICASSP*, vol. 3, no. 3, Mar. 2005, pp. 181–184.

[55] K. Koishida, K. Tokuda, T. Kobayashi, and S. Imai, "Efficient encoding of mel-generalized cepstrum for CELP coders," in *Proc. ICASSP*, vol. 2, May 1997, pp. 1355–1358.

[56] K. Koishida, G. Hirabayashi, K. Tokuda, and T. Kobayashi, "A Wideband Celp Speech Coder At 16 Kbit/s Based On Mel-Generalized Cepstral Analysis," pp. 161–164, Aug. 1998.

[57] Y. Nakatoh, T. Norimatsu, A. H. Low, and H. Matsumoto, "Low bit rate coding for speech and audio using mel linear predictive coding (MLPC) analysis," 1998.

[58] H. Kameoka, M. Goto, and S. Sagayama, "Selective amplifier of periodic and non-periodic components in concurrent audio signals with spectral control envelopes," in *IPSJ SIG Technical Reports*, vol. 2006-MUS-66, Aug. 2006 (in Japanese), pp. 77–84.

[59] R. Yu, X. Lin, S. Rahardja, and C. Ko, "A statistics study of the mdct coefficient distribution for audio," in *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, vol. 2, Jun. 2004, pp. 1483–1486.

[60] M. Oger, S. Ragot, and M. Antonini, "Transform audio coding with arithmetic-coded scalar quantization and model-based bit allocation," in *Proc. ICASSP*, vol. 4, Apr. 2007, pp. IV–545–IV–548.

[61] C. Bouman and K. Sauer, "A generalized gaussian image model for edge-preserving map estimation," *IEEE Trans. on Image Processing*, vol. 2, no. 3, pp. 296–310, Jul. 1993.

[62] P. Moulin and L. Juan, "Analysis of multiresolution image denoising schemes using generalized gaussian and complexity priors," *IEEE Trans. on Information Theory*,

vol. 45, no. 3, pp. 909–919, Apr. 1999.

[63] C. Parisot, M. Antonini, and M. Barlaud, "3D scan-based wavelet transform and quality control for video coding," *EURASIP Journal on Applied Signal Processing*, pp. 56–65, 2003.

[64] S. Yu, A. Zhang, and H. Li, "A review of estimating the shape parameter of generalized gaussian distribution," *Journal of Computational Information Systems*, vol. 8, no. 21, pp. 9055–9064, 2012.

[65] H. Kameoka, Y. Kamamoto, N. Harada, and T. Moriya, "A linear predictive coding algorithm minimizing the Golomb-Rice code length of the residual signal," *IEICE Trans. on Fundamentals of Electronics*, vol. J91-A, no. 11, pp. 1017–1025, Jul. 2008 (in Japanese).

[66] Y. Kamamoto, *Efficient lossless coding of multichannel signal based on time-space linear predictive model.* Graduate school of Information Science and Technology, The University of Tokyo, 2012.

[67] D. Mansour and B. H. Juang, "The short-time modified coherence representation and its application for noisy speech recognition," in *Proc. ICASSP*, vol. 1, Apr. 1988, pp. 525–528.

[68] W. Thanhikam, Y. Kamamori, A. Kawamura, and Y. Iiguni, "Stationary and Non-stationary Wide-Band Noise Reduction Using Zero Phase Signal," *IEICE Trans. on Fundamentals of Electronics*, vol. E95-A, no. 5, pp. 843–852, May 2012.

[69] A. Kawamura, "A restricted impact noise suppressor in zero phase domain," in *Proc. EUSIPCO*, Sep. 2014, pp. 681–685.

[70] D. R. González, E. L. Solano, and J. R. C. de Lara, "Zero phase speech representation for robust formant tracking," in *Proc. EUSIPCO*, Sep. 2014, pp. 1462–1466.

[71] F. Pascal, L. Bombrun, J. Y. Tourneret, and Y. Berthoumieu, "Parameter estimation for multivariate generalized gaussian distributions," *IEEE Trans. on Signal Processing*, vol. 61, no. 23, pp. 5960–5971, Dec. 2013.

[72] G. D. T. Schuller, B. Yu, D. Huang, and B. Edler, "Perceptual audio coding using adaptive preand post-filters and lossless compression," *IEEE/ACM Trans. on Speech and Audio Processing,*, vol. 10, no. 6, pp. 379–389, 2002.

