CODING WITH SIDE INFORMATION

A Dissertation

by

SZE MING CHENG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2004

Major Subject: Electrical Engineering

CODING WITH SIDE INFORMATION

A Dissertation

by

SZE MING CHENG

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Zixiang Xiong (Chair of Committee)	
Costas N. Georghiades (Member)	Andrew K. Chan (Member)
Andreas Klappenecker (Member)	Chanan Singh (Head of Department)

August 2004

Major Subject: Electrical Engineering

ABSTRACT

Coding with Side Information. (August 2004)
Sze Ming Cheng, B.S., University of Hong Kong;
M.S., Hong Kong University of Science and Technology;
M.S., University of Hawaii
Chair of Advisory Committee: Zixiang Xiong

Source coding and channel coding are two important problems in communications. Although side information exists in everyday scenario, the effect of side information is not taken into account in the conventional setups. In this thesis, we focus on the practical designs of two interesting coding problems with side information: Wyner-Ziv coding (source coding with side information at the decoder) and Gel'fand-Pinsker coding (channel coding with side information at the encoder).

For WZC, we split the design problem into the two cases when the distortion of the reconstructed source is zero and when it is not. We review that the first case, which is commonly called Slepian-Wolf coding (SWC), can be implemented using conventional channel coding. Then, we detail the SWC design using the low-density parity-check (LDPC) code. To facilitate SWC design, we justify a necessary requirement that the SWC performance should be independent of the input source. We show that a sufficient condition of this requirement is that the hypothetical channel between the source and the side information satisfies a symmetry condition dubbed *dual symmetry*. Furthermore, under that dual symmetry condition, SWC design problem can be simply treated as LDPC coding design over the hypothetical channel.

When the distortion of the reconstructed source is non-zero, we propose a practical WZC paradigm called Slepian-Wolf coded quantization (SWCQ) by combining SWC and nested lattice quantization. We point out an interesting analogy between

SWCQ and entropy coded quantization in classic source coding. Furthermore, a practical scheme of SWCQ using 1-D nested lattice quantization and LDPC is implemented.

For GPC, since the actual design procedure relies on the more precise setting of the problem, we choose to investigate the design of GPC as the form of a digital watermarking problem as digital watermarking is the precise dual of WZC. We then introduce an enhanced version of the well-known spread spectrum watermarking technique. Two applications related to digital watermarking are presented.

To my parents

ACKNOWLEDGMENTS

I would like to sincerely thank my advisor Professor Zixiang Xiong for his support, guidance, encouragement and trust. He is more a mentor than a mere advisor to me. He has set a high standard for me that really helps me grow. I would like to thank Professors Andrew Chan, Costas Georghiades, and Andreas Klappenecker for spending their valuable time and effort in serving on my committee.

I want to thank my parents for their love and support, and for giving me the freedom to pursue my dream.

I would like to thank all my groupmates in the multimedia lab at TAMU. I want to thank Angelos Liveris and Dr. Vladimir Stankovic̀ for their generosity in sharing many of their ideas with me, and for proofreading several of my writings. I am greatly in debt to Jianping Hua, Zhixin Liu, Yong Sun, and Yang Yang for their help in various projects. I want to thank Tim Lan for his help regarding channel coding in general. I have received numerous help from Jianhong Jiang and Dr. Zhongmin Liu when I first moved to the South. My life would have been awful without them. There are numerous others without whom my Ph.D. years would have been both more difficult and less pleasant. I would like to thank all of them. Unfortunately, it is impossible to name them all; I sincerely apologize to those whom I omitted.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Coding in Communication

The ability to send and receive information over a long distance is a blessing of the modern world. Regardless of the type of information (image, video, audio, etc.) and the transmission medium (a coaxial cable, a band of radio frequency, a beam of light, etc.), many scenarios can be modelled by a point-to-point communication system as shown in Fig. 1.

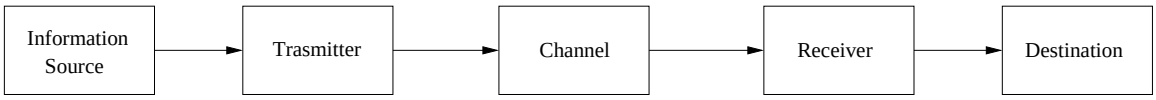

Fig. 1. A point-to-point communication system.

A point-to-point communication system contains five parts: an information source, a transmitter, a channel, a receiver, and a destination. Given a source signal, the transmitter produces a signal suitable for transmission over the channel. The channel, which is generally imperfect, may introduce noise to this signal. The object of the receiver is to reconstruct the original source with the highest possible fidelity.

Provided that the same fidelity of the reconstructed source is maintained, an efficient transmitter-receiver pair should minimize the use of resources such as the power of the transmitted signal and the number of channel uses. On one hand, the transmitter should remove any redundancy in the information source to reduce unnecessary channel use. For example, consecutive frames in a slowly varying video

---

This dissertation follows the style of *IEEE Trans. Inform. Theory*.

sequence are almost the same. Therefore, given the first frame, most pixels in the next frame can be well predicted and hence are redundant. The process of removing redundancy essentially "compresses" the source and is commonly known as *source coding*. On the other hand, the transmitter can introduce useful redundancy with which the receiver can detect and potentially correct transmission errors caused by the channel noise. As a result, compared to an uncoded system that maintains the same fidelity of reconstructed signal, less power will be needed. The process of introducing redundancy is commonly known as *channel coding*.

Our first impulse might suggest that a scheme constructed by designing source coding and channel coding independently cannot be optimum. However, from the Shannon's separation theorem [85, 94], there is no loss in theory in restricting ourselves to a separate design. Therefore, we can design an optimum scheme by combining the best source code for the given the information source and the best channel code for the given channel. This makes source coding and channel coding each an interesting area of study of its own.

B. Coding with Side Information

In many scenarios, besides their regular inputs, the transmitter and/or the receiver are given some extra information regarding the source and the channel. For example, this "side information" can be the nature and the format of the source and the mean and the variance of the channel noise. To incorporate this side information in a communication system, it is thus necessary to study source coding and channel coding with side information.

Since side information can be given to the encoder and/or decoder, this results in four different cases. However, several of these cases are trivial in the sense that

conventional source and channel coding techniques can be employed directly. For example, when side information is given to both the encoder and decoder, we can easily include this side information in the scheme design by using optimized coders for the different outcomes of the side information. Yet another example, consider source coding when side information is given to the encoder alone; it is shown in [11] that the side information is useless and thus can be ignored.[2] The two most interesting cases are source coding with side information at the decoder, a.k.a. Wyner-Ziv coding (WZC) [104], and channel coding with side information at the encoder, a.k.a. Gel'fand-Pinsker coding (GPC) [47]. Therefore, we will be focus on these two cases in this thesis.

C. Applications

Besides the connections with a point-to-point communication system, WZC and GPC are closely related to multiterminal communication systems with more than one transmitter and/or one receiver. More precisely, WZC and GPC can be used as a building block for distributed source coding [78, 107] and broadcast channel coding [31, 33], respectively.

1. Distributed Source Coding and Wyner-Ziv Coding

Consider numerous heat sensors spreading over a region, measuring temperature, and sending it back to a base station. In order to save the production cost of these sensors and simplify the scheme design, we assume these sensors transmit measurements directly to the base station without the help of other sensors as relay. Hence, the transmitter in each sensor can only know its local measurement. However, in most

[2] This is intuitive because all possible side information related to the source can be generated from the source itself, and the latter is always given to the encoder.

cases, the measurements of all these sensors are correlated; so the question is: can we incorporate this correlation effectively to compress these measurements even though joint encoding is not permitted?

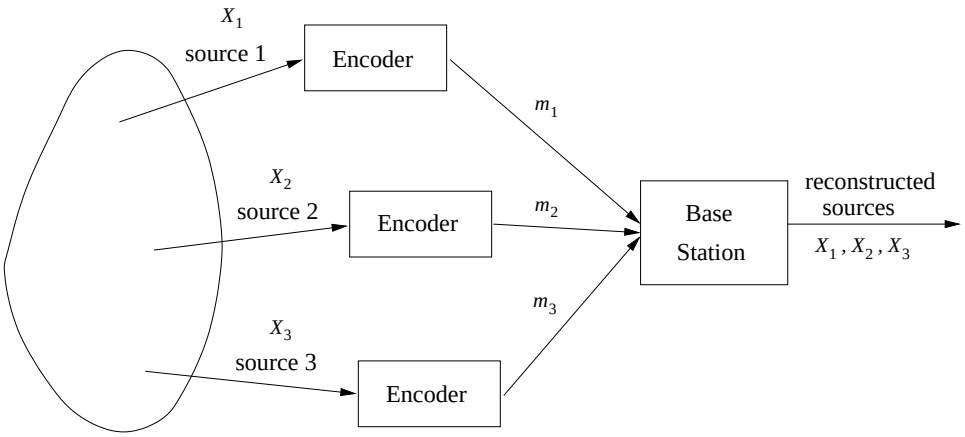

Fig. 2. Distributed source coding with three sources.

The above scenario is a typical example of distributed source coding in which several correlated sources are encoded separately but decoded jointly as shown in Fig. 2. A solution of this interesting problem can be implemented using WZC. As shown in Fig. 3, the first source X_1 will be coded using conventional source coding. At the base station, X_1 will be the first to be decoded and used as side information for the subsequent decoding of all other sources. Knowing the reconstructed $\hat{X}_1$ at the base station, the second source X_2 is coded using WZC. And just as $\hat{X}_1$, the reconstructed $\hat{X}_2$ is also treated as side information for the subsequent decoding stages. Similar decoding procedure with all the previous decoded sources as side information continues until all sources are reconstructed.

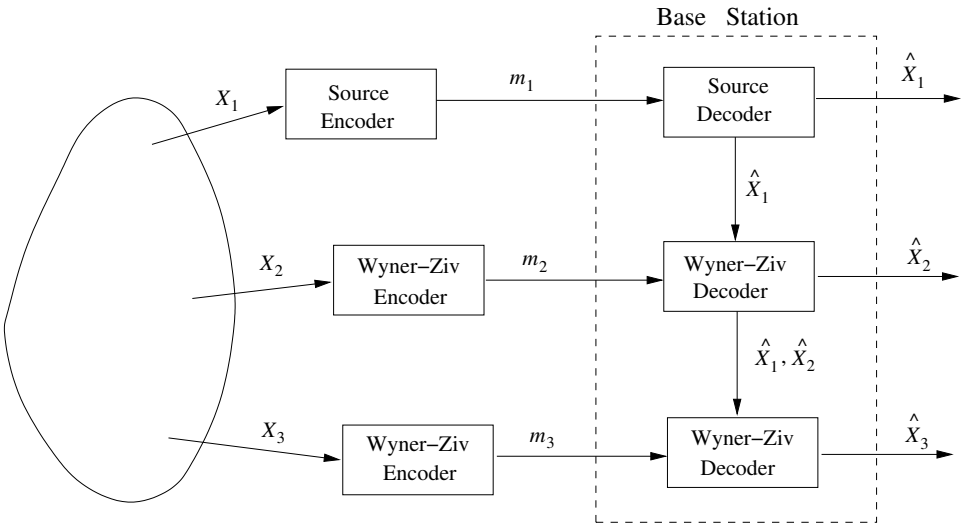

Fig. 3. Distributed source coding implemented by WZC.

2. Broadcast Channel Coding and Gel'fand-Pinsker Coding

As shown in Fig. 4, a broadcast channel setup includes one sender and several receivers. The object is to broadcast messages from the sender to all receivers. A typical example is TV or radio broadcast, where the same "messages" are broadcasted to all receivers. In general, the message to each receiver can be different as will be discussed here.

Similar to the relation between DSC and WZC, broadcast channel coding can be implemented using GPC as building blocks. Whereas decoding is performed layer by layer in DSC, encoding will be implemented layer-wise here instead. As shown in Fig. 5, encoding is split into two steps. Temporary outputs X_1, X_2, and X_3 are generated in the first step and they are combined to form the actual encoding output X in the second step. GPC is incorporated into the broadcast channel setup as follows. The first message m_1 is transmitted through the hypothetical channel between X_1 and Y_1 using conventional channel coding. Upon making the decision of X_1, m_2 is

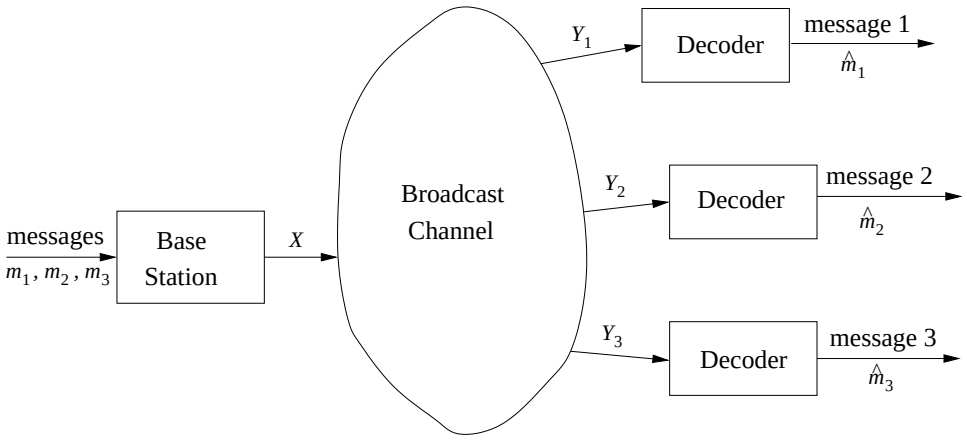

Fig. 4. Broadcast channel coding with three receivers.

sent through the hypothetical channel between X_2 and Y_2 using GPC with X_1 as side information. In general, the message m_i is transmitted through the hypothetical channel between X_i and Y_i using GPC with $X_1, X_2, ..., X_{i-1}$ as side information.

D. Brief History of WZC and GPC

Lossless source coding with side information at the decoder was introduced by Wyner and Ziv in [103, 101]; the achievable region of this problem was addressed by Ahlswede and Körner in [5] and by Wyner in [99, 100]. This problem can be viewed as a special case of lossless distributed source coding, whose theoretical limit for two input sources was found by Slepian and Wolf in [87] and by Gács-Körner in [42]. Due to the renowned work of [87], lossless source coding with side information at the decoder is also commonly known as asymmetric Slepian-Wolf Coding (SWC) or simply SWC. The lossy source coding problem with side information at the decoder, i.e., the WZC problem described in this thesis, was both introduced and solved theoretically by Wyner and Ziv in [104].

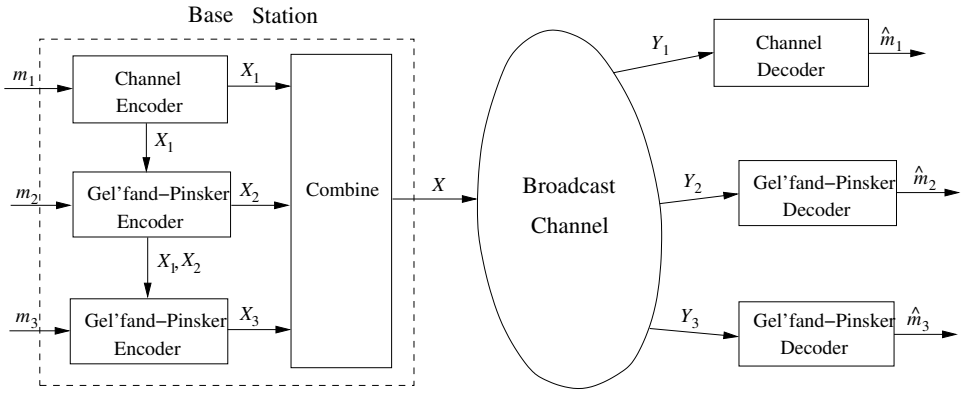

Fig. 5. Broadcast channel coding implemented by GPC.

Channel coding with side information at the encoder was first addressed by Shannon in [86]. However, he considered a causal case where future channel states are not available to the transmitter. The noncausal case was first considered by Kusnetsov and Tsybakov in [55], where channels with random defects and errors were examined. The GPC problem studied in this thesis was a generalized version of this problem and its capacity was found by Gel'fand and Pinsker in [47].

Wyner was the first who hinted a practical solution for SWC [105], which was based on channel coding. However, his approach had been widely forgotten and it was until 1999 when Pradhan and Ramchandran rediscovered it and presented the first implementation [75]. Since then, it is commonly accepted that SWC is a channel coding problem in nature. To achieve the theoretical limit, schemes based on capacity approaching channel codes such as turbo code [12] and low-density parity-check (LDPC) code [43, 59] were studied by numerous researchers in [45, 9, 4, 58] and [84, 88, 26, 57], respectively.

For WZC, Zamir and Shamai proposed a nested coding scheme [112] that potentially can reach the WZC limit. However, this is possible only when high dimensional

lattice codes, which are very difficult to implement in practice, are used. In [75], Pradhan and Ramchandran implemented a practical scheme based on trellis code, which were then extended by Wang and Orchard in [96]. In [79], Rebollo-Monedero *et al.* treated WZC as a quantization problem and attempted to solve it by optimal quantizer design. The best result before our work was by Chou *et al.* described in [21], which was based on a combination of turbo code and trellis coded quantization [63].

The nested coding idea used in WZC was proposed for GPC by Zamir *et al.* in [113]. Based on this idea, numerous attempts [72, 73, 41] have been made to use advance channel codes to implement the nested code. However, this involves the use of channel codes for source coding. This is a challenging problem that remains open. Digital watermarking [91] has been an active research area since early 90's. However, most of the earlier work was ad hoc in nature [91, 49, 71]; a noteworthy counterexample is spread spectrum watermarking introduced by Cox *et al.* in [34], which borrows idea from spread spectrum communications [74]. In the late 90's, it was recognized in [35, 19, 65] that digital watermarking can be treated as a special case of GPC. This opens a whole new perspective in approaching the digital watermarking problem.

E. Organization of the Thesis

Although the theoretical limits of coding problems with side information are well-known, their implementations are not. Therefore, we focus on their practical designs in this thesis, which is organized as follows. In Chapter II, we review the theoretical background of WZC and GPC, present a theoretical result regarding the successive refinability of WZC, and describe a computational algorithm in finding theoretical

limits for coding problems with side information in general. Chapter III is focused on the practical design of SWC and WZC. We depict the SWC design based on low-density parity-check (LDPC) code and describe a general paradigm dubbed *Slepian-Wolf coded quantization (SWCQ)* for WZC. Chapter IV describes the practical design of GPC in the sense of digital watermarking. Application examples are provided.

F. Contributions of the Thesis

The main contributions of the thesis are the following:

- An efficient algorithm to compute the theoretical limit of the coding problem with side information for any discrete source and channel.
- Stating and proving that a general class of sources is successively refinable in the WZC setting.
- An efficient design of SWC based on LDPC codes.
- A sufficient condition when the SWC performance is equivalent to the corresponding LDPC code performance of conventional channel coding.
- A WZC paradigm that outperforms any previous scheme reported in the literature [108].
- Schematic connection between our WZC paradigm and entropy-coded quantization for classic source coding.
- An improved spread spectrum watermarking technique for general digital watermarking problems.
- A novel AAC audio watermarking scheme.

- Error concealment of AAC audio using digital watermarking.

G. Notations and Conventions

Empty set is represented by ø. We use the shorthanded notation x_k^n for the sequence $x_k, x_{k+1}, ..., x_n$. When $k > n$, x_k^n will be understood as a null sequence. Script letters are used for the alphabets of random variables. A channel with the input X and the output Y will be represented by $X \rightarrow Y$. We always assume a binary input channel $X \rightarrow Y$ with input alphabet $\{-1, 1\}$ unless stated otherwise. Without sacrificing clarity, we slightly abuse our notations in which an operation on a vector is interpreted as that on the individual components. For example, $f([x_1, x_2, x_3]) \triangleq [f(x_1), f(x_2), f(x_3)]$.

CHAPTER II

THEORY OF WYNER-ZIV CODING AND GEL'FAND-PINSKER CODING

In this chapter, we will focus on the theoretical aspect of the two coding problems, Wyner-Ziv Coding (WZC) [104] and Gel'fand-Pinsker Coding (GPC) [47]. In the first section, we will present formal definitions for the problems and review their theoretical limits. We will explain the duality of the two problems in Section IIB. In Section IIC, we present a generalization of successive refinement from classic source coding to WZC and our contribution on this area. To end this chapter, we will derive an iterative algorithm in computing the theoretical limits of WZC and GPC problems.

A. Problem Setups and Theoretical Limits

1. Wyner-Ziv Coding

Given two identically and independently distributed (i.i.d.) and correlated sources X and S with joint distribution $p_{S,X}(s,x)$, the WZC problem is the lossy compression problem of X with S as side information provided only to the decoder. Define a distortion mapping $\mathfrak{d}(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the alphabet of X.[1] For a predefined distortion D, the minimum rate required to have the reconstructed $\hat{X}$ satisfy $E\{\mathfrak{d}(X,\hat{X})\} \leq D$ is [104]

$$R_{WZ}(D) = \min_{\substack{p(u|x)p(\hat{x}|s,u) \\ :E[\mathfrak{d}(X,\hat{X})]\leq D}} I(U;X) - I(U;S), \tag{2.1}$$

where U is an auxiliary random variable. Note that in general (2.1) itself is a optimization problem. In Section IID, an efficient way to compute the rate-distortion function

[1] For simplicity, we assume the reconstructed $\hat{X}$ shares the same alphabet $\mathcal{X}$ with X.

will be presented. Before that, we describe here two cases when the rate-distortion function can be solved relatively easily.

a. Binary Symmetric Case

X and S are binary symmetric sources, the correlation between them is modelled as a binary symmetric channel with crossover probability p_Z and the distortion measure is the Hamming distance. We can write $X = S \bigoplus Z$, where Z is a Bernouli(p_Z) source. Then the rate-distortion function $R_Z(D)$ for Z serves as the performance limit $R_{X|S}(D)$ of lossy coding of X given S at both the encoder and the decoder. From [30] we have

$$R_{X|S}(D) = R_Z(D) = \begin{cases} H(p_Z) - H(D), & 0 \le D \le \min\{p_Z, 1-p_Z\}, \\ 0, & D > \min\{p_Z, 1-p_Z\}. \end{cases} \tag{2.2}$$

On the other hand, the Wyner-Ziv rate-distortion function in this case is [104]

$$R_{WZ}(D) = l.c.e\{H(p_Z * D) - H(D), (p_Z, 0)\}, 0 \le D \le p_Z, \tag{2.3}$$

the lower convex envelop of $H(p_Z * D) - H(D)$ and the point $(D = p_Z, R = 0)$, where $p_Z * D = (1-p_Z)D + (1-D)p_Z$.