[73] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis - A unified approach to speech spectral estimation," in *Proc. ICSLP*, 1994, pp.

18–22.

[74] Moffat, A. and Turpin, A., *Compression and coding algorithms.* Kluwer Academic Publishers, 2002.

[75] J. Muramatsu, "Channel Coding and Lossy Source Coding Using a Generator of Constrained Random Numbers," *IEEE Trans. on Information Theory*, vol. 60, no. 5, pp. 2667–2686, May 2014.

[76] ——, "Variable-length lossy source code using a constrained-random-number-generator," *IEEE Trans. on Information Theory*, vol. 61, no. 6, pp. 3574–3592, Jun. 2015.

[77] M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A low complexity, context-based, lossless image compression algorithm," in *Proc. DCC*, 1996, pp. 140–149.

[78] N. Harada, Y. Kamamoto, and T. Moriya, "Escaped-Huffman and adaptive recursive rice coding for lossless compression of the mapped domain linear prediction residual," in *Proc. ICASSP*, Mar. 2010, pp. 4646–4649.

[79] S. Xue, Y. Xu, and B. Oelmann, "Hybrid Golomb codes for a group of quantised GG sources," *IEE Proceedings - Vision, Image and Signal Processing*, vol. 150, no. 4, pp. 256–260, Aug. 2003.

[80] T. Bäckström and C. Helmrich, "Arithmetic coding of speech and audio spectra using tcx based on linear predictive spectral envelopes," in *Proc. ICASSP*, Apr. 2015, pp. 5127–5131.

[81] R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, "Golomb-rice coding optimized via LPC for frequency domain audio coder," in *Proc. GlobalSIP*, Dec. 2014, pp. 1024–1028.

[82] ITU-T Rec. T.87, "Information technology – Lossless and near-lossless compression of continuous-tone still images – Baseline ," 1998.

[83] N. Merhav, G. Seroussi, and M. J. Weinberger, "Optimal prefix codes for sources with two-sided geometric distributions," *IEEE Trans. on Information Theory*, vol. 46, no. 1, pp. 121–135, Jan. 2000.

[84] R. Malvar, "Adaptive Run-Length / Golomb-Rice Encoding of Quantized Generalized Gaussian Sources with Unknown Statistics," in *Proc. DCC.* Institute of Electrical and Electronics Engineers, Inc., Mar. 2006.

[85] J. Wen and J. D. Villasenor, "Structured prefix codes for quantized low-shape-

parameter generalized Gaussian sources," *IEEE Trans. on Information Theory*, vol. 45, no. 4, pp. 1307–1314, May 1999.

[86] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Trans. on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[87] E. Ordentlich, M. Weinberger, and G. Seroussi, "A low-complexity modeling approach for embedded coding of wavelet coefficients," in *Proc. DCC*, Mar. 1998, pp. 408–417.

[88] H. Yamamoto, M. Tsuchihashi, and J. Honda, "Almost Instantaneous Fixed-to-Variable Length Codes," *IEEE Trans. on Information Theory*, vol. 61, no. 12, pp. 6432–6443, Dec 2015.

[89] W. Hu, H. Yamamoto, and J. Honda, "Worst-case redundancy of optimal binary aifv codes and their extended codes," *IEEE Trans. on Information Theory*, vol. 63, no. 8, pp. 5074–5086, Aug 2017.

[90] H. Yamamoto and H. Yokoo, "Average-sense optimality and competitive optimality for almost instantaneous VF codes," *IEEE Trans. on Information Theory*, vol. 47, no. 6, pp. 2174–2184, Sep 2001.

[91] L. Devroye, *Non-Uniform Random Variate Generation.* Springer, 1986, pp. 27–34.

[92] R. Sugiura, Y. Kamamoto, N. Harada, and T. Moriya, "Optimal Golomb-Rice Code Extension for Lossless Coding of Low-Entropy Exponentially Distributed Sources," *IEEE Trans. on Information Theory*, vol. 64, no. 4, pp. 3153 – 3161, Apr. 2018.

[93] Y. A. Reznik, "Coding of prediction residual in MPEG-4 standard for lossless audio coding (MPEG-4 ALS)," in *Proc. ICASSP*, vol. 3, no. 3, May 2004, pp. 1024–1027.

[94] ——, "Low-complexity lossless codes for image and video coding," in *SPIE 7798*, vol. 7798, Sep. 2010, pp. 1–9.