For $p_Z \le 0.5$, $R_{WZ}(D) \ge R_{X|S}(D)$ with equality only at two trivial distortion-rate points: $(p_Z, 0)$ and $(0, H(p_Z))$. See Fig. 6 for $p_Z = 0.27$. Thus Wyner-Ziv coding suffers rate loss in this binary symmetric case for not having the side information S at the decoder. When $D = 0$, the Wyner-Ziv problem degenerates to the Slepian-Wolf problem with $R_{WZ}(0) = R_{X|S}(0) = H(X|S) = H(p_Z)$.

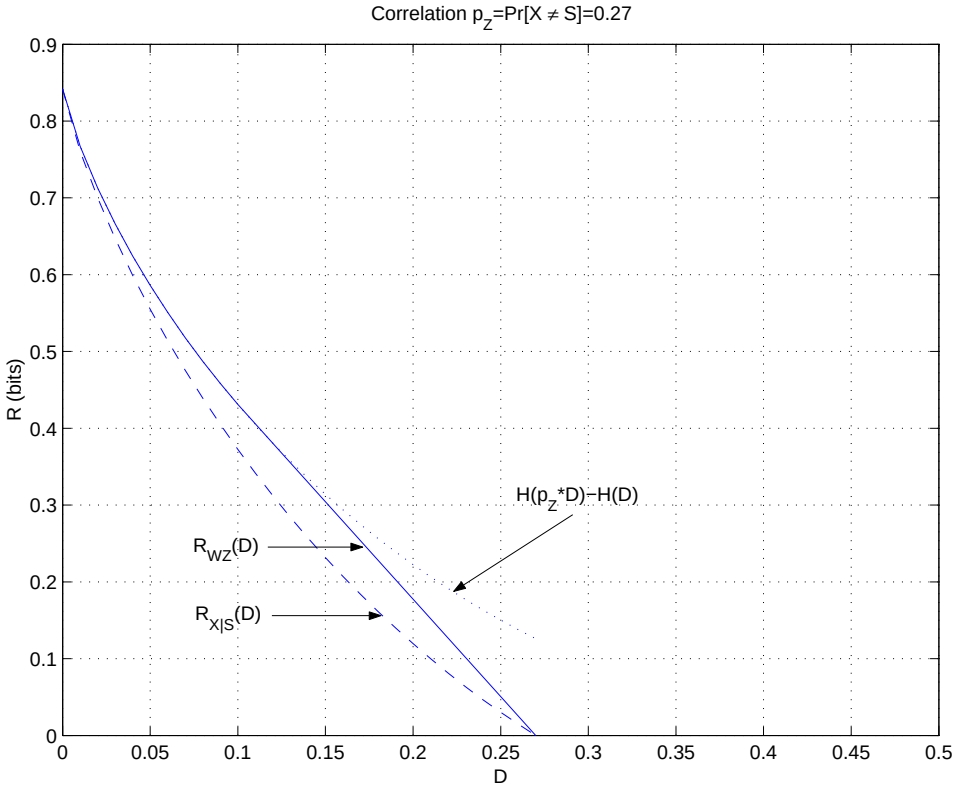

Fig. 6. $R_{WZ}(D)$ and $R_{X|S}(D)$ for the binary symmetric case with $p_Z = 0.27$.

b. Quadratic Gaussian Case

X and S are zero mean and stationary Gaussian memoryless sources and the distortion metric is MSE. Let the covariance matrix of X and S be $\Lambda = \begin{bmatrix} \sigma_X^2 & \varrho\sigma_X\sigma_S \\ \varrho\sigma_X\sigma_S & \sigma_S^2 \end{bmatrix}$ with $|\varrho| < 1$, then [102]

$$R_{WZ}(D) = R_{X|S}(D) = \frac{1}{2}\log^+\left[\frac{\sigma_X^2(1-\varrho^2)}{D}\right], \tag{2.4}$$

where $\log^+ x = \max\{\log x, 0\}$. Surprisingly, there is no rate loss with Wyner-Ziv coding in this quadratic Gaussian case![2] If S can be written as $S = X + Z$, with

[2]This result will be shown in Section IIC as a byproduct of the proof of successive refinability of quadratic Gaussian sources.

independent $X \sim N(0, \sigma_X^2)$ and $Z \sim N(0, \sigma_Z^2)$, then

$$R_{WZ}(D) = R_{X|S}(D) = \frac{1}{2}\log^+\left[\frac{\sigma_Z^2}{(1+\sigma_Z^2/\sigma_X^2)D}\right]. \tag{2.5}$$

On the other hand, if $X = S + Z$, with independent $S \sim N(0, \sigma_S^2)$ and $Z \sim N(0, \sigma_Z^2)$, then

$$R_{WZ}(D) = R_{X|S}(D) = \frac{1}{2}\log^+\left(\frac{\sigma_Z^2}{D}\right). \tag{2.6}$$

2. Gel'fand-Pinsker Coding

Consider a memoryless channel with channel state information S. More precisely, the output of the channel Y is probabilistic with distribution $p(y|s,x)$ when the channel input and channel state information are x and s, respectively. The GPC problem is the channel coding problem when the channel state information S is given only to the encoder. The maximum rate to have lossless transmission, i.e., the capacity of the channel, is [47]

$$C_{GP} = \max_{p(u|s)p(x|u,s)} I(U;Y) - I(U;S), \tag{2.7}$$

where U is an auxiliary random variables. For some cases, we may want to constraint the "power" of the channel input. Define a power mapping $\mathfrak{p}(\cdot,\cdot) : \mathcal{S} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{S}$ and $\mathcal{X}$ are the alphabets of S and X, respectively. For a predefined power constraint P, the capacity of the channel is

$$C_{GP}(P) = \max_{\substack{p(u|s)p(x|u,s) \\ :E[\mathfrak{p}(S,X)]\leq P}} I(U;Y) - I(U;S), \tag{2.8}$$

a. Binary Symmetric Case

Consider the channel described by $Y = S \bigoplus X \bigoplus Z$, where X and Y are the input and output of the channel and S, X and Z are independent. Let Z and S be Bernouli(p_Z) and Bernouli(p_S) sources, respectively. Define the power measure as the Hamming

weight of X. When S is given to both the encoder and decoder as side information, the capacity-power function is [30]

$$C_{Y|S}(P) = H(P * p_Z) - H(p_Z), \tag{2.9}$$

where $P * p_Z = (1 - p_Z)P + (1 - P)p_Z$. On the other hand, the Gel'fand-Pinsker capacity-power function in this case is [77]

$$C_{GP}(P) = u.c.e.\{H(P) - H(p_Z), (0, 0)\}, P \leq 0.5, \tag{2.10}$$

the upper concave envelop of $H(P) - H(p_Z)$ and the point $(P = 0, C = 0)$. For $P \leq 0.5$, $C_{GP}(P) \leq C_{Y|S}(P)$ with equality only at two trival capacity-power points: $(0, 0)$ and $(0.5, 1 - H(p_Z))$ (see Fig. 7 for $p_Z = 0.1$). Thus Gel'fand-Pinsker coding suffers capacity loss in this binary symmetric case for not having the side information S at the decoder.

b. Quadratic Gaussian Case: Dirty Paper Coding

Consider a similar additive channel as the previous case, i.e., $Y = X + S + Z$, where X and Y are the input and output of the channel and S, X and Z are independent. Assume, however, S and Z are Gaussian with variances σ_S^2 and σ_Z^2, respectively. Consider S as the side information and use X^2 as the power measure. We can think of S as an interference known to the encoder but not the decoder. An interesting analogy is writing a message over dirty paper; as a result the writer can tell for sure where the dirt is but the reader cannot because the dirt and the written message may not be distinguishable. Hence, this special case of GPC is also commonly known as dirty paper coding. From [29], we have

$$C_{GP}(P) = C_{Y|S}(P) = \frac{1}{2} \log^+ \frac{P}{\sigma_Z^2}, \tag{2.11}$$

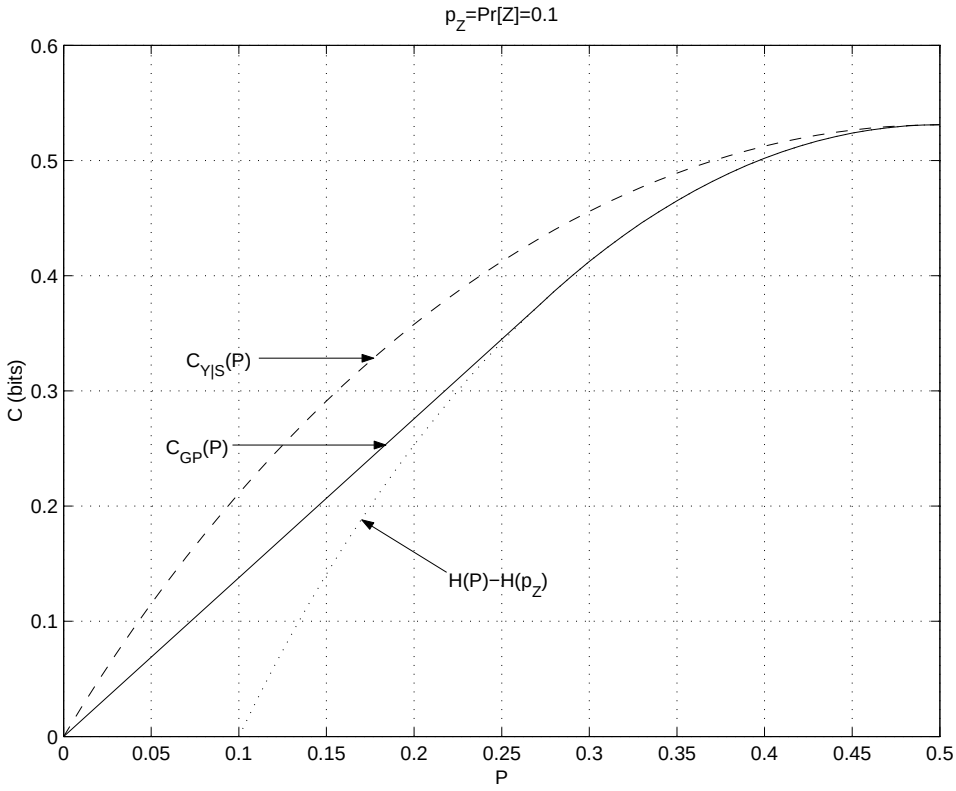

Fig. 7. $C_{GP}(P)$ and $C_{Y|S}(P)$ for the binary symmetric case with $p_Z = 0.1$.

where $\log^+ x = \max\{logx, 0\}$. Surprisingly, there is no capacity loss of Gel'fand-Pinsker coding in this quadratic Gaussian case!

B. Duality of Wyner-Ziv Coding and Gel'fand-Pinsker Coding

The duality of WZC and GPC has been addressed by several research groups [32, 77, 10] and can be visualized if we concatenate the two setups as in Fig. 8. As shown clearly in Fig. 8, the Gel'fand-Pinsker encoder essentially plays the same role as the Wyner-Ziv decoder and so as the reconstructed $\hat{X}$ in WZC and the channel input X in GPC. Similarly, we can reverse the order of WZC and GPC in Fig. 8 to enable us to visualize the duality between the Wyner-Ziv encoder and the Gel'fand-Pinsker decoder. Table I summarizes the dual components of WZC and GPC.

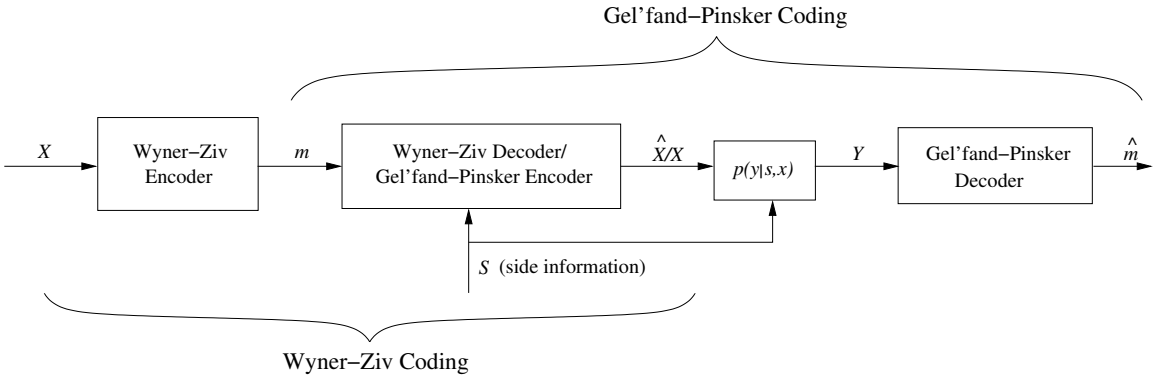

Fig. 8. Illustrating duality of WZC and GPC.

Table I. Dual components of WZC and GPC.

WZC	GPC
X	Y
$\hat{X}$	X
S	S
Encoder	Decoder
Decoder	Encoder

In [77], Pradhan *et al.* defines a stricter sense of duality in which the definition requires the optimum setups (i.e., those achieve the rate-distortion function and the capacity-power function) share the same joint distribution for both problem. However, we will not go further detail into this kind of duality. Instead, we attempt to further clarify the concept of duality via a specific example.

1. Duality Example: Quadratic Gaussian Case

We now depict in more detail the duality of the quadratic Gaussian cases in WZC and GPC. However, to better illustrate the duality, we will reformat dirty paper coding

(Gaussian GPC) into digital watermarking [49].

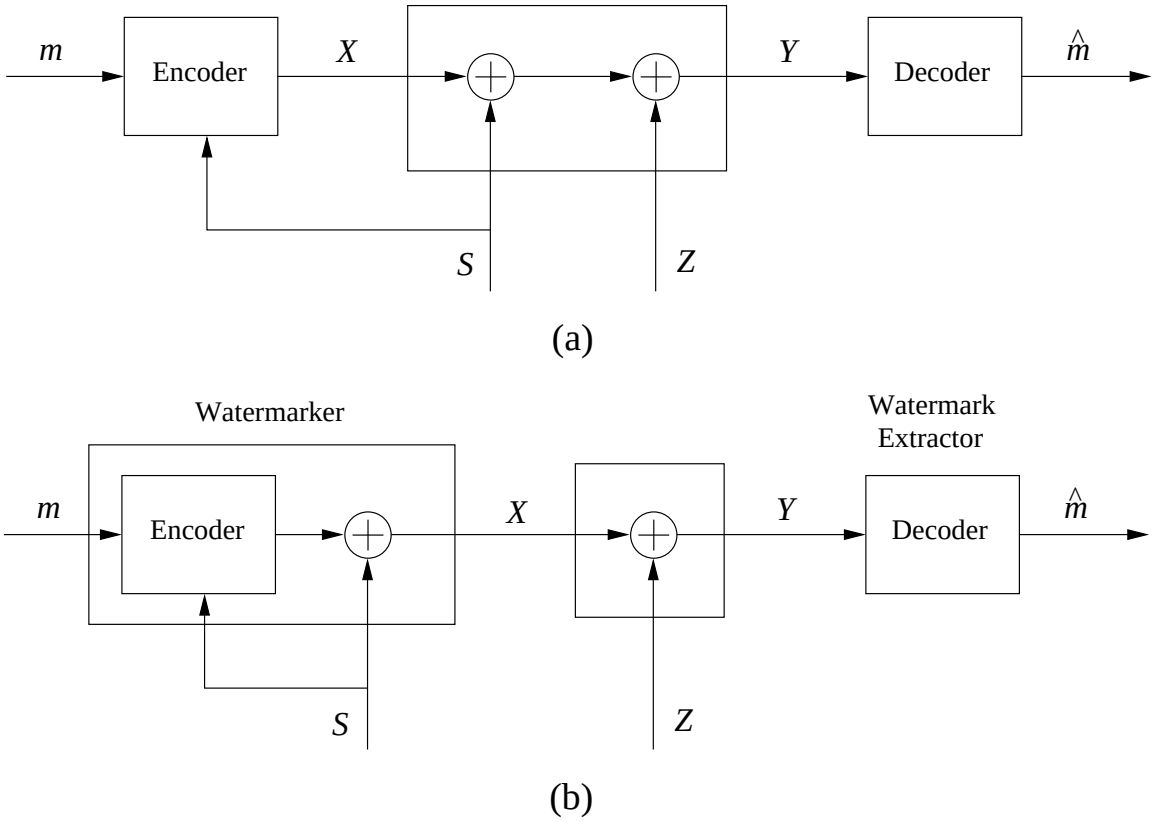

Fig. 9. Gaussian Gel'fand-Pinsker coding in different setups: (a) dirty paper coding and (b) digital watermarking.

As shown in Fig. 9, the two setups are essentially the same. The only difference is that the addition of S is done inside the encoder instead of the channel for the digital watermarking case. As a result, there is a renaming of X, the output of the encoder; X in the digital watermarking setup is now equivalent to $X + S$ in the dirty paper coding setup. Moreover, rather than considered as an interference, S is now interpreted as a host signal into which an watermark is embedded. In a nutshell, the object of digital watermarking is to maximize the robustness of the watermark against noise for the fixed distortion of the watermarked signal. More background

on the digital watermarking problem will be described later in Chapter IV. In the following, we will describe a coding technique, namely nested coding, through which the duality of Gaussian WZC and digital watermarking is exemplified.

Consider a code $\mathcal{C}$ and its subcode $\mathcal{C}_i \subset \mathcal{C}$, $i = 0, 1, 2, ..., N-1$. We call this code collection a nested code if $\mathcal{C}_i$, $i = 0, 1, 2, ..., N-1$, partition $\mathcal{C}$. That is

$$\mathcal{C} = \bigcup_{i=0}^{N-1} \mathcal{C}_i \tag{2.12}$$

$$\mathcal{C}_i \cap \mathcal{C}_j = \emptyset, \forall i, j, i \neq j \tag{2.13}$$

This nested coding setup is used for WZC as follows. Given a source X, the encoder searches for the codeword c that is closest from X. This essentially "quantized" x to c as in conventional source coding. However, instead of directly transmitting c to the decoder, only the index of the subcode containing c will be sent. More precisely, the encoder transmits m to the decoder if $c \in \mathcal{C}_m$. The rationale is as follows. Assuming that the correlation between X and S are sufficiently large, the decoder can correctly identify c out of $\mathcal{C}_m$ with high probability by reconstructing it simply as the closest codeword from S.

For GPC, or digital watermarking in this case, the watermarker attempts to embed a message m by modifying S and the amplitude of this modification should be minimized to preserve the quality of the watermarked signal. To use the nested code for GPC, for a message m, the encoder sends the codeword c in $\mathcal{C}_m$ that is closest to S, whereas the decoder simply recover the message as the index of subset that contains a codeword closest to the received watermarked signal Y. The encoding and decoding procedures of both WZC and GPC are summarized in Table II, which clearly manifests the duality of the two coding problems as the encoder of one is exactly the same as the decoder of the other and vice versa.

Table II. Encoding and decoding procedures of Gaussian WZC and Gaussian GPC using nested code.

	WZC	GPC
Encoding	Input: x Output: m if $\begin{cases} c = \arg\min_{c\in\mathcal{C}}(x-c)^2 \\ c \in \mathcal{C}_m \end{cases}$	Input: m, s Output: $c = \arg\min_{c\in\mathcal{C}_m}(s-c)^2$
Decoding	Input: m, s Output: $c = \arg\min_{c\in\mathcal{C}_m}(s-c)^2$	Input: y Output: m if $\begin{cases} c = \arg\min_{c\in\mathcal{C}}(y-c)^2 \\ c \in \mathcal{C}_m \end{cases}$

C. Successive Refinement of Wyner-Ziv Coding

In this section, we focus on successive refinement of the Wyner-Ziv problem described in [89]. Similar to the problem in classic source coding [40], a successive refinement coding scheme for the Wyner-Ziv problem consists of multi-stage encoders and decoders where each decoder uses all the information generated from previous encoding stages and the side information, which could be different from stage to stage. We call such a scheme successively refinable if the rate-distortion pair associated with any stage falls on the same Wyner-Ziv rate-distortion curve given the corresponding side information. It was shown in [89] that if the side information for all stages are identical, the jointly Gaussian source with squared error distortion measure is successively refinable. We extend successive refinability from jointly Gaussian source to the more general types of sources described by Pradhan *et al.* in [77]. In other words, we

show that a source is successively refinable in the Wyner-Ziv setting as long as the difference between the source and the side information is Gaussian and independent of the side information. As a by-product, we give an alternative proof of a result in [77] regarding no rate loss in WZC.

In the following, we review the definition of successive refinement and successive refinability for the Wyner-Ziv problem. Our theoretical result is presented afterward.

1. Theoretical Background

Definition 1 (Successive refinement code [89]): An $(n, \mathfrak{M}_1, \mathfrak{M}_2, D_1, D_2)$ successive refinement (SR) code for the source X with side information S_1 and S_2 consists of a first-stage encoder-decoder pair $(\mathfrak{f}_1, \mathfrak{g}_1)$:

$$\mathfrak{f}_1 : \mathcal{X}^n \to \{1, 2, ..., \mathfrak{M}_1\}$$
$$\mathfrak{g}_1 : \{1, 2, ..., \mathfrak{M}_1\} \times \mathcal{S}_1^n \to \hat{\mathcal{X}}^n$$

and a second-stage (or refinement) encoder-decoder pair $(\mathfrak{f}_2, \mathfrak{g}_2)$:

$$\mathfrak{f}_2 : \mathcal{X}^n \to \{1, 2, ..., \mathfrak{M}_2\}$$
$$\mathfrak{g}_2 : \{1, 2, ..., \mathfrak{M}_1\} \times \{1, 2, ..., \mathfrak{M}_2\} \times \mathcal{S}_2^n \to \hat{\mathcal{X}}^n$$

such that $E[\mathfrak{d}(X^n, \mathfrak{g}_1(\mathfrak{f}_1(X^n), S_1^n))] \leq D_1$ and $E[\mathfrak{d}(X^n, \mathfrak{g}_2(\mathfrak{f}_1(X^n), \mathfrak{f}_2(X^n), S_2^n))] \leq D_2$.