[95] S. Takamura and Y. Yashima, "Gaussian Golomb Codes," in *Proc. DCC*, Mar. 2007, pp. 401–401.

[96] H. Ohmuro, T. Moriya, K. Mano, and S. Miki, "Vector quantization of LSP parameters using moving average interframe prediction," *Electron. Commun. Japan*, vol. 77, no. 10, pp. 12–26, 1994.

[97] "VoiceAge - AMR-WB+," [Online]. Available: http://www.voiceage.com/AMR-WBplus.html (as of Apr. '19).

[98] "ITU-T Software Tool Library 2009 User's Manual," 2009.

[99] "Speech Signal Processing Toolkit (SPTK) Version 3.11," [Online]. Available: http://sp-tk.sourceforge.net/ (as of Apr. '19).

[100] "AFsp PQevalAudio," [Online]. Available: http://www-mmsp.ece.mcgill.ca/Documents/Software/ (as of Apr. '19).

[101] M. Goto, "Development of the RWC music database," in *the 18th International Congress on Acoustics (ICA 2004)*, vol. 1, 2004, pp. 553–556.

[102] I.-R. BS.1534-1, "Method for the subjective assessment of intermediate quality level of coding systems," 2001.

[103] Digital Compression and Coding of Continuoustone Still Images, "Part 1, requirements and guidelines," iSO/IEC JTC1 Draft International Standard 10918-1, Nov. 1991.

[104] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "Atr japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, 1990.

[105] R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, "Optimal Coding of Generalized-Gaussian-Distributed Frequency Spectra for Low-Delay Audio Coder With Powered All-Pole Spectrum Estimation," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 8, pp. 1309–1321, Aug. 2015.

[106] R. Sugiura, Y. Kamamoto, and T. Moriya, "Shape parameter estimation for generalized-Gaussian-distributed frequency spectra of audio signals," in *Proc. ICASSP*, Mar. 2017, pp. 736–740.

[107] R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, "Resolution warped spectral representation for low-delay and low-bit-rate audio coder," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 288–299, 2015.

[108] [Online]. Available: http://www.soumu.go.jp/main_content/000418409.pdf (in Japanese, as of Apr. '19).

[109] [Online]. Available: https://flets-w.com/opt/hikaridenwa/service/dataconnect/ (in Japanese, as of Apr. '19).

[110] K. Campbell, "Deploying large scale audio over ip networks," in *Proc. AES 126th Convention, #7652*, 2009.

[111] "ISO/IEC 14496-3:2009 information technology – coding of audio-visual objects – part 3: Audio (4th edition)." ISO/IEC, 2009.

[112] J. van der Meer, D. Mackie, V. Swaminathan, D. Singer, and P. Gentric, "RTP Payload Format for Transport of MPEG-4 Elementary Streams," IETF 3640, 2003.

[113] M. Schmidt, F. de Bont, S. Doehla, and J. Kim, "RTP Payload Format for MPEG-4 Audio/Visual Streams," IETF 6416, 2011.

[114] I.-R. BS.1116, "Methods for the subjective assessment of small impairments in audio systems," 1994.

[115] "HDIP-3000V," [Online]. Available: https://www.rf-design.co.jp/products/detail.php?product_id=31 (in Japanese, as of Apr. '19).

[116] ITU-T Rec. G. 728, "Coding of speech at 16 kbit/s using low-delay code excited linear prediction," 2013.

[117] T. Moriya, S. Miki, K. Mano, and H. Ohmuro, "Pitch synchronous innovation celp (psi-celp)," *IEICE Trans. on Fundamentals of Electronics*, vol. E76-A, no. 7, pp. 1177–1180, Jul. 1993.

[118] B.-H. Juang and A. Gray, "Multiple stage vector quantization for speech coding," in *Proc. ICASSP*, vol. 7, May 1982, pp. 597–600.

[119] T. Moriya, "Two-channel conjugate vector quantizer for noisy channel speech coding," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 5, pp. 866–874, Jun. 1992.

[120] R. Zelinski and P. Noll, "Adaptive transform coding of speech signals," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 25, no. 4, pp. 299–309, Aug. 1977.

[121] T. Li, S. Rahardja, and S. N. Koh, "Perceptually Prioritized Bit-Plane Coding for High-Definition Advanced Audio Coding," in *Eighth IEEE International Symposium on Multimedia*, Dec 2006, pp. 245–252.