Definition 2 (Successive refinability [89]): A source X is said to be successively refinable from D_1 to D_2 $(D_1 > D_2)$ with side information S_1 and S_2 if for any $\delta > 0$ and $\epsilon > 0$, there exists an $(n, exp[n(R_{WZ,S_1}(D_1) + \delta)], exp[n(R_{WZ,S_2}(D_2) - R_{WZ,S_1}(D_1) + \delta)], D_1 + \epsilon, D_2 + \epsilon)$ SR code for some sufficiently large n, where $R_{WZ,S_1}(D)$

and $R_{WZ,S_2}(D)$ are the Wyner-Ziv rate-distortion functions with side information S_1 and S_2, respectively.

Successive refinement can be naturally extended to any finite number of stages. We skip the formal definition of a multistage successive code, as it is a straightforward extension of Definition 1. One degenerate, but important scenario, is when the side information at all the decoding stages are the same. Under this situation, we repeat the conditions given in [89] for successive refinability as follows:

A source X with identical side information S is K-stage successively refinable with distortion levels $\mathbf{D} = (D_1, D_2, ..., D_K)$, if and only if there exist random variables, U_1, $U_2, ..., U_K$, and K deterministic functions $f_k : \mathcal{U}_k \times \mathcal{S} \to \hat{\mathcal{X}}$, $1 \leq k \leq K$, such that the following conditions hold:

1. $R_{X|S}(D_k) = I(X; U_k|S)$ and $E[\mathfrak{d}(X, f_k(U_k, S))] \leq D_k, k = 1, 2, ..., K$
2. $(U_1, U_2, ..., U_K) \leftrightarrow X \leftrightarrow S$
3. $(U_1, U_2, ..., U_{k-1}) \leftrightarrow (U_k, S) \leftrightarrow X, k = 2, 3, ..., K.$

2. Main Result

Proposition 1: Given a source X and common side information S for all refinement stages, the Wyner-Ziv problem is successively refinable if $X = S + Z$, where $Z \sim N(0, \sigma_Z^2)$ is the Gaussian noise, independent of S.

Proof. Construct $U = X + T$ as the auxiliary random variable, where $T \sim N(0, \sigma_T^2)$ is independent of X as shown in Fig. 10, then

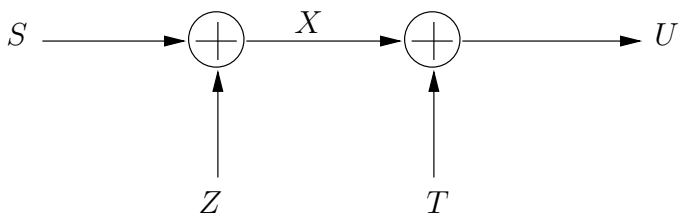

Fig. 10. Optimum setup of the Wyner-Ziv problem for $X = S + Z$.

$$\begin{aligned} R &= I(X;U) - I(S;U) \\ &= H(U) - H(U|X) - H(U) + H(U|S) \\ &= -H(X+T|X) + H((S+Z)+T|S) \\ &= -H(T) + H(Z+T) \\ &= \frac{1}{2}\log^+ \frac{\sigma_Z^2 + \sigma_T^2}{\sigma_T^2}, \end{aligned} \tag{2.14}$$

where $\log^+ x = \max\{\log x, 0\}$. Let $\hat{X} = aS + bU$ be the MMSE linear estimate of X given S and U, then from $E[(\hat{X} - X)S] = E[(\hat{X} - X)U] = 0$, we have

$$a = \frac{\sigma_T^2}{\sigma_Z^2 + \sigma_T^2}, b = \frac{\sigma_Z^2}{\sigma_Z^2 + \sigma_T^2},$$

and

$$D = E[(X - \hat{X})^2] = \frac{\sigma_T^2 \sigma_Z^2}{\sigma_Z^2 + \sigma_T^2}. \tag{2.15}$$

Substitute (2.15) into (2.14), we get

$$R = \frac{1}{2}\log^+ \frac{\sigma_Z^2}{D} = \frac{1}{2}\log^+ \frac{\sigma_{X|S}^2}{D} = R_{X|S}(D). \tag{2.16}$$

Since (2.16) coincides with the rate-distortion function when side information is also given to the encoder, the setup in Fig. 10 must be optimal. Hence

$$R^*(D) = \frac{1}{2}\log^+ \frac{\sigma_Z^2}{D}.$$

Now, we attach a second stage to further decompose U into T' and U' as in Fig. 11, where $T' \sim N(0, \sigma_{T'}^2)$ is independent of U. We can consider $T + T'$ as a Gaussian random variable and this reduces our setup to the one in the previous case in Fig. 10. Thus we also achieve the Wyner-Ziv bound with the auxiliary random variable U'.

To refine from distortion D_1 (with the first stage auxiliary random variable U') to distortion D_2 (with the second stage auxiliary random variable U), we can set

$$\sigma_T^2 = \frac{\sigma_Z^2}{\frac{\sigma_Z^2}{D_2} - 1} \tag{2.17}$$

and

$$\sigma_T^2 + \sigma_{T'}^2 = \frac{\sigma_Z^2}{\frac{\sigma_Z^2}{D_1} - 1}.$$

This gives us

$$\sigma_{T'}^2 = \frac{\sigma_Z^4 (D_1 - D_2)}{(\sigma_Z^2 - D_1)(\sigma_Z^2 - D_2)}. \tag{2.18}$$

This is possible since both σ_T^2 and $\sigma_{T'}^2$ in (2.17) and (2.18) are positive. Hence, condition (1) in Section 1 is satisfied. The other two Markov conditions (2) and (3) can be readily verified from the setup. We can further decompose U' and apply similar arguments when we have more than two stages. □

Remark 1: We can conclude from (2.16) that the Wyner-Ziv problem has no rate loss in this general case with $X = S + Z$. This constitutes a direct proof of a result that was first obtained in [77, p.1194] by invoking the duality between the Wyner-Ziv problem and the Costa problem [29].

Remark 2: We can show with slight modification in the above proof that the Wyner-Ziv problem is also successive refinable with $X = g(S) + Z$ when $g(\cdot)$ is a one-to-one mapping and Z is Gaussian and independent of S. This is because the decoder

can treat $g(S)$ as the actual side information. There is no loss in doing so, i.e., $I(U;X) - I(U;S) = I(U;X) - I(U;g(S))$, because $g(\cdot)$ is one-to-one and thus the Markov chain $S \leftrightarrow g(S) \leftrightarrow U$ holds.

Remark 3: We do not claim that *all* sources that have no rate loss in WZC are successively refinable. This is because successive refinability and no rate loss in WZC are two different concepts. Note that although Equitz and Cover [40] demonstrated a source that is not successively refinable in the classic setting (without side information), this source was shown by Steinberg and Merhav [89] to be successively refinable in the presence of identical side information. We conjecture that there are non-successively refinable sources (with or without rate loss) in the Wyner-Ziv setting, but we are not able to come up with an example. On the other hand, we know that the doubly symmetric binary source (with Hamming distance measure) has rate loss but is successively refinable with WZC.

Equipped with Proposition 1 and Remark 2, we are able to give a short proof that the Wyner-Ziv problem for any jointly Gaussian source is successively refinable.

Corollary 1: The Wyner-Ziv problem is successively refinable if the same side information S, which is jointly Gaussian with the source X, is used in all stages.

Proof. We can model any joint Gaussian pair (S, X) as $X = \alpha S + Z$, where $\alpha = \frac{E[XS]-E[X]E[S]}{E[S^2]-E^2[S]}$ and Z is Gaussian random variable with the mean $E[Z] = \frac{E[S^2]E[X]-E[S]E[SX]}{E[S^2]-E^2[S]}$ and the variance $\sigma_Z^2 = E[(X - \alpha S)^2] - E^2[Z]$ and independent of S. Then the proof follows from Remark 2 and Proposition 1 immediately. □

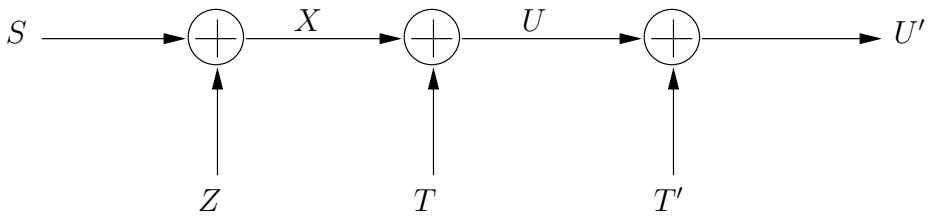

Fig. 11. Successive refinement with side information for $X = S + Z$.

D. Computing Theoretical Limits Using Iterative Algorithm

In the beginning of this chapter, we explain that the capacity and rate-distortion function of GPC and WZC are expressed as optimization problems. Unless for some restrictive setups, like the Gaussian and binary symmetric cases, these theoretical limits cannot be found analytically. Hence, the goal of this section is to derive an algorithm in finding these theoretical limits for general cases. However, instead of directly tackling GPC and WZC, we look into the even more general setups when two different pieces of side information, S_1 and S_2, are given to encoder and decoder, respectively. In [32], the authors showed that the capacity and the rate-distortion function for the corresponding problems are given as

$$C = \max_{q(u|s_1)q'(x|u,s_1)} I(U;Y,S_2) - I(U;S_1) \tag{2.19}$$

and

$$R(D) = \min_{\substack{q(u|s_1,x)q'(\hat{x}|s_2,u) \\ :E[\mathfrak{d}(X,\hat{X})]\leq D}} I(U;X,S_1) - I(U;S_2), \tag{2.20}$$

where X and Y are the input and the output in the channel coding problem, and X and $\hat{X}$ are the source input and the reconstructed output in the source coding problem. In both problems, S_1 and S_2 are side information at the encoder and the decoder, respectively, and U is an auxiliary random variable. Apparently, like in WZC

and GPC, (2.19) and (2.20) themselves are optimization problems that are not trivial to solve. We will illustrate shortly that (2.19) and (2.20) can be computed using iterative algorithms. The main idea, divide-and-conquer, was used in the renowned papers by Blahut [13] and by Arimoto [7], where algorithms in computing the channel capacity and rate-distortion function without side information are devised. First, the optimization problem is divided into easier (convex/concave optimization) problems in which only a subset of variables are optimized with the rest fixed. Then, the solution to the partial optimization problem is fed into another sub-problem and another subset of variables is optimized. The algorithm will continue to iterate until all variables are optimized. This optimization technique was generalized in [36], where the EM algorithm [38] was included as a special case.

In the following subsections, we will derive our iterative algorithms, which resemble the algorithm in computing the capacity of defective computer memory in [50]. However, a simpler proof of convergence described in [111] is adopted. Two numerical examples are given in the last section to demonstrate our iterative algorithms.

1. Channel Capacity

We now derive our iterative algorithm in computing the capacity of the channel coding problem with two-sided state information described in Section IID. From (2.19),

$$C = \max_{q'(x|u,s_1)q(u|s_1)} \sum_{s_1,s_2,u,x,y} p(s_1,s_2)q(u|s_1)q'(x|u,s_1)p(y|x,s_1,s_2) \log \frac{Q_0(u|y,s_2)}{q(u|s_1)},$$

where $p(s_1,s_2)$ and $p(y|x,s_1,s_2)$ are determined by the channel and

$$Q_0(u|y,s_2) \triangleq \frac{\sum_{x,s_1} p(s_1,s_2)q(u|s_1)q'(x|u,s_1)p(y|x,s_1,s_2)}{\sum_{x,s_1,u} p(s_1,s_2)q(u|s_1)q'(x|u,s_1)p(y|x,s_1,s_2)}.$$

Define the functional

$$F(q, q', Q) = \sum_{s_1,s_2,u,x,y} p(s_1, s_2)q(u|s_1)q'(x|u, s_1)p(y|x, s_1, s_2) \log \frac{Q(u|y, s_2)}{q(u|s_1)},$$

and we have the following lemma.

Lemma 1:

$$C = \max_{q'(x|u,s_1)q(u|s_1)} \max_{Q(u|y,s_2)} F(q, q', Q). \tag{2.21}$$

Proof. Since $C = \max_{q'(x|u,s_1)q(u|s_1)} F(q, q', Q_0)$, it suffices to show $F(q, q', Q_0) = \max_{Q(u|y,s_2)} F(q, q', Q)$, which is true because for any Q,

$$\begin{aligned}
&F(q, q', Q) - F(q, q', Q_0) \\
&= \sum_{s_1,s_2,u,x,y} p(s_1, s_2)q(u|s_1)q'(x|u, s_1)p(y|x, s_1, s_2) \log \frac{Q(u|y, s_2)}{Q_0(u|y, s_2)} \\
&\overset{(a)}{\leq} \sum_{s_1,s_2,u,x,y} p(s_1, s_2)q(u|s_1)q'(x|u, s_1)p(y|x, s_1, s_2) \left(\frac{Q(u|y, s_2)}{Q_0(u|y, s_2)} - 1 \right) = 0,
\end{aligned}$$

where the equality in (a) is achieved if $Q = Q_0$. □

Lemma 1 is the key step of our algorithm. By introducing $F(\cdot, \cdot, \cdot)$, we can find the capacity via optimizing variables q, q' and Q one at a time alternatively. It is already known from Lemma 1 that the optimal Q is simply Q_0. Now for q, we have the following lemma.

Lemma 2: For fixed q' and Q, $F(q, q', Q)$ is maximized by

$$q^*(u|s_1) = \frac{\exp \sum_{s_2,x,y} p(s_2|s_1)q'(x|u, s_1)p(y|x, s_1, s_2) \log Q(u|y, s_2)}{\sum_u \exp \sum_{s_2,x,y} p(s_2|s_1)q'(x|u, s_1)p(y|x, s_1, s_2) \log Q(u|y, s_2)} \tag{2.22}$$

and

$$F(q^*, q', Q) = \sum_{s_1} p(s_1) \max_u \sum_{s_2,x,y} p(s_2|s_1) q'(x|u,s_1) p(y|x,s_1,s_2) \log \frac{Q(u|y,s_2)}{q(u|s_1)}. \tag{2.23}$$

Proof. For fixed q' and Q, $F(q, q', Q)$ is maximized by $q^*(u|s_1)$ if and only if the following Kuhn-Tucker conditions are satisfied:

$$\left.\frac{\partial F}{\partial q}\right|_{q^*} = \gamma_{s_1}, \qquad \text{if } q^*(u|s_1) > 0, \tag{2.24}$$

and

$$\left.\frac{\partial F}{\partial q}\right|_{q^*} \le \gamma_{s_1}, \qquad \text{if } q^*(u|s_1) = 0. \tag{2.25}$$

Since $\frac{\partial F}{\partial q} = \sum_{s_2,x,y} p(s_1,s_2) q'(x|u,s_1) p(y|x,s_1,s_2) \left(\log \frac{Q(u|y,s_2)}{q(u|s_1)} - 1\right)$, the first Kuhn-Tucker condition (2.24) becomes

$$\sum_{s_2,x,y} p(s_1,s_2) q'(x|u,s_1) p(y|x,s_1,s_2) \log \frac{Q(u|y,s_2)}{q(u|s_1)} = \tilde{\gamma}_{s_1}, \tag{2.26}$$

where $\tilde{\gamma}_{s_1}$ depends only on s_1. Then, (2.22) follows easily from (2.26) after some manipulation. For the second part, note that

$$\begin{aligned}
&F(q, q', Q) \\
=&\sum_{s_1,u} p(s_1) q(u|s_1) \sum_{s_2,x,y} p(s_2|s_1) q'(x|u,s_1) p(y|x,s_1,s_2) \log \frac{Q(u|y,s_2)}{q(u|s_1)} \\
\le&\sum_{s_1} p(s_1) \max_u \sum_{s_2,x,y} p(s_2|s_1) q'(x|u,s_1) p(y|x,s_1,s_2) \log \frac{Q(u|y,s_2)}{q(u|s_1)},
\end{aligned}$$

where equality holds when the Kuhn-Tucker conditions, and hence (2.26), are satisfied. That is, when q is equal to the optimal q^*. □

The results of Lemmas 1 and 2 can be summarized in the following corollary.

Corollary 1: For fixed q', $F(q, q', Q)$ is maximized by q^* and Q^* if

$$F(q^*, q', Q^*) = \sum_{s_1} p(s_1) \max_u \sum_{s_2,x,y} p(s_2|s_1) q'(x|u, s_1) p(y|x, s_1, s_2) \log \frac{Q_0(u|y, s_2)}{q(u|s_1)} \triangleq A_F. \tag{2.27}$$

Now, to optimize q' for fixed q and Q, note that

$$\begin{aligned} F(q, q', Q) &= \sum_{s_1,u,x} p(s_1) q(u|s_1) q'(x|u, s_1) \sum_{s_2,y} p(s_2|s_1) p(y|x, s_1, s_2) \log \frac{Q(u|y, s_2)}{q(u|s_1)} \\ &\leq \sum_{s_1,u} p(s_1) q(u|s_1) \max_x \underbrace{\sum_{s_2,y} p(s_2|s_1) p(y|x, s_1, s_2) \log \frac{Q(u|y, s_2)}{q(u|s_1)}}_{f(x,u,s_1)} \\ &\triangleq B_F. \end{aligned} \tag{2.28}$$

The equality holds if we select

$$q'(x|u, s_1) = \begin{cases} 1, & \text{if } f(x, u, s_1) = \max_{x'} f(x', u, s_1), \\ 0, & \text{otherwise.} \end{cases} \tag{2.29}$$

Note that there may be more than one q''s that optimize F. Let $S_{q'}(q, Q)$ be the set of q''s that achieves the maximum, then $||S_{q'}(q, Q)|| \leq ||\mathcal{X}||^{||\mathcal{U} \times \mathcal{S}_1||}$ is finite, where $\mathcal{U}$ and $\mathcal{S}_1$ are the alphabets of U and S_1, respectively. Combining (2.28) and Corollary 1, we have

Corollary 2:

$$F(q, q', Q) \leq \sum_{s_1} p(s_1) \max_u \max_x \sum_{s_2,y} p(s_2|s_1) p(y|x, s_1, s_2) \log \frac{Q(u|y, s_2)}{q(u|s_1)} \triangleq C_F$$

and equality holds if q' optimizes $F(q, q', Q)$ with the rest two variables fixed and q and Q optimize $F(q, q', Q)$ with q' fixed.

Note that $F(q, q', Q) = C_F$ does not promise $F(q, q', Q) = C$ since there are more than one optimal q''s in general. However, if $F(q, q', Q) = C_F$ for all $q' \in S_{q'}(q, Q)$,

then $F(q, q', Q) = C$.

The overall algorithm for computing C in (2.19) is summarized in Fig. 12. We initialize $q(u|s_1)$ as $\frac{1}{||\mathcal{U}||}$ and $q'(x|u, s_1)$ as random Kronecker delta functions of x for fixed u and s_1. We first optimize q and Q for fixed q'; F will then be compared with A_F to determined if q and Q are optimum. If so, q' will be updated as a unused element from $S_{q'}(q, Q)$. The process repeats until all elements in $S_{q'}(q, Q)$ are exhausted.

Proof of Convergence

We adopt a simpler proof of convergence introduced by Yueng in [111, Chapter 10], which shows that a two-step iterative maximization algorithm converge to the global optimum if the optimization function is concave. Therefore for fixed q' , our algorithm will converge to the general optimal $q^*(q')$ and $Q^*(q')$ since the following lemma holds.

Lemma 3: $F(q, q', Q)$ is concave over q and Q for fixed q'.

Proof. By the log-sum inequality, for an arbitrary $\gamma \leq 1$ and $\bar{\gamma} = 1 - \gamma$,

$$\begin{aligned}(\gamma q_1(u|s_1) + \bar{\gamma} q_2(u|s_1)) \log \frac{\gamma q_1(u|s_1) + \bar{\gamma} q_2(u|s_1)}{\gamma Q_1(u|y, s_2) + \bar{\gamma} Q_2(u|y, s_2)} \\ \leq \gamma q_1(u|s_1) \log \frac{q_1(u|s_1)}{Q_1(u|y, s_2)} + \bar{\gamma} q_2(u|s_1) \log \frac{q_2(u|s_1)}{Q_2(u|y, s_2)}.\end{aligned} \tag{2.30}$$

Taking reciprocal in the logarithms, multiplying both sides by $p(s_1, s_2)q'(x|u, s_1)p(y|x, s_1, s_2)$, and summing over s_1, s_2, u, x, and y, we obtain

$$F(\gamma q_1 + \bar{\gamma} q_2, q', \gamma Q_1 + \bar{\gamma} Q_2) \geq \gamma F(q_1, q', Q_1) + \bar{\gamma} F(q_2, q', Q_2).$$

□

Once q and Q are optimized, q' is updated by (2.29). Since F is strictly increasing in the algorithm and the number of q''s is finite, F will ultimately converge to the global optimum C.

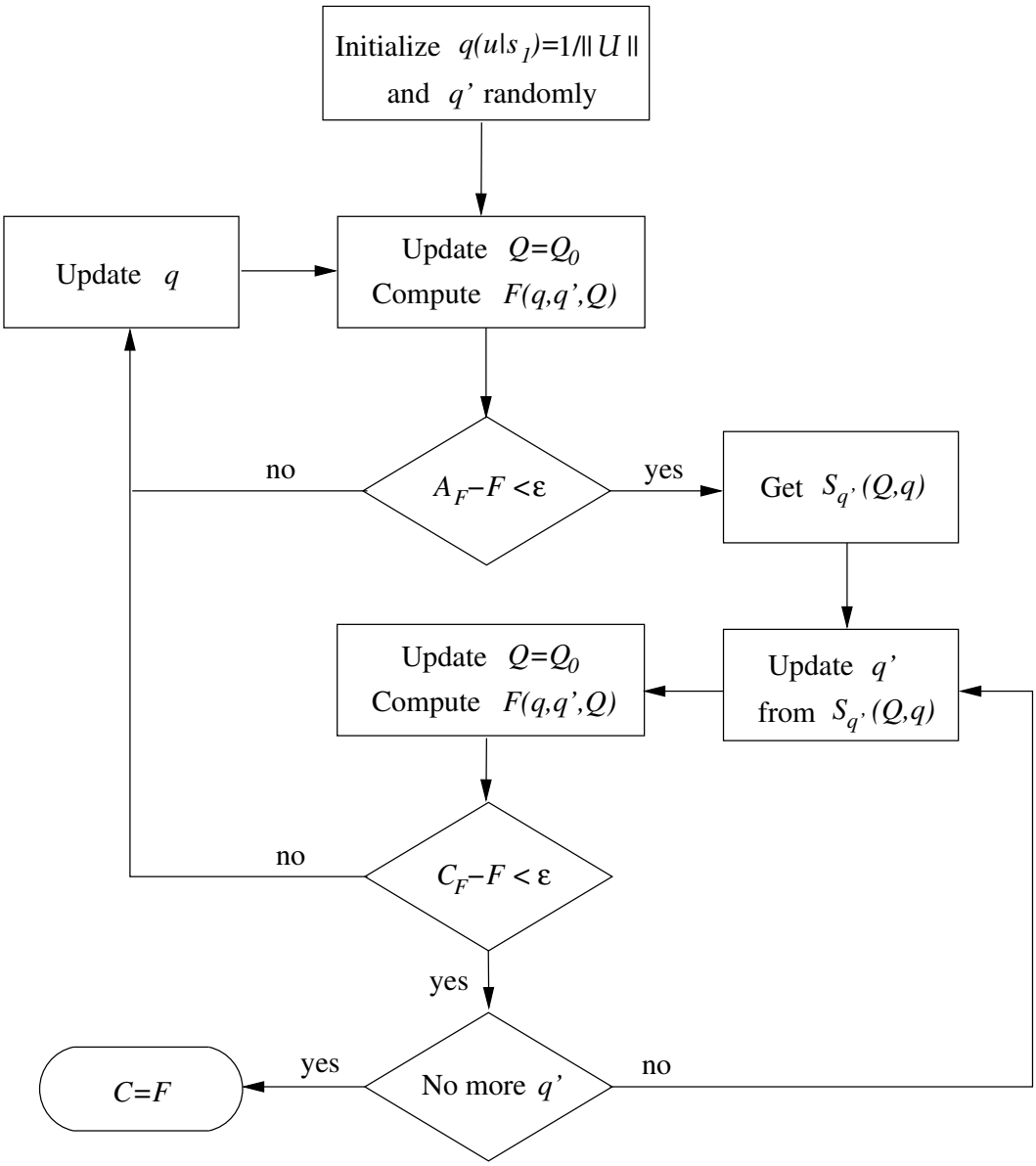

Fig. 12. Algorithm for computing capacity of a channel with side information.

2. Rate-Distortion Function

The iterative algorithm for computing the rate-distortion function with two-side state information is similar to that for capacity computation described in Section IID.1. However, the additional distortion constraint has to be taken into account. Using the standard Lagrange multiplier technique, we convert (2.20) into

$$R(D) = \min_{q(u|s_1,x),q'(\hat{x}|s_2,u)} I(U;X,S_1) - I(U;S_2) + \mu(E[\mathfrak{d}(X,\hat{X})] - D), \tag{2.31}$$

where μ, the Lagrange multiplier, rather than D is the actual input of computation. Both D and $R(D)$ are generated at the point where the $R(D)$ curve has slope $-\mu$. After optimization, D can be computed as

$$D = \sum_{s_1,s_2,x,u,\hat{x}} q^*(u|s_1,x)q'^*(\hat{x}|s_2,u)\mathfrak{d}(x,\hat{x}),$$

where $q^*(u|s_1,x)$ and $q'^*(\hat{x}|s_2,u)$ being the optimum conditional probabilities. Expand (2.31) and we have

$$\begin{aligned} R(D) = \min_{q(u|s_1,x),q'(\hat{x}|s_2,u)} & \sum_{s_1,s_2,x,u,\hat{x}} p(s_1,s_2,x)q(u|s_1,x)q'(\hat{x}|s_2,u)\log\frac{q(u|s_1,x)}{Q_0(u|s_2)} \\ & + \mu\left(\sum_{s_1,s_2,x,u,\hat{x}} p(s_1,s_2,x)q(u|s_1,x)q'(\hat{x}|s_2,u)\mathfrak{d}(x,\hat{x}) - D\right), \end{aligned}$$

where $Q_0(u|s_2)$ is the conditional probability induced by $p(s_1,s_2,x)$, $q(u|s_1,x)$, and $q'(\hat{x}|s_2,u)$. That is, $Q_0(u|s_2) \triangleq \frac{\sum_{s_1,x,\hat{x}} p(s_1,s_2,x)q(u|s_1,x)q'(\hat{x}|s_2,u)}{\sum_{s_1,x,\hat{x},u} p(s_1,s_2,x)q(u|s_1,x)q'(\hat{x}|s_2,u)}$.

Define the functional

$$G(q,q',Q)=\sum_{s_1,s_2,x,u,\hat{x}} p(s_1,s_2,x)q(u|s_1,x)q'(\hat{x}|s_2,u)\log\frac{q(u|s_1,x)}{Q(u|s_2)}$$
$$+\mu\sum_{s_1,s_2,x,u,\hat{x}} p(s_1,s_2,x)q(u|s_1,x)q'(\hat{x}|s_2,u)\mathfrak{d}(x,\hat{x}),$$

and we have the following lemma in contrast to Lemma 1.

Lemma 4:

$$R(D)=\min_{q(u|s_1,x),q'(\hat{x}|s_2,u)}\min_{Q(u|s_2)} G(q,q',Q)-\mu D. \tag{2.32}$$

Proof. Since $R(D)=\min_{q(u|s_1,x),q'(\hat{x}|s_2,u)}(G(q,q',Q_0)-\mu D)=\min_{q(u|s_1,x),q'(\hat{x}|s_2,u)} G(q,q',Q_0)-\mu D$. It suffices to show

$$\min_{Q(u|s_2)} G(q,q',Q)=G(q,q',Q_0),$$

which is true because for any Q,

$$\begin{aligned}
&G(q,q,Q_0)-G(q,q',Q)\\
&=\sum_{s_1,s_2,x,u,\hat{x}} p(s_1,s_2,x)q(u|s_1,x)q'(\hat{x}|s_2,u)\log\frac{Q(u|s_2)}{Q_0(u|s_2)}\\
&\le\sum_{s_1,s_2,x,u,\hat{x}} p(s_1,s_2,x)q(u|s_1,x)q'(\hat{x}|s_2,u)\left(\frac{Q(u|s_2)}{Q_0(u|s_2)}-1\right)\\
&=0,
\end{aligned}$$

where equality is achieved if $Q=Q_0$. □

Just as Lemma 1 in the capacity computation algorithm, Lemma 4 is the key step of the rate-distortion computation algorithm. We can now find the minimum rate R by optimizing variables q, q', and Q one at a time alternatively. The optimal value of Q is Q_0 from Lemma 4. Now to optimize q, we have the following lemma in contrast to Lemma 2.

Lemma 5: For fixed q' and Q, $G(q, q', Q)$ is minimized by

$$q^*(u|s_1, x) = \frac{\exp\left[\sum_{s_2} p(s_2|s_1, x) \log Q(u|s_2) - \mu \sum_{s_2, \hat{x}} p(s_2|s_1, x) q'(\hat{x}|s_2, u) \mathfrak{d}(x, \hat{x})\right]}{\sum_u \exp\left[\sum_{s_2} p(s_2|s_1, x) \log Q(u|s_2) - \mu \sum_{s_2, \hat{x}} p(s_2|s_1, x) q'(\hat{x}|s_2, u) \mathfrak{d}(x, \hat{x})\right]} \tag{2.33}$$

and

$$\begin{aligned} G(q^*, q', Q) = \sum_{s_1, x} p(s_1, x) \min_u \Bigg[& \sum_{s_2, \hat{x}} p(s_2|s_1, x) q'(\hat{x}|s_2, u) \log \frac{q(u|s_1, x)}{Q(u|s_2)} \\ & + \mu \sum_{s_2, \hat{x}} p(s_2|s_1, x) q'(\hat{x}|s_2, u) \mathfrak{d}(x, \hat{x}) \Bigg]. \end{aligned} \tag{2.34}$$

Proof. For fixed q' and Q, $G(q, q', Q)$ is minimized if and only if the following Kuhn-Tucker conditions are satisfied:

$$\left.\frac{\partial G}{\partial q}\right|_{q^*} = \gamma_{s_1, x}, \qquad \text{if } q^*(u|s_1, x) > 0, \tag{2.35}$$

and

$$\left.\frac{\partial G}{\partial q}\right|_{q^*} \leq \gamma_{s_1, x}, \qquad \text{if } q^*(u|s_1, x) = 0. \tag{2.36}$$

Since

$$\begin{aligned} \frac{\partial G}{\partial q} = & \sum_{s_2, \hat{x}} p(s_1, s_2, x) q'(\hat{x}|s_2, u) \left(\log \frac{q(u|s_1, x)}{Q(u|s_2)} + 1 \right) \\ & + \mu \sum_{s_2, \hat{x}} p(s_1, s_2, x) q'(\hat{x}|s_2, u) \mathfrak{d}(x, \hat{x}), \end{aligned} \tag{2.37}$$

the first Kuhn-Tucker condition (2.35) becomes

$$\sum_{s_2,\hat{x}} p(s_2|s_1,x)q'(\hat{x}|s_2,u)\log\frac{q(u|s_1,x)}{Q(u|s_2)} + \mu\sum_{s_2,\hat{x}} p(s_2|s_1,x)q'(\hat{x}|s_2,u)\mathfrak{d}(x,\hat{x}) = \tilde{\gamma}_{s_1,x}, \tag{2.38}$$

where $\tilde{\gamma}_{s_1,x}$ depends only on s_1 and x. Then (2.33) follows from (2.38) after some manipulation. For the second part, note that

$$\begin{aligned}
&G(q,q',Q)\\
&= \sum_{u,s_1,x} p(s_1,x)q(u|s_1,x)\left[\sum_{s_2,\hat{x}} p(s_2|s_1,x)q'(\hat{x}|s_2,u)\log\frac{q(u|s_1,x)}{Q(u|s_2)}\right.\\
&\qquad\left. + \mu\sum_{s_2,\hat{x}} p(s_2|s_1,x)q'(\hat{x}|s_2,u)\mathfrak{d}(x,\hat{x})\right]\\
&\leq \sum_{s_1,x} p(s_1,x)\min_u\left[\sum_{s_2,\hat{x}} p(s_2|s_1,x)q'(\hat{x}|s_2,u)\log\frac{q(u|s_1,x)}{Q(u|s_2)}\right.\\
&\qquad\left. + \mu\sum_{s_2,\hat{x}} p(s_2|s_1,x)q'(\hat{x}|s_2,u)\mathfrak{d}(x,\hat{x})\right],
\end{aligned}$$

where equality holds when the Kuhn-Tucker conditions, hence (2.38), are satisfied. That is when q is equal to the optimal q^*. □

In contrast to Corollary 1, Lemmas 4 and 5 can be summarized by the following corollary.

Corollary 3: For fixed q', $G(q,q',Q)$ is minimized by q^* and Q^* if

$$
\begin{aligned}
& G(q^*, q', Q^*) \\
=& \sum_{s_1,x} p(s_1,x) \min_u \left[\sum_{s_2,\hat{x}} p(s_2|s_1,x) q'(\hat{x}|s_2,u) \log \frac{q(u|s_1,x)}{Q_0(u|s_2)} \right. \\
& \left. + \mu \sum_{s_2,\hat{x}} p(s_2|s_1,x) q'(\hat{x}|s_2,u) \mathfrak{d}(x,\hat{x}) \right] \triangleq A_G.
\end{aligned} \tag{2.39}
$$

To optimize q' for fixed q and Q, note that

$$
\begin{aligned}
& G(q, q', Q) \\
=& \sum_{\hat{x},u,s_2} q'(\hat{x}|s_2,u) \underbrace{\left[\sum_{s_1,x} p(s_1,s_2,x) q(u|s_1,x) \left(\log \frac{q(u|s_1,x)}{Q(u|s_2)} + \mu \mathfrak{d}(x,\hat{x}) \right) \right]}_{g(u,s_2,\hat{x})} \\
\leq & \sum_{u,s_2} \min_{\hat{x}} g(u,s_2,\hat{x}) \triangleq B_G,
\end{aligned} \tag{2.40}
$$

where equality holds if we select

$$
q'(\hat{x}|u,s_2) = \begin{cases} 1, & \text{if } g(\hat{x},u,s_2) = \min_{\hat{x}'} g(\hat{x}',u,s_2), \\ 0, & \text{otherwise.} \end{cases} \tag{2.41}
$$

Similar to that in the channel coding problem, there may be more than one q''s that optimize G. Let $S_{q'}(q,Q)$ be the set of q''s that achieves the minimum, then $||S_{q'}(q,Q)|| \leq ||\hat{\mathcal{X}}||^{||\mathcal{U}\times\mathcal{S}_2||}$.

Unlike in capacity computation, we need to verify both conditions, $G = A_G$ and $G = B_G$, for optimality since there is no simple way in combining (2.39) and (2.40). However, as in capacity computation, even when both conditions are satisfied, $G(q,q',Q)$ may not be the global optimal since there are more than one optimal q''s in general. However, if the above two conditions are satisfied for all $q' \in S_{q'}(q,Q)$, then $R(D) = G(q,q',Q) - \mu D$.

The overall algorithm is summarized in Fig. 13. The procedure is similar to that for capacity computation.

Proof of Convergence

The same argument is used as in the channel coding case except that the maximization problem is replaced by a minimization one. Therefore we only need to show the following lemma to prove convergence.

Lemma 6: $G(q, q', Q)$ is convex over q and Q for fixed q'.

Proof. Using the log-sum inequality, we can show $\sum_{s_1,s_2,x,u,\hat{x}} p(s_1, s_2, x)q(u|s_1, x)q'(\hat{x}|s_2, u) \log \frac{q(u|s_1,x)}{Q(u|s_2)}$ to be convex over q and Q for fixed q'. Since $\mu \sum_{s_1,s_2,x,u,\hat{x}} p(s_1, s_2, x)q(u|s_1, x)q'(\hat{x}|s_2, u)\mathfrak{d}(x, \hat{x})$ is linear with respect to q and Q. The sum of the two expressions, i.e., G, is convex. □

3. Capacity-Power Function

In some cases, it is necessary to constrain the transmission power in a communication system. The transmission power is only a function of X in conventional communication system. However, to allow channel coding to model other problems such as watermarking, a more general power function $\mathfrak{p}(S_1, S_2, X)$ that also depends on S_1 is considered here. Hence the capacity-power function is

$$C(P) = \max_{\substack{q(u|s_1)q'(x|s_1,u) \\ :E[\mathfrak{p}(S_1,S_2,X)]\leq P}} I(U; Y, S_2) - I(U; S_1). \tag{2.42}$$

The derivation of the capacity-power function is almost the same as the previous two cases. Hence, we will only state the results and skip all the proofs. Using the standard Lagrange multiplier technique, we convert (2.42) into

$$C(P) = \max_{q(u|s_1)q'(x|s_1,u)} I(U; Y, S_2) - I(U; S_1) - \mu(E[\mathfrak{p}(S_1, S_2, X)] - P), \tag{2.43}$$

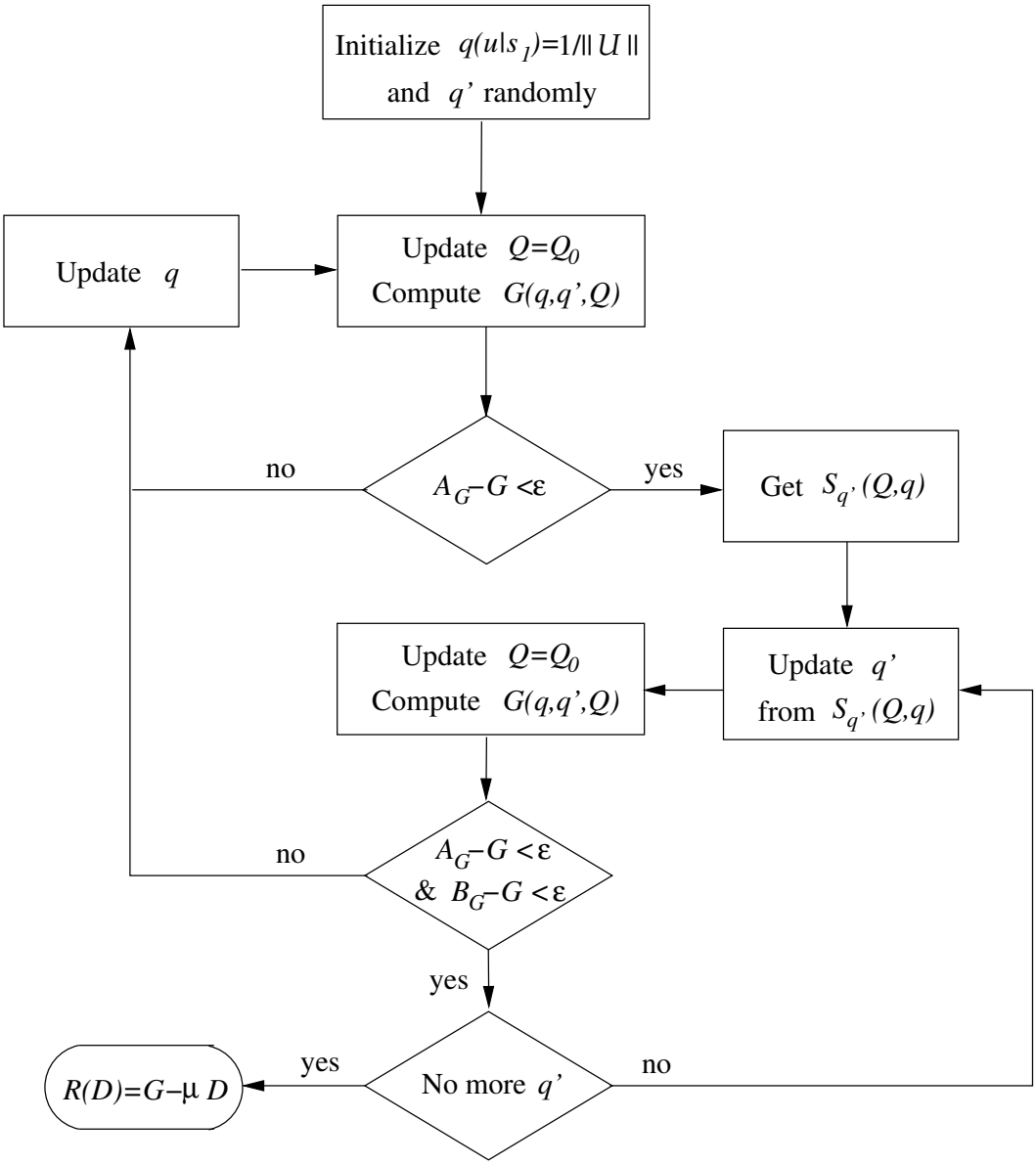

Fig. 13. Algorithm for computation of rate-distortion function with side information

where μ, the Lagrange multiplier, rather than P is the actual input of computation. Both P and $C(P)$ are generated at the point where $C(P)$ curve has slope μ. After optimization, P can be computed as

$$P = \sum_{s_1,x,u} p(s_1)q^*(u|s_1)q'^*(x|s_1,u)\mathfrak{p}(s_1,s_2,x),$$

where $q^*(u|s_1)$ and $q'^*(x|s_1,u)$ being the optimum conditional probabilities. Expand (2.43) and we have

$$C(P) = \max_{q'(x|u,s_1)q(u|s_1)} \sum_{s_1,s_2,u,x,y} p(s_1,s_2)q(u|s_1)q'(x|u,s_1)p(y|x,s_1,s_2)\log\frac{Q_0(u|y,s_2)}{q(u|s_1)} - \mu\left(\sum_{s_1,x,u} p(s_1)q(u|s_1)q'(x|s_1,u)\mathfrak{p}(s_1,s_2,x) - P\right),$$

where $Q_0(u|y,s_2)$ is the conditional probability induced by $p(s_1,s_2)$, $q(u|s_1)$, $q'(x|u,s_1)$, and $p(y|x,s_1,s_2)$. That is,

$$Q_0(u|y,s_2) \triangleq \frac{\sum_{x,s_1} p(s_1,s_2)q(u|s_1)q'(x|u,s_1)p(y|x,s_1,s_2)}{\sum_{x,s_1,u} p(s_1,s_2)q(u|s_1)q'(x|u,s_1)p(y|x,s_1,s_2)}.$$

Define the functional

$$F_c(q,q',Q) = \sum_{s_1,s_2,u,x,y} p(s_1,s_2)q(u|s_1)q'(x|u,s_1)p(y|x,s_1,s_2)\log\frac{Q(u|y,s_2)}{q(u|s_1)} - \mu\sum_{s_1,x,u} p(s_1)q(u|s_1)q'(x|s_1,u)\mathfrak{p}(s_1,s_2,x),$$

and we have the following lemma in contrast to Lemmas 1 and 4.

Lemma 7:

$$C(P) = \max_{q(u|s_1),q'(x|s_1,u)} \max_{Q(u|y,s_2)} F_c(q,q',Q) + \mu P. \tag{2.44}$$

From Lemma 7, we can find $C(P)$ by maximizing F_c one variable at a time. It is already known that the optimum Q is Q_0. To optimize q, we have the following

lemma in contrast to Lemmas 2 and 5.