[122] W. Kleijn and K. Paliwal, *Speech coding and synthesis.* Elsevier, 1995, pp. 433–466.

[123] ITU-T Rec. P.800, "Methods for subjective determination of transmission quality," iTU-T, 1992.

A

Equivalence between square error and IS divergence

Here we represent the envelope as $H_k^2 = \sigma^2 h_k^2$ with

$$h_k^2 \equiv \left| 1 + \sum_{n=1}^{p} a_n e^{-\mathrm{j}\frac{\pi n k}{N}} \right|^{-2}. \tag{A.1}$$

The minimization problem of the LP with frame length $2N$ can be equivalently written, using the Parseval's theorem, as

$$\min_{\{a_n\}} \mathcal{L}_{\mathrm{MMSE}} = \min_{\{a_n\}} \frac{1}{2N} \sum_{k=0}^{2N-1} \frac{|X_k|^2}{h_k^2}, \tag{A.2}$$

with $\{X_k\}_{k=0}^{2N-1}$ regarded as the DFT spectra of a real-valued signal. We respectively use $\mathcal{L}_{\mathrm{MMSE}}$ and $\mathcal{L}_{\mathrm{ML}}$ here to represent the objective functions of the LP and the IS divergence in order to make it clear that the functions corresponds to the Minimum Mean Squared Error (MMSE) criterion and the Maximum Likelihood (ML) criterion, respectively. Since $|X_k|^2 = |X_{2N-k}|^2$ where $1 \leq k \leq N-1$, ignoring the edge effects of $\{|X_k|^2\}_{k=0}^{N-1}$, namely assuming $|X_0|^2, |X_N|^2 \ll N$, the objective function can be approximated by the variance given from half of the frame as

$$\mathcal{L}_{\mathrm{MMSE}} \approx \frac{1}{N} \sum_{k=0}^{N-1} \frac{|X_k|^2}{h_k^2} \equiv \sigma^2. \tag{A.3}$$

On the other hand, the IS divergence discussed in this paper can be transformed as

$$\begin{aligned}\mathcal{L}_{\mathrm{ML}} &= \sum_{k=0}^{N-1} D_{\mathrm{IS}}(H_k^2 \mid |X_k|^2) \qquad \text{(A.4)}\\ &= \sum_{k=0}^{N-1} \left(\frac{|X_k|^2}{H_k^2} - \ln \frac{|X_k|^2}{H_k^2} - 1 \right) \\ &= \frac{1}{\sigma^2} \sum_{k=0}^{N-1} \frac{|X_k|^2}{h_k^2} - \sum_{k=0}^{N-1} \ln \frac{|X_k|^2}{H_k^2} - N \\ &= - \sum_{k=0}^{N-1} \ln \frac{|X_k|^2}{H_k^2} \\ &= N \left(\ln \sigma^2 - \ln \left(\prod_{k=0}^{N-1} h_k^2 \right)^{\frac{1}{N}} \right) + C(\{X_k\})\end{aligned}$$

where $C(\{X_k\})$ is a constant for $\{a_n\}_{n=1}^p$. Additionally ignoring the edge effects of $\{h_k\}_{k=0}^{N-1}$, it becomes as

$$\mathcal{L}_{\mathrm{ML}} \approx N \left(\ln \sigma^2 - \ln \left(\prod_{k=0}^{2N-1} h_k^2 \right)^{\frac{1}{2N}} \right) + C(\{X_k\}). \qquad \text{(A.5)}$$

Therefore, if we can prove $(\prod_{k=0}^{2N-1} h_k^2)^{\frac{1}{2N}} = 1$, we can say that the two objective functions satisfies

$$\arg\min_{\{a_n\}} \mathcal{L}_{\mathrm{MMSE}} = \arg\min_{\{a_n\}} \mathcal{L}_{\mathrm{ML}}, \qquad \text{(A.6)}$$

and the equivalence of the problems will be proved. The rest of this section shows the proof for the statement $(\prod_{k=0}^{2N-1} h_k^2)^{\frac{1}{2N}} = 1$.