Lemma 8: For fixed q' and Q, $F_c(q, q', Q)$ is maximized by

$$q^*(u|s_1, x) = \frac{\exp\left[\sum_{s_2,x,y} p(s_2|s_1)q'(x|s_1, u)p(y|x, s_1, s_2)[\log Q(u|y, s_2) - \mu\mathfrak{p}(x, s_1)]\right]}{\sum_u \exp\left[\sum_{s_2,x,y} p(s_2|s_1)q'(x|s_1, u)p(y|x, s_1, s_2)[\log Q(u|y, s_2) - \mu\mathfrak{p}(x, s_1)]\right]} \tag{2.45}$$

and

$$\begin{aligned} &F_c(q^*, q', Q) \\ =&\sum_{s_1} p(s_1) \max_u \sum_{s_2,x,y} \left[p(s_2|s_1)q'(x|u, s_1)p(y|x, s_1, s_2)\left(\log \frac{Q(u|y, s_2)}{q(u|s_1)} - \mu\mathfrak{p}(s_1, s_2, x)\right)\right]. \end{aligned}$$

Lemmas 7 and 8 can be summarized by the following corollary in contrast to Corollaries 1 and 3.

Corollary 4: For fixed q', $F_c(q, q', Q)$ is minimized by q^* and Q^* if

$$\begin{aligned} &F_c(q^*, q', Q^*) \\ =&\sum_{s_1} p(s_1) \max_u \sum_{s_2,x,y} \left[p(s_2|s_1)q'(x|u, s_1)p(y|x, s_1, s_2)\left(\log \frac{Q_0(u|y, s_2)}{q(u|s_1)} - \mu\mathfrak{p}(s_1, s_2, x)\right)\right] \\ \triangleq&A_{Fc}. \end{aligned}$$

Now, To optimize q' for fixed q and Q, note that

$$
\begin{aligned}
&F_c(q,q',Q)\\
&=\sum_{s_1,u,x} p(s_1)q(u|s_1)q'(x|u,s_1)\sum_{s_2,y}\left[p(s_2|s_1)p(y|x,s_1,s_2)\left(\log\frac{Q(u|y,s_2)}{q(u|s_1)}-\mu\mathfrak{p}(s_1,s_2,x)\right)\right]\\
&=\sum_{s_1,u} p(s_1)q(u|s_1)\max_x \underbrace{\sum_{s_2,y}\left[p(s_2|s_1)p(y|x,s_1,s_2)\left(\log\frac{Q(u|y,s_2)}{q(u|s_1)}-\mu\mathfrak{p}(s_1,s_2,x)\right)\right]}_{f_c(x,u,s_1)}\\
&\triangleq B_{Fc}. \qquad (2.46)
\end{aligned}
$$

The equality holds if we select

$$
q'(x|u,s_1)=\begin{cases}1, & \text{if } f_c(x,u,s_1)=\max\limits_{x'} f_c(x',u,s_1),\\ 0, & \text{otherwise.}\end{cases} \qquad (2.47)
$$

Like the previous two cases, since there may be more than one q''s that optimize F_c, let $S_{q'}(q,Q)$ be the set of q''s that achieves the maximum, then $||S_{q'}(q,Q)|| \leq ||\mathcal{X}||^{||\mathcal{U}\times\mathcal{S}_1||}$ is finite, where $\mathcal{U}$ and $\mathcal{S}_1$ are the alphabets of U and S_1, respectively. Combining (2.46) and Corollary 4, we have the following corollary in contrast to Corollary 2.

Corollary 5:

$$
\begin{aligned}
&F_c(q,q',Q)\\
&\leq\sum_{s_1} p(s_1)\max_u\max_x\sum_{s_2,y}\left[p(s_2|s_1)p(y|x,s_1,s_2)\left(\log\frac{Q(u|y,s_2)}{q(u|s_1)}-\mu\mathfrak{p}(s_1,s_2,x)\right)\right]\triangleq C_{Fc}
\end{aligned}
$$

and equality holds if q' optimizes $F_c(q,q',Q)$ with the rest two variables fixed and q and Q optimize $F_c(q,q',Q)$ with q' fixed.

Like capacity computation without power constraint, $F_c(q,q',Q)=C_{Fc}$ does not promise $F_c(q,q',Q)$ to be the global optimal since there are more than one optimal

q''s in general. However, if $F_c(q, q', Q) = C_{Fc}$ for all $q' \in S_{q'}(q, Q)$, then $C(P) = F_c(q, q', Q) + \mu P$.

The overall algorithm is summarized in Fig. 14. The procedure is very similar to those in the previous two cases.

Proof of Convergence

Similar to previous cases, we need to show the following lemma to prove convergence.

Lemma 9: $F_c(q, q', Q)$ is concave over q and Q for fixed q'.

Proof. From Lemma 3, $F(q, q', Q)$ is concave. Since $F_c(q, q', Q) = F(q, q', Q) - \mu E[\mathfrak{p}(S_1, S_2, X)]$ and $\mu E[\mathfrak{p}(S_1, S_2, X)]$ is linear with respect to q and Q, $F_c(q, q', Q)$ is concave. □

4. Numerical Examples

In this section, we provide numerical examples for our iterative algorithms. As we shall see, while the setups of these examples are rather simple, the results are highly non-trivial.

Example 1: Binary Symmetric Channel with Channel State Information

Consider a binary symmetric channel $Y = X \bigoplus \tau \bigoplus Z$ as shown in Fig. 15, where X is the channel input and τ and Z are the channel noises. The transition probability of τ is fixed to be P_τ, whereas the transition probability of Z can take two different values and is controlled by a binary random variable θ with $p(\theta = 1) = p_\theta$ as follows:

$$P_Z = \begin{cases} P_{Z_1}, & \text{if } \theta = 1, \\ P_{Z_0}, & \text{if } \theta = 0. \end{cases}$$

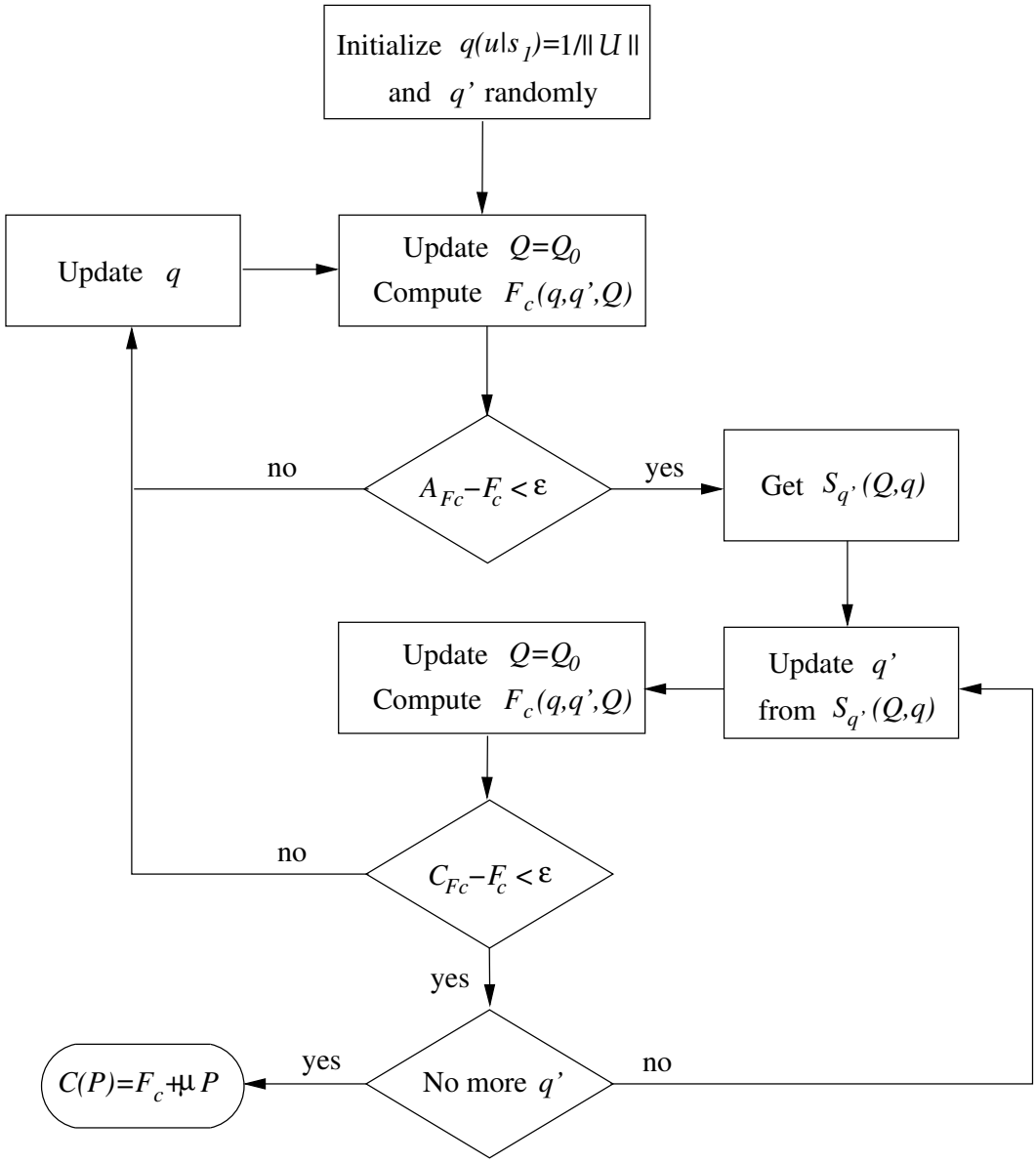

Fig. 14. Algorithm for computation of capacity-power function with side information

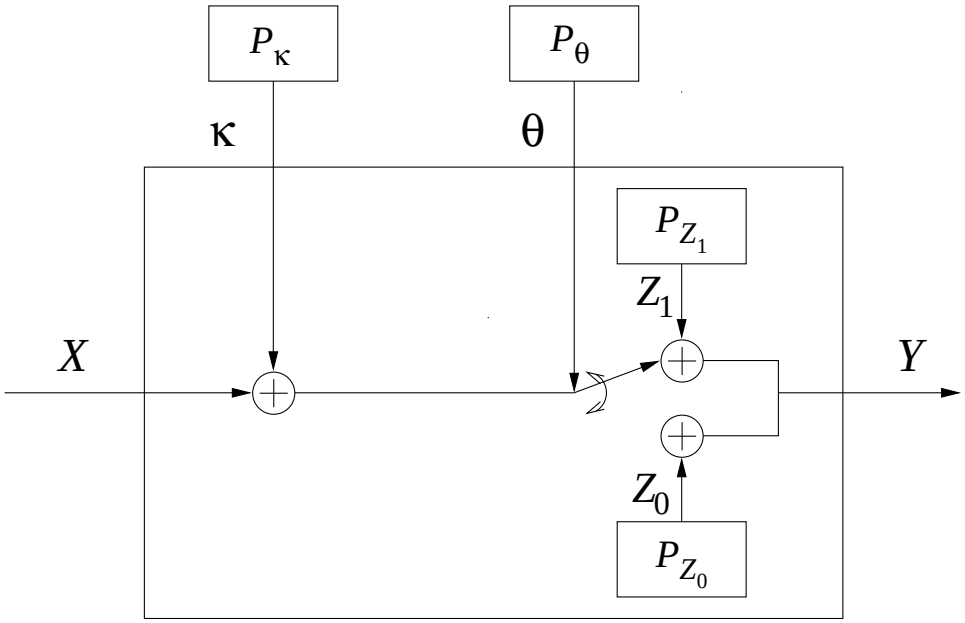

Fig. 15. Binary symmetric channel with channel state information θ and τ.

Consider θ and/or τ as channel state information that may be available to the encoder and decoder. Since each coder can have 4 combination of side information (with both state information, with only θ, with only τ, or with none of them), there are totally 16 different cases.

We use our algorithm described in Section IID.1 with $P_\tau = 0.5, P_{Z_1} = 0.001$, and $P_{Z_0} = 0.3$. Since $P_\tau = 0.5$, when τ is not given to either coder, X and Y are effectively independent and hence for all these 4 cases, the channel capacity is simply 0. This is verified in our result. More interestingly, the 16 cases can be grouped into only 3 cases as shown in Table III, where the capacity for each case is plotted in Fig. 16. Furthermore, when τ is available at either coder, we can reach the higher capacity C_2 only if θ is available at the decoder.

Example 2: Binary Symmetric Source with Side Information

Consider the source generated by passing an all-zero sequence through the binary

Table III. Channel capacities for different cases in Example 1. (C_1 and C_2 are illustrated in Fig. 16 for different P_θ's.)

Capacity	Cases	
0	$S_1 = \emptyset, S_2 = \emptyset$;	$S_1 = \{\theta\}, S_2 = \emptyset$;
	$S_1 = \emptyset, S_2 = \{\theta\}$;	$S_1 = \{\theta\}, S_2 = \{\theta\}$
C_1	$S_1 = \emptyset, S_2 = \{\tau\}$;	$S_1 = \{\tau\}, S_2 = \emptyset$;
	$S_1 = \{\theta\}, S_2 = \{\tau\}$;	$S_1 = \{\tau\}, S_2 = \{\tau\}$;
	$S_1 = \{\theta, \tau\}, S_2 = \emptyset$;	$S_1 = \{\theta, \tau\}, S_2 = \{\tau\}$
C_2	$S_1 = \emptyset, S_2 = \{\theta, \tau\}$;	$S_1 = \{\tau\}, S_2 = \{\theta\}$;
	$S_1 = \{\theta\}, S_2 = \{\theta, \tau\}$;	$S_1 = \{\tau\}, S_2 = \{\theta, \tau\}$;
	$S_1 = \{\theta, \tau\}, S_2 = \{\theta\}$;	$S_1 = \{\theta, \tau\}, S_2 = \{\theta, \tau\}$

symmetric channel described in Example 1 (see Fig. 15) and assume the same numerical setting with $p_\tau = 0.5$, $p_{Z_1} = 0.01$, and $p_{Z_0} = 0.3$. We compute the rate-distortion functions for this source when $p_\theta = 0.5$. Like in the previous example, τ and/or θ may be provided to the source encoder and decoder as side information, and hence we have totally 16 different cases. Interestingly, these 16 cases can be grouped into only 5 cases as shown in Table IV, where the rate-distortion function for each case is plotted in Fig. 17. The reasons of some of these degenerate cases are apparent. For instance, if τ is given to neither the encoder nor the decoder, the source is effectively just a binary symmetric source regardless of the availability of θ. Hence, the rate-distortion function for these cases should be the same as that for a binary symmetric source with completely no side information. Another interesting observation is that side information is not helpful if it is provided to the encoder alone; for instance, the case $S_1 = \emptyset, S_2 = \emptyset$ and the case $S_1 = \{\theta, \tau\}, S_2 = \emptyset$ share the same rate-distortion

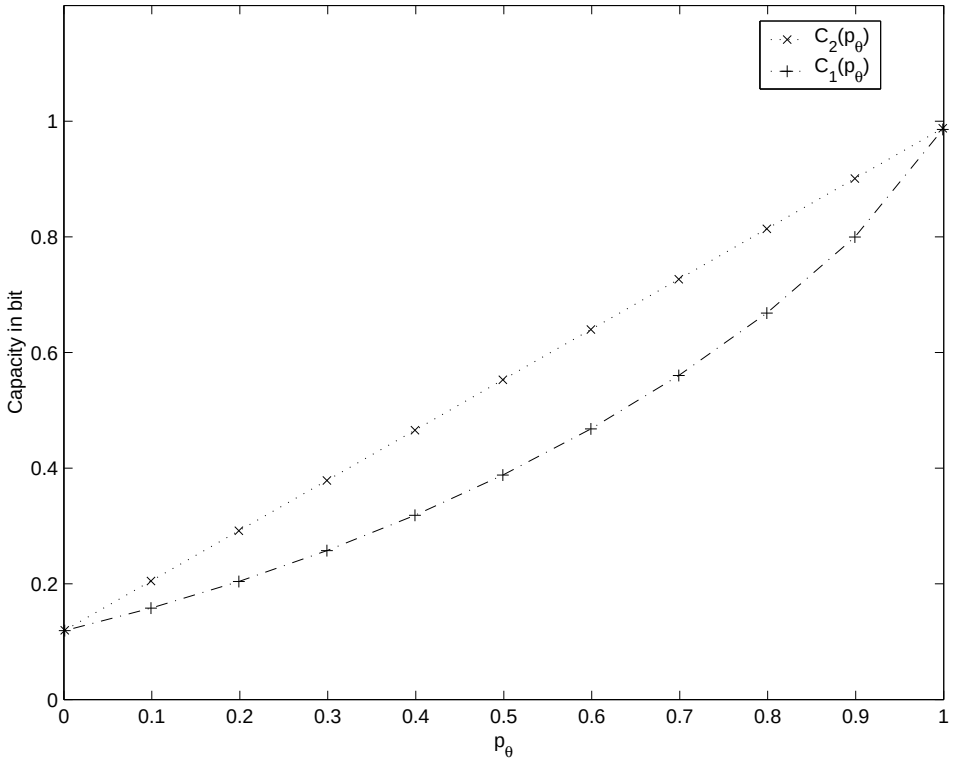

Fig. 16. Channel capacity C versus p_θ for different cases in Example 1.

function. This is consistent with the classic result by Berger in [11].

Table IV. Rate-distortion function for different cases in Example 2. ($R_1(D)$, $R_2(D)$, $R_3(D)$, $R_4(D)$, and $R_5(D)$ are illustrated in Fig. 17 for different μ's.)

R-D function	Cases	
$R_1(D)$	$S_1 = \emptyset, S_2 = \emptyset;$	$S_1 = \{\theta\}, S_2 = \emptyset;$
	$S_1 = \emptyset, S_2 = \{\theta\};$	$S_1 = \{\theta\}, S_2 = \{\theta\};$
	$S_1 = \{\tau\}, S_2 = \emptyset;$	$S_1 = \{\tau\}, S_2 = \{\theta\};$
	$S_1 = \{\theta, \tau\}, S_2 = \emptyset;$	$S_1 = \{\theta, \tau\}, S_2 = \{\theta\}$
$R_2(D)$	$S_1 = \emptyset, S_2 = \{\tau\};$	$S_1 = \{\theta\}, S_2 = \{\tau\}$
$R_3(D)$	$S_1 = \{\tau\}, S_2 = \{\tau\};$	$S_1 = \{\tau, \theta\}, S_2 = \{\tau\}$
$R_4(D)$	$S_1 = \emptyset, S_2 = \{\theta, \tau\};$	$S_1 = \{\theta\}, S_2 = \{\theta, \tau\}$
$R_5(D)$	$S_1 = \{\tau\}, S_2 = \{\theta, \tau\};$	$S_1 = \{\theta, \tau\}, S_2 = \{\theta, \tau\}$

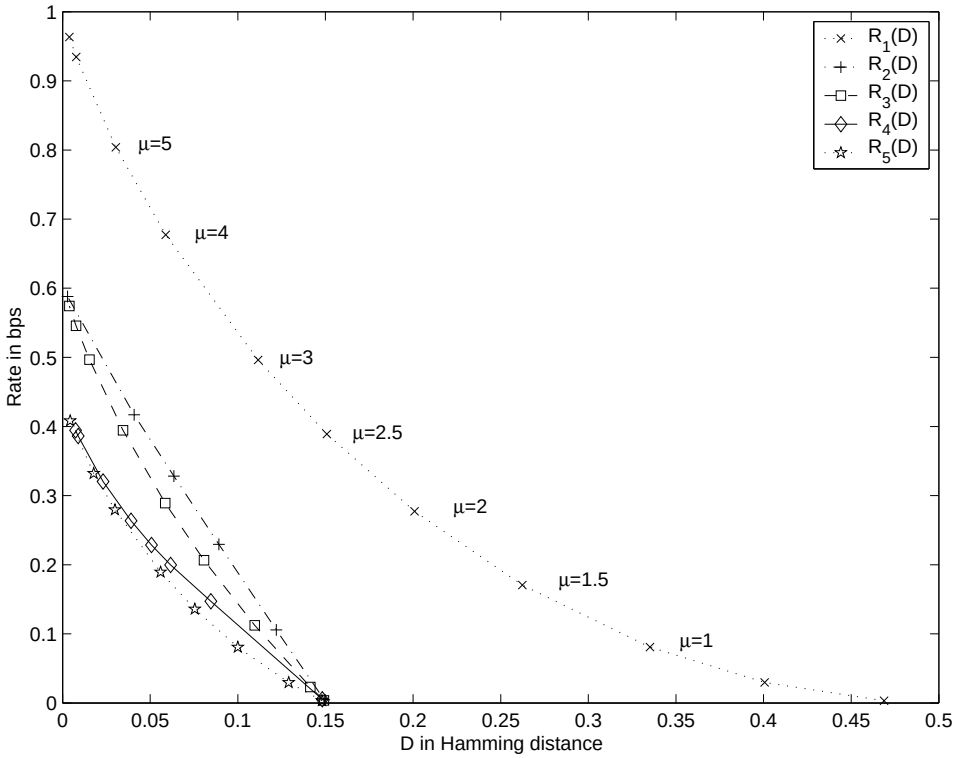

Fig. 17. Rate-distortion functions for different cases in Example 2.

CHAPTER III

WYNER-ZIV CODING DESIGN

In this chapter, we will focus on the design of Wyner-Ziv coding (WZC) [104]. In the first section, we will first investigate the lossless case when the distortion of the reconstructed source is 0. This case is commonly known as Slepian-Wolf Coding (SWC) [87]. We describe how SWC can be implemented using conventional channel coding. In specific, we detail the SWC design using the low-density parity-check (LDPC) code [43]. We point out that the SWC performance is needed to be independent of the input source to facilitate efficient SWC design. We show that a sufficient condition of this assumption is that the hypothetical channel between the source and the side information satisfies a symmetry condition dubbed *dual symmetry*. Moreover, when dual symmetry is satisfied, the LDPC code performance over the hypothetical channel precisely translates to the SWC performance. Therefore, under that dual symmetry condition, SWC design problem can be simply treated as LDPC coding design over the hypothetical channel.

When the distortion of the reconstructed source can be non-zero, we propose a practical WZC paradigm dubbed *Slepian-Wolf coded quantization (SWCQ)* by combining SWC and nested lattice quantization [112], where nested lattice quantization is just a special case of nested coding described in Section IIB.1. We point out an interesting analogy between SWCQ and entropy coded quantization [48] in classic source coding. A practical scheme of SWCQ using 1-D nested lattice quantization and LDPC is implemeted, where detail design issues are discussed.

A. Slepian-Wolf Coding: Zero Distortion Case

When the distortion of the reconstructed source is forced to be 0, WZC degenerates to lossless source coding with side information at the decoder. We can easily visualize this as a special case of SWC, a synonym of lossless distributed source coding, if we only code one of the sources and treat all other sources in conventional SWC as side information. Therefore, this setup is also known as the asymmetric SWC. However, we will simply call it SWC from now on as we will only consider this asymmetric case.

1. General Approaches

a. Random Binning

Let V and S be the source and side information, respectively. Since we are considering lossless coding, V has a finite alphabet in general.

We will consider a block code of length-n here. The idea of random binning [30, pp. 410-413] is to partition all the length-n sequences of V randomly into bins and only the indices of these bins are transmitted to the decoder. For an i.i.d. discrete source V, the set of all length-n sequences generated by V is randomly partitioned into 2^{nR} bins. Hence, if we compress V at rate R, there should be 2^{nR} bins.

Knowing the bin index and the sequence of side information S^n, the decoder reconstructs $\hat{V}^n$ as the sequence that is jointly typical[1] with S^n and lies inside the desired bin. We can interpret the above reconstruction process as a channel decoding procedure and S as the output of a hypothetical channel with input V. Therefore, for a sufficiently large n, V can be reconstructed with arbitrarily small error probability

[1]This may involve joint typicality of a continuous random variable and a discrete random variable since S may be continuous. However, such joint typicality can be easily obtained by generalizing the classic case.

as long as the rate of transmission via this hypothetical channel is less than $I(V;S)$, or in other words, each bin can have maximally $\approx 2^{nI(V;S)}$ elements to have lossless reconstruction. Since the total number of typical sequence of V with length n is approximately $2^{nH(V)}$, the number of bins required is $2^{nH(V)}/2^{nI(V;S)} = 2^{nH(V|S)}$. Hence, we can compress V at a rate $H(V|S)$ with this random binning scheme.

Assume now side information S is also given to the encoder. For the instance when $S = s$, we can optimally compress V at rate $H(V|s)$ using classic source coding. Hence, the optimal average compression rate is $\sum_s H(V|s)p(s) = H(V|S)$. Comparing this rate with that obtained by random binning scheme in SWC, we can draw two important conclusions. First, the random binning scheme must attain maximum possible compression since it cannot outperform the optimal scheme in the better equipped setup when side information is also provided to the encoder. Second, contrary to the fact that WZC setup has rate loss in general (see Section IIA.1), SWC setup has no rate loss comparing with this better equipped setup when side information is also given to the encoder.

b. Structure Binning

Unfortunately, the random binning scheme is not friendly to implement. The main difficulty is to assign a random binning that yet can facilitate decoding with low computational complexity. However, a more detail observation of our previous discuss concludes that purely random assignment of codewords is not necessary; it is more important instead to have each bin to behave like a good channel code so as to approach the hypothetical channel capacity $I(V;S)$.

An interesting approach that was first suggested by Wyner [105] and was rediscovered and first implemented by Pradhan and Ramchandran [76] is to use an arbitrary linear channel code to partition the set of all v^n's into cosets or bins with

different syndromes. Since all cosets of a linear channel code share the same distance properties, all bins (cosets) now are indeed good channel code as desired provided that the linear channel code itself is good. Note that the syndrome now acts as the bin index to be transmitted at the encoder. Hence, for the (n, k)-channel code and thus with a $(n-k) \times n$ parity matrix $\mathbf{H}$, Slepian-Wolf encoding is to compute and output the syndrome

$$w^{n-k} = v^n \mathbf{H}^T.$$

Since the length of the syndrome is $n-k$, the compression ratio is $n : n-k$.

In order to perform Slepian-Wolf decoding, channel decoding is modified in such a way that $\hat{v}^n$ is reconstructed as a code vector inside the coset with the desired syndrome instead of a codeword of the channel code. More precisely, receiving s^n, the decoder should select from the bin that maximize the a posteriori probability, i.e.,

$$\hat{v}^n = \arg \max_{v \in \{v' | w^m = v'^n \mathbf{H}^T\}} p(v^n | s^n). \tag{3.1}$$

c. Multilevel Slepian-Wolf Coding

Since in general the bin index V has an alphabet size larger than two, we may need a non-binary channel code to implement SWC. As a non-binary code is usually harder to deploy and design, a better alternative is to first map V into its binary representation $B_0, B_1, ..., B_{\Lambda-1}$ and then code $B_i, i = 0, ..., \Lambda - 1$ one level at a time. More precisely, each bit level now employs a different channel code and generates its own syndrome during encoding, and then multistage decoding [51] is employed. Note that bits obtained from previous stages will be used along with S as side information for decoding the bit in the current stage. If we assign SWC rate for the i^{th} stage as $H(B_i|S, B_0, ...B_{i-1})$ and assume bits from previous stages are perfectly recovered, B_i can be reconstructed losslessly with the given rate by the "no rate loss" argument of

SWC. Furthermore, since V and its binary representation $B_0, B_1, ..., B_{\Lambda-1}$ are one-to-one correspondence, we have

$$\begin{aligned} H(V|S) &= H(B_0, B_1, ..., B_{\Lambda-1}|S) \\ &= H(B_0|S) + H(B_1|S, B_0) + ... + H(B_{\Lambda-1}|S, B_0, ...B_{\Lambda-2}). \end{aligned}$$

Note that each term in the L.H.S. is the assigned rate for a stage. Hence, the rate required to compress V in one shot is equal to the total rate required to compress V one bit level at a time; we have no performance loss in splitting the coding scheme into stages!

2. LDPC Code Based Slepian-Wolf Coding

The low-density parity-check (LDPC) code [44] is a very good choice in implementing SWC. First, the LDPC code has very good performance. Second, it allows flexible code designs to adapt any kind of channel [22, 23, 80, 82]. The second benefit is especially appealing since the hypothetical channel between V and S can be weird in the sense of conventional channel coding.

A LDPC code is a linear block code. As the name suggested, the parity check matrix is sparse such that the number of non-zero elements in the parity matrix is relatively small. A LDPC code is best represented using a Tanner graph [92]. As an example, the Tanner graph of a binary (6,2)-LDPC code is shown in Fig. 18. The circles in the left are called the variable nodes and the squares on the right are called the check nodes. Each check node corresponds to a parity check equation of the LDPC code. parity checks equal to 0. The number of branch enumerated from a variable/check node is called the degree of that variable/check node. Note that each branch in a Tanner graph corresponds to a non-zero elements in the parity check matrix. Hence, the "low-density" property of LDPC codes translates to small average

degrees of the variable and check nodes. If all variable nodes have the same degree and so are all the check nodes, then the LDPC code is called regular. Otherwise, the LDPC code is irregular.

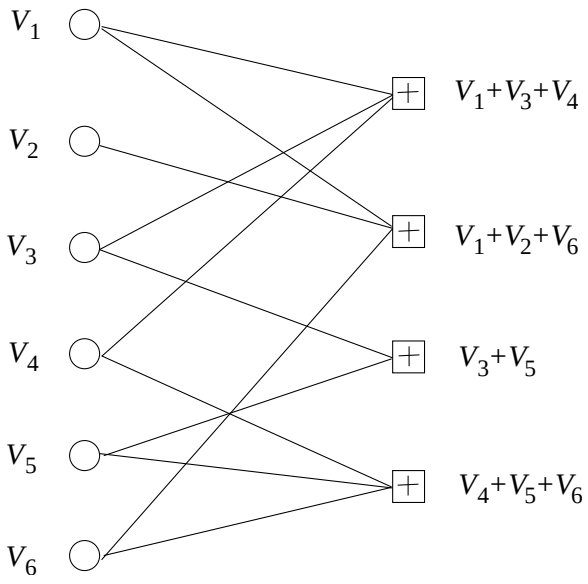

Fig. 18. The Tanner graph of a binary (6,2)-LDPC code.

To perform Slepian-Wolf encoding, the encoder computes and output the values of all check nodes, which are equivalent to the syndrome bits W^{n-k}. Given the side information S^n, the Slepian-Wolf decoder should reconstruct V^n as the best estimate out of all code vectors with syndrome W^{n-k}. MAP decoder (3.1) is optimum but is not realistic to implement for large code length n. Alternatively, a very good estimate $\hat{V}^n$ can be obtained using message-passing algorithm [81] as in conventional LDPC decoding.

As the name suggested, messages are exchanged between two ends of each branch in a message-passing algorithm. The message going into or out of a variable node possesses the "belief" of the value of that variable node. For binary LDPC codes, these

messages are typically in the form of log-likelihood ratios (i.e., $\log \frac{p(\text{observation}|V_i=1)}{p(\text{observation}|V_i=-1)}$ for the messages passing into or out of the variable node V_i). Upon receiving the messages, both variable and check nodes update the messages by combining the beliefs of the messages, and send the new messages to the other ends. To avoid the belief in a message is doubly counted, the message originated from the same branch is not included in the update (see Fig. 19).

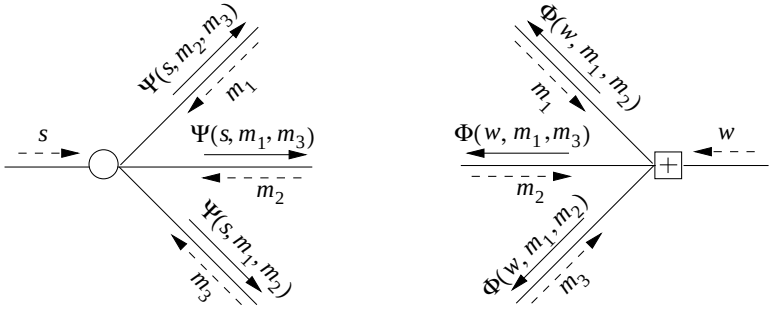

Fig. 19. Message updates of a variable node and a check node.

Recall that V, W, and S represent the value of a variable node, the value of a check node, and side information, respectively. Denote $\mathcal{V}, \mathcal{W}$, and $\mathcal{S}$ as their corresponding alphabets. Use $\mathfrak{m}$ to represent the value of a message and $\mathcal{M}$ to represent its alphabet. For a variable node i, denote the initial message mapping as $\Psi_i^{(0)} : \mathcal{S} \to \mathcal{M}$, the variable node message mapping as $\Psi_i : \mathcal{S} \times \mathcal{M}^{d_i-1} \to \mathcal{M}$, and the final message mapping as $\Psi_i^{(f)} : \mathcal{S} \times \mathcal{M}^{d_i} \to \mathcal{M}$, where d_i is the degree of the variable node i and the final message mapping combine all received messages to facilitate estimation of the actual values of the variable node i. Similarly, for a check node j, denote the check node message mapping as $\Phi_j : \mathcal{W} \times \mathcal{M}^{d_j-1} \to \mathcal{M}$, where d_j is the degree of the check node j. It is understood that the $d_i - 1$ ($d_j - 1$) input messages are those connected to the variable (check) node excluding to the message coming from the same branch

as the output message. Now, the message-passing algorithm can be more precisely summarized as follows:

1. Initialization: for every variable node i, generate message using initial message mapping $\Psi_i^{(0)}$ and pass it to every connected check node.

2. Loop:

   - Check node update: for every check node j and for every branch in that check node, update message using check node message mapping Φ_j and pass it back to the connected variable node.
   - Variable node update: for every variable node i and for every branch in that variable node, update message using variable node message mapping Ψ_i and pass it back to the connected check node.
   - Exit conditions: for every variable node i, use the final message mapping $\Psi_i^{(f)}$ to estimate the value of the variable node i. Exit if 1). the estimated variable nodes possess the desired syndrome, or 2). the maximum number of iterations is reached.

It is generally impossible to combine the beliefs of the messages exactly. However, if we assume all received message are independent of the others, then the mappings Ψ_i, $\Psi_i^{(f)}$, and Φ_j have relatively simple forms that [57]

$$\Psi_i(s, \mathfrak{m}_1, \mathfrak{m}_2, ..., \mathfrak{m}_{d_i-1}) = \log \frac{p(s|V_i = 1)}{p(s|V_i = -1)} + \sum_{i'=1}^{d_i-1} \mathfrak{m}_{i'} \tag{3.2}$$

$$\Psi_i^{(f)}(s, \mathfrak{m}_1, \mathfrak{m}_2, ..., \mathfrak{m}_{d_i}) = \log \frac{p(s|V_i = 1)}{p(s|V_i = -1)} + \sum_{i'=1}^{d_i} \mathfrak{m}_{i'}, \tag{3.3}$$

and

$$\Phi_j(w, \mathfrak{m}_1, \mathfrak{m}_2, ..., \mathfrak{m}_{d_j-1}) = 2 \operatorname{atanh} \left(w \prod_{i=1}^{d_j-1} \tanh \left(\frac{\mathfrak{m}_i}{2} \right) \right), \tag{3.4}$$

when the messages are in the form of log-likelihood ratios that

$$\Psi_i^{(0)}(s) = \log \frac{p(s|V_i = 1)}{p(s|V_i = -1)}.$$

The resulting message-passing algorithm is commonly known as the belief propagation algorithm [70] or the sum-product decoding algorithm [54]. When the information of a message passes back to itself, the assumption that the received messages are independent will obviously fail. This happens if there exists cycles in the Tanner graph and the number of iterations is larger than or equal to half of the length of the shortest cycle. For long block length n and small average node degree (low-density), the average length of cycles is large and the belief-propagation algorithm has good performance.

For LDPC coding in conventional channel coding, the decoding error probability is independent of the transmitting codeword provided that the channel satisfies certain symmetry condition. Hence, we can assume any codeword to be sent when we analyze the LDPC code performance. In specific, by assuming all-one codeword is sent and by tracking the density distribution of the average beliefs of the variable nodes, we could estimate the probability of decoding error after any number of iterations in theory. However, this cannot be easily done for a specific LDPC code since each variable/check node can have different degree. Nonetheless, if we consider an ensemble of codes which bear the same degree profile in the sense that the fraction of nodes with any particular degree is the same, then the problem become tractable and this technique is commonly known as density evolution. Density evolution can be employed for LDPC code design. The basic idea is to adjust the degree profile interactively such that the decoding error probability predicted by density evolution is smallest.

It is desirable to translate this design technique to SWC. However, we have to ensure that the SWC performance should be independent of the input codeword

as in conventional channel coding; otherwise, it is practically impossible to analyze the SWC performance. Our major result in this section is that under a symmetry condition dubbed *dual symmetry*, the SWC performance is independent of the input codeword. Moreover, if we assume all-one codeword is transmitted, Slepian-Wolf decoding is exactly equivalent to LDPC decoding since w in (3.4) is now always equal to 1. This concludes that the performance of LDPC coding translates to that of the SWC precisely. This means that if a LDPC code can perform well over the hypothetical channel $V \to S$, then the resulting SWC performs just as well. Hence, we can simply use the conventional density evolution based design for SWC without any modification!

a. Symmetry Conditions

Definition 3: A binary input channel $V \to S$ is called *sign-symmetric* if $p(s|V = 1) = p(-s|V = -1)$.

Remark 4: Sign symmetry is first addressed in [44] and is referred to as *output symmetry* in [81]. Note that sign symmetry is different from the usual notion of symmetry for discrete channels [30, pp. 189-190].

Definition 4: We call a message-passing decoding algorithm for SWC *symmetric* if it satisfies the following conditions [81]:

[Variable node symmetry]

$\Psi_v(-s, -m_1, ..., -m_{d_v-1}) = -\Psi_v(s, m_1, ..., m_{d_v-1})$

and $\Psi_v^{(0)}(-s) = -\Psi_v^{(0)}(s)$

for $s \in \mathcal{S}$ and $m_k \in \mathcal{M}$, $k = 1, ..., d_v - 1$.

[Check node symmetry] $\Psi_c(b_0, b_1 m_1, ..., b_{d_c-1} m_{d_c-1})$

$= \Psi_c(1, m_1, ..., m_{d_c-1}) \left(\prod_{i=0}^{d_c-1} b_i\right)$

for any ± 1 sequence $b_0, ..., b_{d_c-1}$ and $m_k \in \mathcal{M}$, $k = 1, ..., d_c - 1$.

Note that the belief propagation algorithm is symmetric if the hypothetical channel $V \to S$ is sign-symmetric.

As mentioned previously, the performance analysis of SWC will only be tractable if the probability of error is independent of the input V^n. We now show that if the message-passing decoding algorithm is symmetric and the hypothetical channel $V \to S$ is sign-symmetric then the above assumption is valid.

Lemma 1: Denote the error probability of Slepian-Wolf decoding after i iterations with an input v^n as $P_e^{(i)}(v^n)$. If the hypothetical channel $V \to S$ is sign-symmetric and the message-passing decoding algorithm is symmetric, then $P_e^{(i)}(v^n)$ is independent of v^n.

Proof. Our proof follows closely the proof of Lemma 1 in [81]. Assume v^n and s^n are realizations of the correlated sources, where v^n is input to the variable nodes of the LDPC and the check node values w^{n-k} are computed and transmitted to the decoder. Let z^n be a length-n vector with component $z_i = s_i v_i$. Since $V \to S$ is sign-symmetric, it is easy to verify that $p(z_i|V^n = 1^n) = p(s_i|V^n = v^n)$, where 1^n is an all-one sequence with length n. In other words, the probability of receiving s^n given v^n being transmitted is the same as that of receiving z^n given all-one sequence being transmitted.

Let v_i and c_j denote an arbitrary variable node and one of its neighboring check nodes, respectively. For any received channel output s^n and syndromes w^{n-k}, let $\mathfrak{m}_{ij}^{(l)}(s^n, w^{n-k})$ and $\mathfrak{m}_{ji}^{(l)}(s^n, w^{n-k})$ denote the message sent from v_i to c_j and the message sent from c_j to v_i in iteration l. At $l = 0$, we have $\mathfrak{m}_{ij}^{(0)}(s^n, w^{n-k}) = v_i \mathfrak{m}_{ij}^{(0)}(z^n, 1^{n-k})$ from the variable node symmetry (Definition 4).

Assume that we have $\mathfrak{m}_{ij}^{(l)}(s^n, w^{n-k}) = v_i \mathfrak{m}_{ij}^{(l)}(z^n, 1^{n-k})$ in iteration l. Since

w^{n-k} is syndrome of v^n, i.e., $w_j \prod_{k:\exists e=(v_k,c_j)} v_k = 1$ and thus $\mathfrak{m}_{ji}^{(l+1)}(s^n, w^{n-k}) = v_i \mathfrak{m}_{ji}^{(l+1)}(z^n, 1^{n-k})$ from the check node symmetric condition (Definition 4). From this, we can conclude $\mathfrak{m}_{ij}^{(l+1)}(s^n, w^{n-k}) = v_i \mathfrak{m}_{ij}^{(l+1)}(z^n, 1^{n-k})$ using variable node symmetry again in iteration $l+1$. Hence by induction, we can show that any message from the check nodes and the variable nodes given s^n being received is equal to the product of v_i and the correspond message given z^n is received. Therefore, both cases cause the same number of errors and this completes the proof. □

For the multilevel SWC, all the previous decoded bit planes can also be considered as side information. Hence, in this case, our hypothetical channel, which includes decoded bits from other bit planes, is not even a single real number. The notion of sign symmetry is too restrictive for our purpose. Thus we introduce a more general type of symmetry as follows.

Definition 5: We call a binary input channel $V \to S$ dual-symmetric if there exists a mapping $g : \mathcal{S} \to \mathcal{S}$ such that for any s,

$$p(s|V=1) = p(g(s)|V=-1) \quad \text{and} \quad g(g(s)) = s. \tag{3.5}$$

Before proceeding to our main result, we will present some properties of dual symmetry.

Lemma 2: Sign symmetry implies dual symmetry.

Proof. Pick $g(s) = -s$. □

Lemma 3: Strong symmetry[2] [30, pp. 189-190] implies dual symmetry.

[2]This condition is usually simply referred to as *symmetry*, but we use the term *strong symmetry* to avoid confusion.

Proof. If a binary input channel $V \to S$ is strongly symmetric, then the probability of S given $V = 1$ is a permutation of the probability of S given $V = -1$. Therefore, there exists a mapping g such that $p(s|V = 1) = p(g(s)|V = -1)$, $\forall s$. Moreover, strong symmetry requires

$$p(s|V = -1) + p(s|V = 1) = p(s'|V = -1) + p(s'|V = 1),$$

for all s and s'. In particular,

$$\begin{aligned} & p(s|V = -1) + p(s|V = 1) = p(g(s)|V = -1) + p(g(s)|V = 1) \\ \Rightarrow & p(s|V = -1) = p(g(s)|V = 1). \end{aligned}$$

This implies $g(g(s)) = s$. □

Remark 5: Although strong symmetry and sign symmetry both imply dual symmetry, it is easy to find a dual-symmetric channel that satisfies neither of the former symmetries. Hence dual symmetry is the weakest among the three.

Remark 6: Weak symmetry [30, pp. 190] does not imply dual symmetry. For example[3], consider a binary input channel with $\mathcal{S} = \{s_1, s_2, s_3\}$ such that $p(s_1|V = -1) = 0.5$, $p(s_2|V = -1) = 0.3$, $p(s_3|V = -1) = 0.2$, $p(s_1|V = 1) = 0.3$, $p(s_2|V = 1) = 0.2$, and $p(s_3|V = 1) = 0.5$. This channel is weakly symmetric, we have that $g(s_1) = s_2$, $g(s_2) = s_3$, and $g(s_3) = s_1$. Since it holds obviously that $g(g(s_1)) = g(s_2) = s_3 \neq s_1$, the channel is not dual-symmetric.

Lemma 4: If the binary input channel $V \to S$ is dual-symmetric, the input distribution that achieves the capacity is uniform, i.e., $p(V = 1) = p(V = -1) = 0.5$.

[3]This example is originated from an anonymous reviewer.