It should be noted that the statement is trivial when using continuous signals because the complex line integral of logarithmic function around the unit circle becomes zero. However, when it comes to discrete signals, we have to consider a little more carefully. Taking into account that $\{h_k\}_{k=0}^{2N-1}$ are composed of Fourier series, the envelope $\{h_k^2\}_{k=0}^{2N-1}$ can be interpreted, when the coefficients $\{a_n\}_{n=1}^p$ are stable, as the eigenvalues of a $2N \times$

$2N$ matrix $(\Phi^{\mathsf{T}}\Phi)^{-1}$ represented by a circulant matrix:

$$\Phi = \begin{bmatrix} 1 & & & a_p & \dots & a_1 \\ a_1 & 1 & & & \ddots & \vdots \\ \vdots & \ddots & \ddots & & & a_p \\ a_p & & \ddots & \ddots & & \\ & \ddots & & \ddots & \ddots & \\ & & a_p & \dots & a_1 & 1 \end{bmatrix}. \tag{A.7}$$

Thus, the geometric mean of $\{h_k^2\}_{k=0}^{2N-1}$ is given from the determinant of $(\Phi^{\mathsf{T}}\Phi)^{-1}$:

$$\left(\prod_{l=0}^{2N-1} h_k^2\right)^{\frac{1}{2N}} = (\det(\Phi^{\mathsf{T}}\Phi)^{-1})^{\frac{1}{2N}} = (\det\Phi)^{-\frac{1}{N}}. \tag{A.8}$$

Hereinafter, without loss of generality, we regard $2N$ and p as powers of two. To evaluate $(\det\Phi)^{-\frac{1}{N}}$, we divide the matrix Φ as

$$\left[\begin{array}{cccccc|cccccc} 1 & & & & & & & & & a_p & \dots & a_1 \\ a_1 & 1 & & & & & & & & & \ddots & \vdots \\ \vdots & \ddots & & & & & & & & & & a_p \\ a_p & & \ddots & \ddots & & & & & & & & \\ & \ddots & & \ddots & \ddots & & & & & & & \\ & & a_p & \dots & a_1 & 1 & & & & & & \\ \hline & & & a_p & \dots & a_1 & 1 & & & & & \\ & & & & \ddots & \vdots & a_1 & 1 & & & & \\ & & & & & a_p & \vdots & \ddots & & & & \\ & & & & & & a_p & & \ddots & \ddots & & \\ & & & & & & & \ddots & & \ddots & \ddots & \\ & & & & & & & & a_p & \dots & a_1 & 1 \end{array}\right],$$

in other words,

$$\Phi = \begin{bmatrix} P_{N\times N}^{(p)} & Q_{N\times N}^{(p)} \\ Q_{N\times N}^{(p)} & P_{N\times N}^{(p)} \end{bmatrix}. \tag{A.9}$$

We first evaluate the upper-bound of the determinant. Since $\det(P+Q) \geq \det P + \det Q$ holds for positive-semidefinite matrices P and Q, and $P_{N\times N}^{(p)}$ is a lower triangular matrix with all of its diagonal elements valued one, the determinant satisfies

$$\begin{aligned}\det \Phi &= \left|\left(\begin{bmatrix} P^{(p)}_{N\times N} & 0 \\ Q^{(p)}_{N\times N} & P^{(p)}_{N\times N} \end{bmatrix} + \begin{bmatrix} 0 & Q^{(p)}_{N\times N} \\ 0 & 0 \end{bmatrix}\right)\right| \\ &\geq \begin{vmatrix} P^{(p)}_{N\times N} & 0 \\ Q^{(p)}_{N\times N} & P^{(p)}_{N\times N} \end{vmatrix} + \begin{vmatrix} 0 & Q^{(p)}_{N\times N} \\ 0 & 0 \end{vmatrix} \\ &= 1 + 0 = 1. \end{aligned} \tag{A.10}$$

Therefore, $(\det \Phi)^{-\frac{1}{N}} \leq 1$.