Proof. Let $p(V=1)=q$, the optimum q should maximize the mutual information $I(V;S)=H(S)-H(S|V)$. From (3.5), it can be readily verified that $H(S|V)$ is independent of q. Therefore, the optimum q should maximize $H(S)$. Expand

$$\begin{aligned} H(S) &= -\sum_s p(s)\log p(s) \\ &= -\sum_s (p(s|V=1)q + p(s|V=-1)(1-q)) \\ &\quad \log\left(p(s|V=1)q + p(s|V=-1)(1-q)\right), \end{aligned}$$

and note that $H(S)$ is a concave function with respect to q because each summand is concave and concavity is preserved by summation. By its concavity, $H(S)$ achieves the global maximum when $\frac{\partial H}{\partial q}=0$ is satisfied. Since

$$\begin{aligned} \frac{\partial H}{\partial q} =& -\sum_s (p(s|V=1)-p(s|V=-1))\log\left(p(s|V=1)q + p(s|V=-1)(1-q)\right) \\ & \sum_s (p(s|V=1)-p(s|V=-1)) \\ =& -\sum_s (p(s|V=1)-p(s|V=-1))\log\left(p(s|V=1)q + p(s|V=-1)(1-q)\right), \end{aligned}$$

we have

$$\begin{aligned} \left.\frac{\partial H}{\partial q}\right|_{q=0.5} &= \sum_s (p(s|V=-1)-p(s|V=1)) \\ &\quad \log\left(p(s|V=1)+p(s|V=-1)\right) \\ &\overset{(a)}{=} \sum_s (p(g(s)|V=1)-p(g(s)|V=-1)) \\ &\quad \log\left(p(g(s)|V=-1)+p(g(s)|V=1)\right) \\ &= \sum_s (p(s|V=1)-p(s|V=-1)) \\ &\quad \log\left(p(s|V=-1)+p(s|V=1)\right) \\ &= -\left.\frac{\partial H}{\partial q}\right|_{q=0.5} = 0, \end{aligned}$$

where (a) comes from the dual-symmetric properties of $V \to S$. Thus the optimum input distribution is uniform ($p(V=1) = q = 0.5$). □

We are interested in dual symmetry because we can show that the sign-symmetric condition in Lemma 1 can be replaced by this weaker condition.

Lemma 5: If the message-passing decoding algorithm is symmetric and the hypothetical channel $V \to S$ is dual-symmetric, then there exists a sufficient statistic L for S such that $V \to L$ satisfies sign symmetry.

Proof. Define $f(s) \triangleq \log \frac{p(s|V=1)}{p(s|V=-1)}$, then $L \triangleq f(S)$ is the log-likelihood ratio (LLR) of V given S. Note that L is a sufficient statistic for S. Hence the proof is complete if we can show $V \to L$ to be sign-symmetric.

$$\begin{aligned}
p_{L|V}(l|V=1) &= \sum_{s\in\{s|f(s)=l\}} p_{S|V}(s|V=1) \\
&\overset{(a)}{=} \sum_{s\in\{s|f(s)=l\}} p_{S|V}(g(s)|V=-1) \\
&= \sum_{z\in\{z|f(g^{-1}(z))=l\}} p_{S|V}(z|V=-1) \\
&\overset{(b)}{=} \sum_{z\in\{z|f(z)=-l\}} p_{S|V}(z|V=-1) \\
&= p_{L|V}(-l|V=-1),
\end{aligned}$$

where (a) comes from the dual symmetry of $V \to S$ and (b) follows from $f(g^{-1}(z)) = f(g(z)) = \log \frac{p(g(z)|V=1)}{p(g(z)|V=-1)}$
$= \log \frac{p(z|V=-1)}{p(z|V=1)} = -f(z)$. □

Theorem 1: If the message-passing decoding algorithm is symmetric and the hypothetical channel $V \to S$ is dual-symmetric, then without loss of performance, the decoder can preprocess the channel output such that the resulting error probability of Slepian-Wolf decoding after i iterations, $P_e^{(i)}(v^n)$, is independent of the input

sequence v^n.

Proof. From Lemma 5, the message-passing decoder can treat L rather than S as the output without loss of performance. Then result follows directly from Lemma 1. □

Theorem 1 allows us to choose any input sequence for our analysis. In particular, if we select the all-one sequence as our input, then all the check nodes will have value one and the message-passing decoding algorithm degenerates to that for conventional channel decoding. Therefore, the SWC performance will be exactly equivalent to the LDPC coding performance in conventional channel coding. Moreover, all code designing tools for conventional channel coding can be used for SWC provided that the conditions in Theorem 1 are satisfied. In specific, this allows us to use density evolution [80] to analyze LDPC code based SWC performance. The above discussion is summarized in the below corollary.

Corollary 6: If the message-passing decoding algorithm is symmetric and the hypothetical channel $V \to S$ is dual-symmetric, then the performance of the resulting SWC is exactly the same as that of the LDPC coding applying on $V \to S$.

When the belief propagation decoding algorithm is employed, the preprocessing step in Theorem 1 is just equivalent to setting the initial message as the LLR L. In this case, stronger statements can be made about density evolution if the hypothetical channel satisfies dual symmetry. First, we start with a definition.

Definition 6 (Symmetric distribution [81]): A distribution is symmetric if its density $p(v)$ satisfies $p(v) = e^v p(-v), \forall v$.

It is shown in [80] that if the initial message distribution is symmetric and the belief propagation decoding algorithm is used, then density evolution converges to a fixed point. Moreover, an upper bound for the code threshold, which describes the

minimum correlation between the source and side information to have no SWC error, can be derived from the stability condition analysis [80]. We will show below that if the hypothetical channel is dual-symmetric and the LLR of the source given the side information L is selected as the initial message, then the initial message distribution is symmetric.

Theorem 2: For a binary SWC scheme with input V and side information S, if the hypothetical channel $V \to S$ is dual-symmetric, then the initial message in log-likelihood ratio given all-one input sequence is symmetric.

Proof. Recall that L from the proof of Lemma 1 is a sufficient statistic for S and hence can be used as the initial message without performance loss. Assuming that all-one sequence is transmitted, we have

$$\begin{aligned}
p_{L|V}(l|V=1) &= \sum_{s\in\{s|f(s)=l\}} p_{S|V}(s|V=1) \\
&\overset{(a)}{=} \sum_{s\in\{s|f(s)=l\}} e^l p_{S|V}(s|V=-1) \\
&\overset{(b)}{=} e^l \sum_{s\in\{s|f(s)=l\}} p_{S|V}(g^{-1}(s)|V=1) \\
&= e^l \sum_{z\in\{z|f(g(z))=l\}} p_{S|V}(z|V=1), \\
&\overset{(c)}{=} e^l \sum_{z\in\{z|f(z)=-l\}} p_{S|V}(z|V=1), \\
&= e^l p_{L|V}(-l|V=1),
\end{aligned}$$

where (a) is due to the definition of l, (b) is due to (3.5), and (c) is obtained from $f(g(z)) = \log \frac{p(g(z)|V=1)}{p(g(z)|V=-1)} = \log \frac{p(z|V=-1)}{p(z|V=1)} = -f(z)$. □

From Theorem 1, it follows that we can analyze the SWC performance assuming all-one input sequence. Then from Theorem 2 and [80], density evolution assuming all-one input sequence will converge to a fixed point. For completeness, we present

the upper bound of the code threshold derived from the stability condition as follows. Given that the length of the LDPC code and the number of iterations tend to infinity, the necessary condition for no SWC error is [80]

$$\lambda'(0)\rho'(1) < \left(\int_{\mathbb{R}} p_{L|V}(l|1) e^{-\frac{l}{2}} dl \right)^{-1},$$

where $\lambda(\cdot)$ and $\rho(\cdot)$ are the left and right degree distributions of the LDPC code, respectively.

B. Wyner-Ziv Coding: Non-Zero Distortion Case

In this section, we will move to the more general case when the distortion of the reconstructed source can be nonzero. Hence, there will be no restriction in the source alphabet and the source can be continuous in general. We will emphasize this by denoting the source as X while keeping the same notation S for the side information.

1. General Approaches

a. Nested Lattice Quantization

Recall the nested coding scheme described in Chapter II. In practice, it is hard to implement nested code without any structure. Therefore, it is common to constraint all the codewords of both subcodes and the original code as lattice point and this results in nested lattice [112, 113]. For instance, Fig. 20 shows examples of 1-D and 2-D nested lattices [113] based on similar sublattices [27]. The fine lattice code corresponds to the codewords represented by all the numbers in Fig. 20, while a subcode or a coarse lattice code includes only the codewords indexed by one particular number. The coarse lattice is nested in the fine lattice in the sense that each point of the coarse lattice is also a point of the fine lattice but not vice versa. To encode, $\mathbf{x}$

is first quantized with respect to the fine source code, resulting in quantization loss. However, only the index identifying the coarse lattice that contains the quantized $\mathbf{x}$ is coded to save rate. Note that this index is essentially the bin index in SWC with the bin corresponding to the union of the fine lattice Vornoni regions of elements of the coarse lattice. Knowing the coarse lattice that $\mathbf{x}$ lies closest to and the side information s^n, the decoder estimates $\hat{\mathbf{x}}$ appropriately.

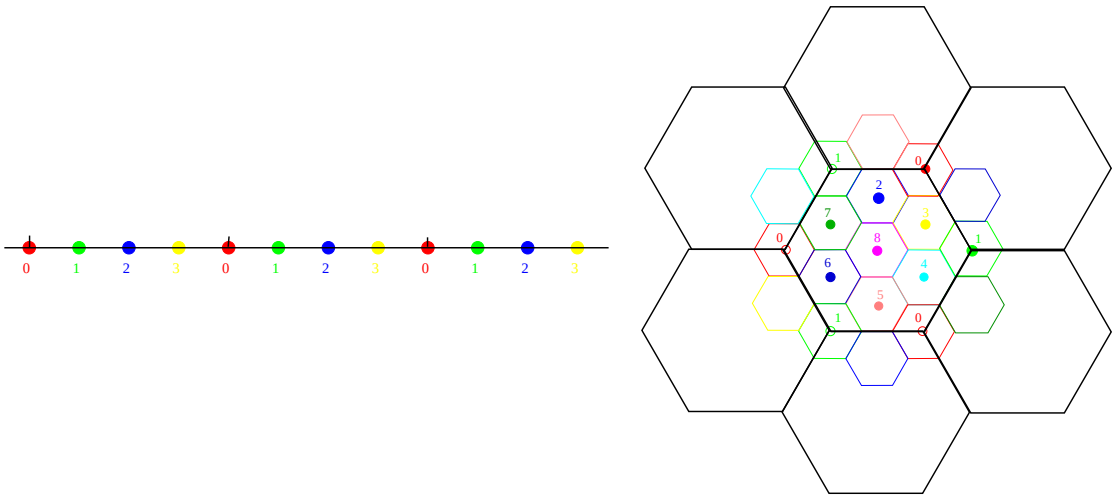

Fig. 20. 1-D and 2-D nested lattices based on similar sublattices.

b. Slepian-Wolf Coded Quantization

Although it is proven in [113] that nested lattice quantization approaches the Wyner-Ziv limit for infinite dimensional source and channel codes, high dimensional nested lattice is difficult to implement whereas low dimensional nested lattice quantization has rather poor performance. For example, Fig. 21 shows the operational rate-distortion function for 1-D nested lattice quantization, which exhibits a huge gap from the Wyner-Ziv limit at high rate. An immediate attempt to improve the performance of nested lattice quantization is to compress the bin index with entropy encoding. However, as we shall see, conventional entropy coding is not sufficient.

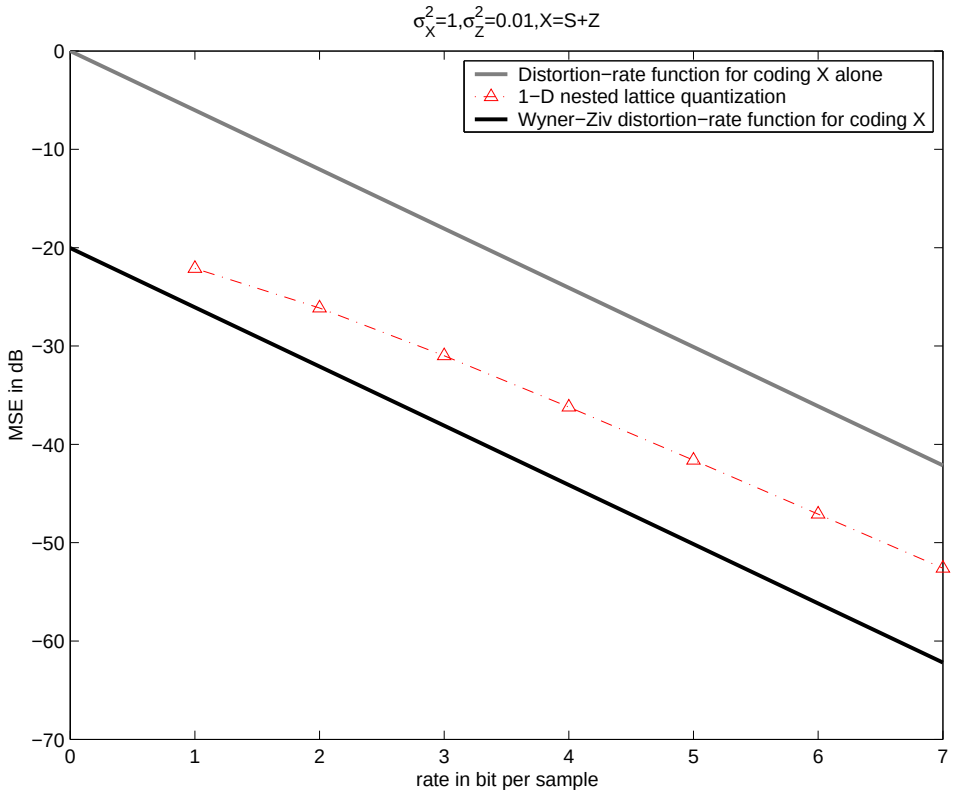

Fig. 21. Operational rate-distortion function for 1-D nested lattice quantization

One interesting observation for low-dimensional nested lattice quantization is that the bin index V and the side information S is highly correlated. This means $I(V;S) > 0$ and $H(V)$ is strictly larger than $H(V|S)$. Hence, there will be performance loss if we attempt to compress V ignoring the side information S. Note that conventional entropy coding (e.g., context based arithmetic coding [83]) does not work since S is only available to the decoder. However, since V is discrete and S is available to the decoder, we can compress V losslessly using SWC. And better still, it is possible to achieve the theoretical limit $H(V|S)$ since SWC has no rate loss. To summarize, we propose a WZC paradigm dubbed as *Slepian-Wolf coded quantization (SWCQ)* as nested lattice quantization followed by SWC.

For practical lossless source coding, conventional technique (e.g., Huffman cod-

ing, arithmetic coding, Lempel-Ziv coding [114], PPM [24] and CTW [98]) have dominated so far. However, if one regards lossless source coding as a special case of Slepian-Wolf coding without side information at the decoder, then channel coding techniques can also used for source coding based on syndromes [64]. In this light, the SWC component in SWCQ can be viewed as the counterpart of entropy coding in classic source coding. Although the idea of using channel codes for source coding dates back to the Shannon-MacMillan theorem [85, 60] and theoretical results appeared in [97, 6], practical turbo/LDPC code based noiseless data compression scheme did not appear until very recently [46, 17].

Starting from syndrome based approaches for entropy coding, one can easily make the schematic connection between entropy-coded quantization for classic source coding and SWC-NQ for Wyner-Ziv coding, as syndrome based approaches can also be employed for SWC (or source coding with side information at the decoder) in the latter case. Performance-wise, our work in [106, 56, 108] reveals that the performance gap of high-rate Wyner-Ziv coding (with ideal Slepian-Wolf coding) to $D_{WZ}(R)$ is exactly the same as that of high-rate classic source coding (with ideal entropy coding) to the distortion-rate function $D_X(R)$. This interesting and important finding is highlighted in Table V.

Table V. High-rate classic source coding vs. high-rate Wyner-Ziv coding.

Classic source coding		WZC	
Coding scheme	Gap to $D_X(R)$	Coding scheme	Gap to $D_{WZ}(R)$
ECSQ [48]	1.53 dB	SWC-NSQ	1.53 dB
ECLQ (2-D) [28]	1.36 dB	SWC-NQ (2-D) [56]	1.36 dB
ECTCQ [93]	0.2 dB	SWC-TCQ [108]	0.2 dB

2. 1-D Slepian-Wolf Coded Quantization

In this section, we will illustrate a WZC scheme based on 1-D nested lattice quantization and LDPC code. For simplicity, we assume the source X and the side information S is related by $X = S + Z$, where both $S \sim N(\mu_S, \sigma_S^2)$ and $Z \sim N(\mu_Z, \sigma_Z^2)$ are i.i.d. and independent of each other.

a. Basic Setup

Fig. 22 shows a nested scalar quantizer, which consists of a coarse coset channel code with minimum distance $d_{\min}$ nested in a fine uniform scalar quantizer with stepsize q. Obviously $d_{\min}$ is an integer multiple of q and we call $N = \frac{d_{\min}}{q}$ the nesting ratio. To encode, X is quantized by the fine source code (uniform quantizer). However, only the index J $(0 \le J \le N-1)$ of the coset channel code that the quantized X belongs to is coded by SWC.

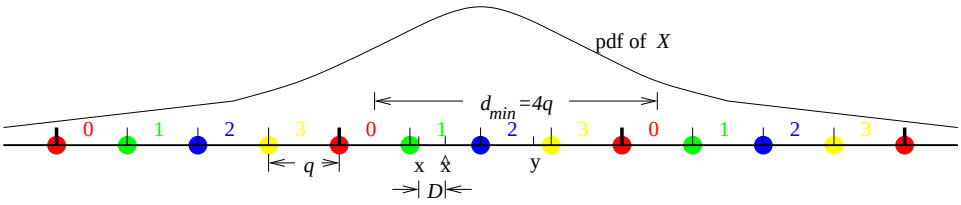

Fig. 22. A nested scalar quantizer with nesting ratio $N = 4$.

To employ (n, k)-LDPC codes for SWC, the coset index J is first grouped into a block J^n. Since binary LDPC codes are used, multilevel SWC described in Section IIIA.1.c will be employed. Hence, we split J^n into bit planes before SWC. Assume N is a power of 2 and $\Lambda = \log_2 N$ is the number of bit planes, and define

$$B_k(x) \equiv \left\lfloor \frac{2^{k+1}x}{d_{\min}} \right\rfloor \mod 2 \quad \text{and} \quad b_k(x) \equiv \left\lfloor \frac{x}{2^k q} \right\rfloor \mod 2,$$

then $B_k(X^n)$ represents the k^{th} bit plane of J^n starting from the most significant bit

(MSB) plane, and $b_k(X^n)$ represents the k^{th} bit plane of J^n starting from the least significant bit (LSB) plane. While these bit planes can be transmitted in many different orders, we focus on the two most natural choices: sequentially coding bit planes, $B_0(X^n)$, $B_1(X^n)$, ..., $B_{\Lambda-1}(X^n)$, starting from the MSB plane $B_0(X^n)$ (*top-down approach*), and sequentially coding bit planes, $b_0(X^n)$, $b_1(X^n)$,..., $b_{\Lambda-1}(X^n)$, starting from the LSB plane $b_0(X^n)$ (*bottom-up approach*). The proposed WZC scheme for the top-down approach is illustrated in Fig. 23.

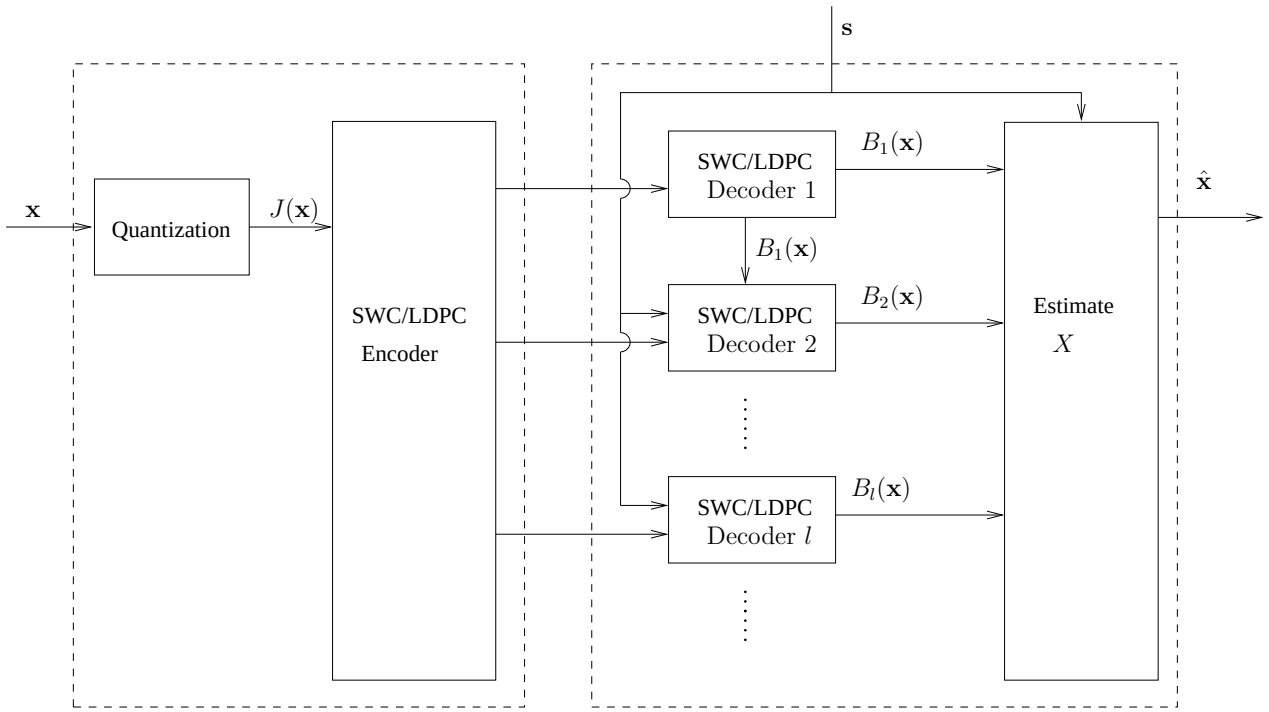

Fig. 23. The proposed Wyner-Ziv scheme with SWC.