Next, for the lower-bound, we use the theorem of determinants as

$$\begin{aligned}\det \Phi &= \begin{vmatrix} P^{(p)}_{N\times N} & Q^{(p)}_{N\times N} \\ Q^{(p)}_{N\times N} & P^{(p)}_{N\times N} \end{vmatrix} \\ &= \left|P^{(p)}_{N\times N}\right| \cdot \left|P^{(p)}_{N\times N} - Q^{(p)}_{N\times N}\left(P^{(p)}_{N\times N}\right)^{-1} Q^{(p)}_{N\times N}\right| \\ &= \left|P^{(p)}_{N\times N} - Q^{(p)}_{N\times N}\left(P^{(p)}_{N\times N}\right)^{-1} Q^{(p)}_{N\times N}\right|. \end{aligned} \tag{A.11}$$

In accordance with the structure of $P^{(p)}_{N\times N}$ and $Q^{(p)}_{N\times N}$, the $N \times N$ matrix in Eq. (A.11) can be written, using $p \times p$ matrix $c^{(N/2)}_{p\times p}$, as

$$P^{(p)}_{N\times N} - Q^{(p)}_{N\times N}\left(P^{(p)}_{N\times N}\right)^{-1} Q^{(p)}_{N\times N} =$$

$$\left[\begin{array}{cccccc|cccccc}
1 & & & & & & & & & & & \\
a_1 & 1 & & & & & & & & & c^{(N/2)}_{p\times p} & \\
\vdots & \ddots & & & & & & & & & & \\
a_p & & \ddots & \ddots & & & & & & & & \\
 & \ddots & & \ddots & \ddots & & & & & & & \\
 & & a_p & \cdots & a_1 & 1 & & & & & & \\
\hline
 & & & a_p & \cdots & a_1 & 1 & & & & & \\
 & & & & \ddots & \vdots & a_1 & 1 & & & & \\
 & & & & & a_p & \vdots & \ddots & & & & \\
 & & & & & & a_p & & \ddots & \ddots & & \\
 & & & & & & & \ddots & & \ddots & \ddots & \\
 & & & & & & & & a_p & \cdots & a_1 & 1
\end{array}\right]$$

$$\equiv \begin{bmatrix} P^{(p)}_{N/2\times N/2} & C^{(p)}_{N/2\times N/2} \\ Q^{(p)}_{N/2\times N/2} & P^{(p)}_{N/2\times N/2} \end{bmatrix}, \tag{A.12}$$

and then we can rewrite the determinant using this:

$$\begin{aligned}&\det\Phi \\ &= \left|P^{(p)}_{N/2\times N/2} - Q^{(p)}_{N/2\times N/2}\left(P^{(p)}_{N/2\times N/2}\right)^{-1} C^{(p)}_{N/2\times N/2}\right|.\end{aligned} \tag{A.13}$$

The $N/2 \times N/2$ matrix $Q^{(p)}_{N/2\times N/2}\left(P^{(p)}_{N/2\times N/2}\right)^{-1} C^{(p)}_{N/2\times N/2}$ has non-zero elements in the same place as $C^{(p)}_{N/2\times N/2}$ so that the transformation of the determinant from Eq. (A.11) to Eq. (A.13) can be repeated until the matrix in the equation becomes $2p \times 2p$:

$$\det\Phi = \begin{vmatrix} P^{(p)}_{p\times p} & c^{(p)}_{p\times p} \\ Q^{(p)}_{p\times p} & P^{(p)}_{p\times p} \end{vmatrix}. \tag{A.14}$$

Finally applying to this equation the Hadamard's inequality, it comes as

$$(\det\Phi) \leq \left(\prod_{i=1}^{2p}\sum_{j=1}^{2p}\Phi^2_{ij}\right)^{\frac{1}{2}} \leq \left(2p|a_{\max}|^2\right)^p, \tag{A.15}$$

where Φ_{ij} and $a_{\max}$ are respectively the (i,j) element of Φ and the maximum of $\{1, |a_1|, ..., |a_p|\}$, resulting in

$$(\det\Phi)^{-\frac{1}{N}} \geq \left(2p|a_{\max}|^2\right)^{-\frac{p}{N}}. \tag{A.16}$$

From the above, the determinant is proved to hold

$$\left(2p|a_{\max}|^2\right)^{-\frac{p}{N}} \leq (\det\Phi)^{-\frac{1}{N}} \leq 1, \tag{A.17}$$

and since $\left(2p|a_{\max}|^2\right)^{-\frac{p}{N}} = 1$ when $p \ll N$, the squeeze theorem leads to

$$\lim_{p/N\to 0}\left(\prod_{k=0}^{2N-1} h^2_k\right)^{\frac{1}{2N}} = \lim_{p/N\to 0}(\det\Phi)^{-\frac{1}{N}} = 1. \tag{A.18}$$

Therefore, the minimization problem of the LP is approximately equivalent to the minimization of IS divergence when the frame length N is sufficiently larger than the prediction order p.

B

List of related publications

B.1 Journal papers

J1 R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, “Resolution Warped Spectral Representation for Low-Delay and Low-Bit-Rate Audio Coder,” IEEE/ACM Trans. on Audio, Speech, and Language Processing, vol. 23, no. 2, pp. 288-299, Feb., 2015.

J2 R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, “Optimal Coding of Generalized-Gaussian-Distributed Frequency Spectra for Low-Delay Audio Coder With Powered All-Pole Spectrum Estimation,” IEEE/ACM Trans. on Audio, Speech, and Language Processing, vol. 23, no. 8, pp. 1309-1321, Aug. 2015.

J3 R. Sugiura, Y. Kamamoto, N. Harada, and T. Moriya, “Optimal Golomb–Rice Code Extension for Lossless Coding of Low-Entropy Exponentially Distributed Sources,” IEEE Trans. on Information Theory, vol. 64, no. 4, pp. 3153-3161, Apr., 2018.

J4 R. Sugiura, Y. Kamamoto, and T. Moriya, “Shape Control of Discrete Generalized Gaussian Distributions for Frequency-Domain Audio Coding,” IEEE/ACM Trans. on Audio, Speech, and Language Processing, vol. 27, no. 12, pp. 2234-2248, Dec., 2019.

J5 R. Sugiura, Y. Kamamoto, and T. Moriya, “Extended-domain Golomb code and symmetry of relative redundancy,” IEICE Trans. on Fundamentals of Electronics, 2021 (Accepted).

B.2 Peer-reviewed international conferences

I1 R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, “Representation of spectral envelope with warped frequency resolution for audio coder,” in Proc. EUSIPCO, vol. TU-L03-1, pp. 1-5, Sep., 2014.

I2 R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, “Direct linear conversion of LSP parameters for perceptual control in speech and audio coding,” in Proc. EUSIPCO, vol. TU-L03-2, pp. 1-5, Sep., 2014.

I3 R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, “Golomb–Rice Coding Optimized via LPC for Frequency Domain Audio Coder,” in Proc. GlobalSIP, pp. 1196-1200, Dec., 2014.

I4 R. Sugiura, Y. Kamamoto, and T. Moriya, “Shape Parameter Estimation for Generalized-Gaussian-Distributed Frequency Spectra of Audio Signals,” in Proc. ICASSP, pp. 736-740, Mar., 2017.

I5 R. Sugiura, Y. Kamamoto, and T. Moriya, “CLEAR: Conditionally Lossless Encoding under Allowed Rates for low-delay sound data transmission,” in Proc. AES 143rd convention, #9899, Oct., 2017.

I6 R. Sugiura, Y. Kamamoto, and T. Moriya, “Integer Nesting/Splitting for Golomb–Rice Coding of Generalized Gaussian Sources,” in Proc. DCC, pp. 427-427, Mar., 2018.

I7 R. Sugiura, Y. Kamamoto, and T. Moriya, “Spectral-Envelope-Based Least Significant Bit Management for Low-Delay Bit-Error-Robust Speech Coding,” in Proc. ICASSP, pp. 671-675, Apr., 2018.

B.3 Domestic conferences and technical report

D1 R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, “Frequency warped spectral envelope representation for audio coding,” in Proc. ASJ, no. 1-6-8, pp.281-284, Mar., 2014 (in Japanese).

D2 R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, “Frequency-warped representation of spectral envelope for audio coding, ” IPSJ SIG Technical

Report, vol. 2014-MUS-103, no. 39, pp.1-5, May, 2014 (in Japanese).

D3 R. Sugiura, Y. Kamamoto, N. Harada, H. Kameoka, and T. Moriya, “Optimal representation of spectral envelope for Golomb–Rice coding,” in Proc. ASJ, Mar. 2015 (in Japanese).

D4 K. Hasegawa, K. Noguchi, T. Morizumi, J. Nakajima, T. Kanekiyo, R. Sugiura, Y. Kamamoto, T. Moriya, H. Oda, and O. Kawasaki, ”Prototyping of Low-Delay Audio Codec Appliance for Bandlimited Network,” ITE technical report 41 (23), 37-40, Jul., 2017.

D5 Y. Kamamoto, R. Sugiura, T. Moriya, K. Furukado, K. Noguchi, and T. Kanekiyo, ”Overview and experimental evaluation of CLEAR (Conditionally Lossless Encoding under Allowed Rate),” in Proc. ASJ, no. 3-12-1, pp.491-492, Sep., 2017 (in Japanese).

D6 R. Sugiura, Y. Kamamoto, and T. Moriya, ”Technical review on Conditionally Lossless Encoding under Allowed Rates (CLEAR),” in Proc. ASJ, no. 3-12-2, pp.493-494, Sep., 2017 (in Japanese).

D7 K. Noguchi, K. Hasegawa, J. Nakajima, T. Morizumi, T. Kanekiyo, R. Sugiura, Y. Kamamoto, and T. Moriya, ”Implementation of CLEAR (Conditionally Lossless Encoding under Allowed Rates),” in Proc. ASJ, no. 3-12-3, pp.495-496, Sep., 2017 (in Japanese).

D8 R. Sugiura, Y. Kamamoto, and T. Moriya, ”Bit-error-robust low-delay speech encoding,” in Proc. ASJ, no. 2-4-8, pp.1147-1148, Sep., 2018 (in Japanese).

B.4 International registered patents

PI1 ”FREQUENCY DOMAIN PARAMETER SEQUENCE GENERATION METHOD, CODING METHOD, DECODING METHOD, FREQUENCY DOMAIN PARAMETER SEQUENCE GENERATION DEVICE, CODING DEVICE, DECODING DEVICE, PROGRAM, AND RECORDING MEDIUM,” PCT/JP2015/054135, EP3136387, US20170249947, Oct., 2015.

PI2 ”ENCODING DEVICE, DECODING DEVICE, AND METHOD AND PROGRAM FOR SAME,” PCT/JP2015/082839, US20170272766, Nov., 2015.

PI3 ”PARAMETER DETERMINATION DEVICE, METHOD, PROGRAM, AND

RECORDING MEDIUM,” PCT/JP2016/052362, US20180268843, Jan., 2016.

PI4 ”ENCODING DEVICE, DECODING DEVICE, METHODS THEREFOR, PROGRAM, AND RECORDING MEDIUM,” PCT/JP2016/052365, US20180047401, Jan., 2016.

PI5 ”LINEAR PREDICTIVE CODING DEVICE, LINEAR PREDICTIVE DECODING DEVICE, AND METHOD, PROGRAM, AND RECORDING MEDIUM THEREFOR,” PCT/JP2016/061682, US20180096694, Apr., 2016.

PI6 ”MATCHING DEVICE, DETERMINATION DEVICE, METHOD THEREFOR, PROGRAM, AND RECORDING MEDIUM,” PCT/JP2016/061683, US20180090155, Apr., 2016.

B.5 Domestic registered patents

PJ1 Japanese Patent No. 6220701, Oct., 2017.
PJ2 Japanese Patent No. 6259378, Dec., 2017.
PJ3 Japanese Patent No. 6270992, Jan., 2018.
PJ4 Japanese Patent No. 6387117, Aug., 2018.
PJ5 Japanese Patent No. 6392450, Aug., 2018.
PJ6 Japanese Patent No. 6422813, Oct., 2018.
PJ7 Japanese Patent No. 6457552, Dec., 2018.
PJ8 Japanese Patent No. 6484325, Feb., 2019.
PJ9 Japanese Patent No. 6486450, Mar., 2019.
PJ10 Japanese Patent No. 6499206, Apr., 2019.
PJ11 Japanese Patent No. 6512607, Apr., 2019.
PJ12 Japanese Patent No. 6517924, Apr., 2019.
PJ13 Japanese Patent No. 6629256, Dec., 2019.
PJ14 Japanese Patent No. 6633787, Dec., 2019.
PJ15 Japanese Patent No. 6650540, Jan., 2020.
PJ16 Japanese Patent No. 6652469, Jan., 2020.
PJ17 Japanese Patent No. 6712643, Jun., 2020.
PJ18 Japanese Patent No. 6766264, Sep., 2020.
PJ19 Japanese Patent No. 6780108, Oct., 2020.

B.6 Awards

A1 IEEE Signal Processing Society Japan Student Paper Award, 2015/11/18.

A2 Telecom System Technology Student Award from the Telecommunications Advancement Foundation, 2016/03/28.

A3 IEICE 2017th Fundamentals Review Best Author Award, 2018/05/07.