Note that multistage decoding is used and the values of all previous received bit planes are considered as a part of the hypothetical channel output in each Slepian-Wolf decoder, i.e., we have the hypothetical channels $B_k(X) \rightarrow (S, \hat{B}_0(X), ..., \hat{B}_{k-1}(X))$ for the top-down approach and $b_k(X) \rightarrow (S, \hat{b}_0(X), ..., \hat{b}_{k-1}(X))$ for the bottom-up approach at the k^{th} bit plane. However, for sufficiently small error probability ($\sim 10^{-5}$

in our experiment), the channels $B_k(X) \rightarrow (S, B_0(X), ..., B_{k-1}(X))$ and $b_k(X) \rightarrow (S, b_0(X), ..., b_{k-1}(X))$ essentially have the same statistics as the previous pair. Hence, these two approximated channels will be used instead in our analysis. Furthermore, given this approximation, it is easy to verify using the chain rule that the performances of both top-down and bottom-up approaches are the same.

Given the coset index J and the side information S, the decoder recovers X using the optimum non-linear estimator. If mean square error is used as the distortion measure, the optimum estimate for a particular sample x_i is the centroid of X given s_i and the received bits. For example, when Λ bits $B_1(x_i), B_2(x_i), ..., B_\Lambda(x_i)$ are received by the decoder, the centroid should be computed as $E[X|s_i]$ over $\{x : B_k(x) = B_k(x_i), k = 1, 2, ..., \Lambda\}$ in which x_i can only exist according to the information obtained from the Λ received bits. In other words,

$$\hat{x_i}(s_i, B_1(x_i), B_2(x_i), ..., B_\Lambda(x_i)) = \frac{1}{\sqrt{2\pi\sigma_Z^2}} \int_{\{x:B_k(x)=B_k(x_i),i=1,2,...,\Lambda\}} e^{-\frac{(x-s_i)^2}{2\sigma_Z^2}} dx. \quad (3.6)$$

b. Design Issues

Dual Symmetry

We will now show that the hypothetical channels for all bit levels are dual-symmetric for both top-down and bottom-up approaches. Hence by Theorem 1, design techniques based on density evolution can be employed. We will only focus on the top-down approach since the proof for the bottom-up approach is similar.

Assume $\mu_X = \mu_S = \mu_Z = 0$. In order to match the notation in previous section, we relabel 0 as 1 and 1 as -1. Denote $[B_j(X), B_{j+1}(X), ..., B_k(X)] = B_j^k(X)$. Then, it is easy to verify that

$$p(B_k(X) = 1) = p(B_k(X) = -1) = \frac{1}{2} \quad (3.7)$$

and

$$p(B_j^k(X) = v_j^k, S = s) = p(B_j^k(X) = -v_j^k, S = -s) \tag{3.8}$$

for any j and k. Consider the hypothetical channel $B_k(X) \to (B_0^{k-1}(X), S)$ at the k^{th} bit plane, then

$$\begin{aligned} & p(B_0^{k-1}(X) = v_0^{k-1}, S = s | B_k(X) = -1) \\ = & \frac{p(B_0^k(X) = [v_0^{k-1}, -1], S = s)}{p(B_k(X) = -1)} \\ \overset{(a)}{=} & \frac{p(B_0^k(X) = [-v_0^{k-1}, 1], S = -s)}{p(B_k(X) = 1)} \\ = & p(B_0^{k-1}(X) = -v_0^{k-1}, S = -s | B_k(X) = 1), \end{aligned}$$

where (a) is due to (3.7) and (3.8). Define $g_k : \{-1, 1\}^{k-1} \times \mathbb{R} \to \{-1, 1\}^{k-1} \times \mathbb{R}$ with $g_k(v_0^{k-1}, s) = g_k(-v_0^{k-1}, -s)$. Then $B_k(X) \to (B_0^{k-1}(X), S)$ is dual-symmetric and $g_k(\cdot, \cdot)$ is the mapping required in Definition 5.

Assume $\mu_X = \mu_S = \mu \neq 0$ but $\mu_Z = 0$. The hypothetical channel considered by each Slepian-Wolf decoder is no longer dual-symmetric. Hence, designs based on density evolution will not perform well in general. However, this can be solved easily by adjusting the quantization function. Specifically, define

$$B'_k(x) = \begin{cases} 1 & 0 \equiv \left\lfloor \frac{2^k(x-\mu)}{d_{\min}} \right\rfloor \mod 2, \\ -1 & \text{otherwise}, \end{cases} \tag{3.9}$$

to be transmitted, and replace $B_k(X^n)$ by $B'_k(X^n)$ in the described WZC scheme. Define $g'_k(v_0^{k-1}, s) = g'_k(-v_0^{k-1}, 2\mu - s)$; then it is easy to verify that $g'_k(g'_k(v_0^{k-1}, s)) = (v_0^{k-1}, s)$ and $p_{(B'^{k-1}_0(X), S)|B'_k(X)}(v_0^{k-1}, s|1) = p_{(B'^{k-1}_0(X), S)|B'_k(X)}(g'_k(v_0^{k-1}, s)| - 1)$. Hence, the hypothetical channel $B'_k(X) \to (B'^{k-1}_0(X), S)$ is dual-symmetric. Note that the above discussion actually suggests an intuitive scheme: one should shift the

source to zero mean before quantization.

Table VI. Rates distributed over different Slepian-Wolf coders for top-down and bottom-up approaches when $d_{\min} = 12\sigma_Z$ and $\Lambda = 5$.

k	0	1	2	3	4
$H(B_k(X)\|B_0^{k-1}, S)$	0.43	0.43	0.72	0.91	0.97
$H(b_k(X)\|b_0^{k-1}, S)$	1.00	1.00	0.98	0.47	0.01

Top-down versus bottom-up approaches

Although the overall rate for the two approaches are the same, they distribute the rates differently among Slepian-Wolf coders. For example, Table VI shows the optimum compression rate for each Slepian-Wolf coder when $d_{\min} = 12\sigma_Z$ and $\Lambda = 5$ for both approaches. The higher the rate means the poorer the hypothetical channel (the weaker the correlation) and vice versa. The required rate of the LDPC code should be one minus the compression rate. If the code rate reaches 0 or 1, then there is no use (or no need) to employ SWC, which is beneficial since each LDPC code introduces certain rate loss. Hence from Table VI, we expect that the bottom-up approach performs better than the top-down approach. Our experiments showed that the above statement is generally true. One intuitive reason is that the lower bit planes are more uncertain than the higher ones. Without the knowledge of the higher bit planes, the conditional entropies of the lower bit planes given side information, i.e., the compression rates of the first few bit planes in the bottom-up approach, are very close to one, whereas for the top-down approach, the compression rates for these bit planes also approach one but not as fast as in the bottom-up approach because the values of the upper bit planes, an extra side information, are now available.

On the other hand, the advantage of the top-down approach is its progressive

nature. Its decoder can have a good estimate of X even if only the first few bit planes are received. This is not true for the bottom-up approach since these bit planes have the least impact to the overall distortion and hence almost no distortion reduction is expected.

Choice of Quantization Step Size $d_{\min}$

Table VII. The table shows the conditional entropy of $B_k(X)$ given previous decoded bits and the side information S, the overall rate $R = H(B^{\Lambda}(X)|S)$ and the corresponding squared error distortion $D = E[(X - \hat{X})^2]$ for different $d_{\min}$. We assume the jointly Gaussian model of $X = S + Z$ where S and Z are independent Gaussian random variables with $\sigma_Z^2 = 0.01$ and $\sigma_S^2 = 1$, respectively.

	$H(B_k(X)\|B^{k-1}(X), S)$							R	D
k	1	2	3	4	5	6	7	–	–
$d_{\min} = 3\sigma_Z$,$\Lambda = 3$	0.99	0.99	1.00	–	–	–	–	2.98	8.95e-3
$d_{\min} = 6\sigma_Z$,$\Lambda = 4$	0.79	0.77	0.92	0.98	–	–	–	3.46	8.44e-4
$d_{\min} = 12\sigma_Z$,$\Lambda = 5$	0.43	0.43	0.72	0.91	0.98	–	–	3.46	1.16e-4
$d_{\min} = 24\sigma_Z$,$\Lambda = 6$	0.22	0.21	0.43	0.72	0.91	0.98	–	3.46	1.16e-4
$d_{\min} = 48\sigma_Z$,$\Lambda = 7$	0.12	0.10	0.22	0.43	0.72	0.91	0.98	3.47	1.16e-4

If we exclude SWC in our scheme, the total rate R is just equal to the number of received bit planes Λ. Thus the optimum scheme requires minimizing D over $d_{\min}$ for each fixed Λ. However, it turns out that each Λ has a different optimum $d_{\min}$ and D increases rather rapidly from the minimum as $d_{\min}$ deviates from its optimal value, thus it would be impossible to find a single $d_{\min}$ that allows the scheme to approach the optimum performance for all refinement stages [37]. However, when NSQ is followed by SWC, the performance of our scheme is much less sensitive to the choice of $d_{\min}$.

Note that the main difference now is that R depends not only on Λ but also on $d_{\min}$. Specifically, by adjusting $d_{\min}$, it might happen that more than one Λ will result in the same R. For easy exposition, we list in Table VII the conditional entropies of $B_k(X)$ given S and the previous decoded bits with different $d_{\min}$, the corresponding total rate R, and the overall distortion D when Λ bit planes are received. For every doubling of $d_{\min}$, we deliberately increase the number of decoded layers Λ by one to ensure q and hence D to be a constant since $D \approx \frac{q^2}{12}$ for $d_{\min} \gg \sigma_Z$ [37], where this fact is verified in the last column of Table VII. Moreover, it is interesting to see from the second to the last column of Table VII that the R's are about the same for these settings too, where the inertness of R can be explained as follows. Let P_e be the outage probability when S and X are relatively far apart with respect to $d_{\min}$. If $d_{\min} \gg \sigma_Z$ as described above, then $P_e \approx 0$. Denote ΔR as the rate increase if all bit planes more significant than B_1 are also transmitted. In other words, $R + \Delta R$ is the rate when q is kept fixed while $d_{\min}$ tends to infinity, i.e., that with classic uniform scalar quantization having the step size q. Denote the additionally transmitted bit planes as $B^0_{-\infty}(X)$. Then all these bit planes will be different from those obtained by quantizing $\hat{X}$ (i.e., $B^0_{-\infty}(X) \neq B^0_{-\infty}(\hat{X})$) only if S and X are far apart with respect to $d_{\min}$. Therefore,

$$\begin{aligned}
\Delta R &= H(B^0_{-\infty}(X)|S, B^\Lambda(X)) \\
&\leq (1-P_e)H(B^0_{-\infty}(X)|B^0_{-\infty}(X) = B^0_{-\infty}(\hat{X}), S, B^\Lambda(X)) + \\
&\quad P_e H(B^0_{-\infty}(X)|B^0_{-\infty}(X) \neq B^0_{-\infty}(\hat{X}), S, B^\Lambda(X)) \\
&\approx 0,
\end{aligned}$$

since $H(B^0_{-\infty}(X)|B^0_{-\infty}(X) = B^0_{-\infty}(\hat{X}), S, B^\Lambda(X)) = 0$ and $H(B^0_{-\infty}(X)|B^0_{-\infty}(X) \neq B^0_{-\infty}(\hat{X}), S, B^\Lambda(X)) \leq H(B^0_{-\infty}(X))$, which is the entropy of quantized X with step

size $d_{\min}$, is finite because $d_{\min}$ and σ_X^2 (assuming $\sigma_S^2 < \infty$ and $\sigma_Z^2 < \infty$) are finite.

From the above discussion we concludes that for a sufficiently large $d_{\min}$, both R and D, and hence the performance of the scheme will not vary with further increase of $d_{\min}$. More interestingly, one can show as a stronger result that any $d_{\min}$ is very close to be optimum in terms of minimizing the gap between the resulting (R, D)-pair and the Wyner-Ziv rate-distortion function, provided that $d_{\min}$ is large compared to σ_Z [37]. In addition, the optimum gap will be 1.53 dB asymptotically at high rate [106]. Even though our scheme uses a fixed $d_{\min}$ for all Λ, the above conclusion ensures that our scheme is practically successively refinable, i.e., it achieves the operational rate-distortion function for all refinement stages, as long as $d_{\min}$ is sufficiently large and ideal SWC is assumed.

A large $d_{\min}$ is appealing on the surface as it allows finer rate control in practice. However, each practical Slepian-Wolf code is subject to a small probability of error and hence we may want to limit the number of layers. On the other hand, $d_{\min}$ should not be too small since otherwise the overall distortion would increase significantly as shown in Table VII.

c. Experimental Results

We carry out experiments for both the top-down and bottom-up approaches. For both cases, we assume $\sigma_S^2 = 1$ and $\sigma_Z^2 = 0.01$. For the top-down approach, we set $d_{\min} = 12\sigma_Z$, which is the optimal choice for any rate approximately less than 10 bits per sample as is shown in [56]. The ideal compression rates for the corresponding first five MSB planes, which are also shown in Table VI, are approximately 0.43, 0.43, 0.72, 0.91, and 0.97 bit per sample, respectively. From the fourth bit plane on, the rate for each bit plane is very close to 1 bit and thus the corresponding bit plane becomes almost impossible to compress. Hence the Slepian-Wolf coders are only employed for

the first four MSBs. For the bottom-up approach, we adjust $d_{\min}$ such that for each Λ only two bit planes require SWC. The resulting $d_{\min}$'s are 2.3, 2.1, 1.85, 1.8, and 1.7 for $\Lambda = 3, 4, 5, 6$, and 7, respectively. We design the LDPC code for each SWC using Gaussian approximation [23], which is built upon density evolution [81]. Due to limited space, we present only the degree profiles for the top-down approach in Table VIII.

Fig. 24 and Fig. 25 show results of the top-down approach and the bottom-up approach and the same schemes without SWC are included for comparison. To validate our results, we include in the same figures the high-rate analysis obtained by [56]. Our result with practical SWC (codelength= 10^6 bits) is approximately 1.33 dB away from the Wyner-Ziv bound at low rate (0.47 bit per sample) and up to about 2.83 dB away at high rate (5.65 bits per sample) for the top-down approach. For the bottom-up approach, our performance gap is 1.66 dB at low rate (0.93 bit per sample) and up to 1.80 dB at high rate (5.00 bits per sample). Thus in high rate (e.g., 5 bits per sample), our practical SWC design loses 1.3 dB and 0.27 dB with the top-down and bottom-up approaches, respectively. However, while the bottom-up approach performs better than the top-down approach, the latter has the advantage of being progressive as explained in Section IIIB.2.b.

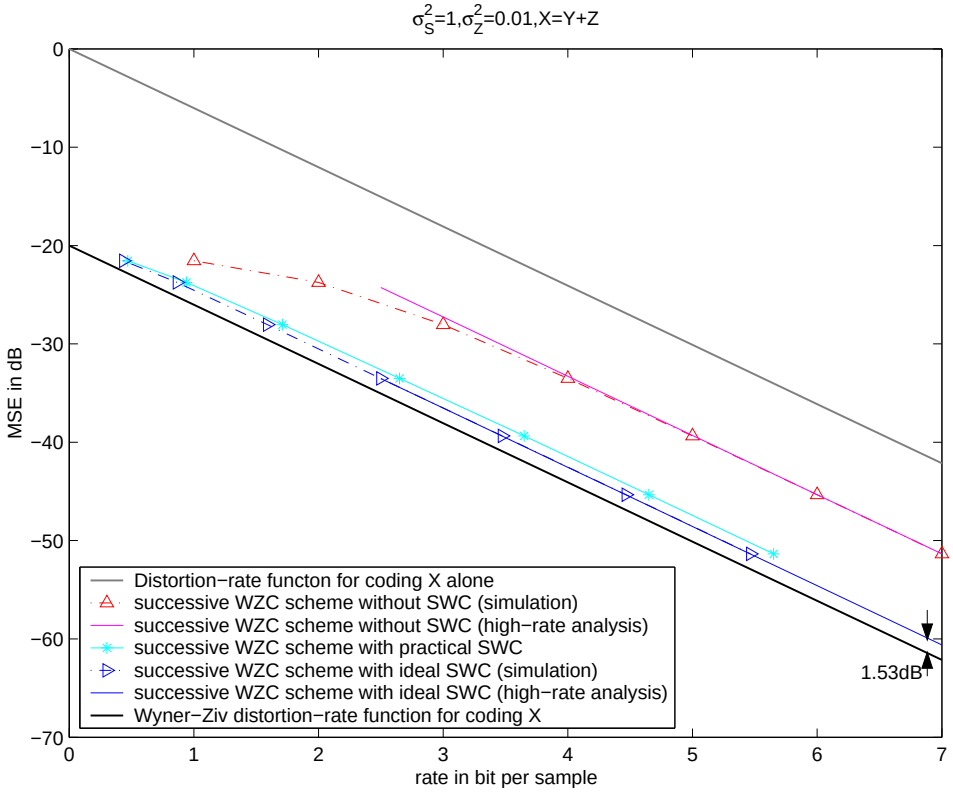

Fig. 24. Results based on nested scalar quantization with and without SWC for the top-down approach.

Table VIII. Degree profiles of the first four bit planes obtained with the top-down approach. Only the left profiles (λ) are shown since the right profiles (ρ) can be derived from the rate and λ given that ρ is concentrated on two consecutive degrees.

$B_0(X)$ and $B_1(X)$		$B_2(X)$		$B_3(X)$	
i	λ_i	i	λ_i	i	λ_i
2	0.16145	2	0.24662	2	0.37896
3	0.15690	3	0.16433	3	0.16390
4	0.00648	4	0.00001	4	0.02500
5	0.01472	6	0.14552	5	0.01662
6	0.02235	7	0.04093	6	0.12736
7	0.08156	8	0.00229	7	0.02027
8	0.10091	12	0.00699	8	0.00008
20	0.01793	14	0.09045	10	0.02903
21	0.01386	17	0.00857	11	0.01409
22	0.10531	27	0.06355	15	0.01249
23	0.04517	28	0.03661	20	0.11028
24	0.01099	30	0.02529	62	0.01723
41	0.01885	99	0.00001	66	0.07192
44	0.00414	100	0.16877	100	0.01270
99	0.01713				
100	0.22216				

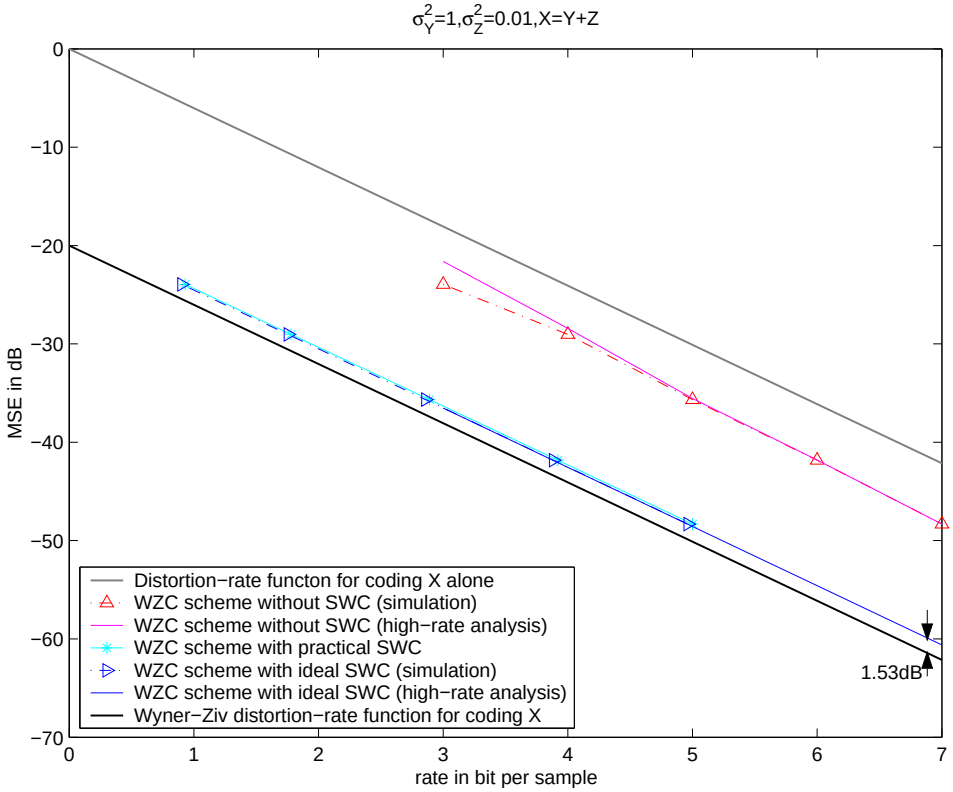

Fig. 25. Results based on nested scalar quantization with and without SWC for the bottom-up approach.

CHAPTER IV

GEL'FAND-PINSKER CODING DESIGN

In this chapter, we will describe the practical design of Gel'fand-Pinsker Coding (GPC) [47]. However, GPC is a rather general problem that actual design procedure relies on the more precise setting of the problem. For example, the side information can take as a form of interference or channel state information; the design procedures for these two cases can be rather different. In this thesis, we focus on the design of GPC as the form of a digital watermarking problem. We make this decision because digital watermarking is the precise dual of Wyner-Ziv coding (WZC) [104] as is shown in Section IIB. Although the nested coding approach is applicable in theory for the digital watermarking problem, the "channel noise" resulting from malicious attacks can be rather arbitrary in a practical digital watermarking scenario; in specific, the nested coding approach is not even robust against the common scaling attack. Hence, we will deviate from the nested coding approach and instead introduce an enhanced version of the well-known spread spectrum watermarking technique.

In the following, we will give an overview of digital watermarking and we will then describe the classic spread spectrum watermarking technique. We will introduce our enhance approach in Section IVC. To end this chapter, two applications related to digital watermarking will be depicted.

A. Overview of Digital Watermarking

Advances in compression technology have allowed multimedia data to be stored and distributed in digital form. On one hand, digital medium is very convenient for consumers; on the other hand, it poses severe threat to copyright owners (e.g., the record and film companies) as illegal copies can be freely reproduced and distributed

without any quality degradation. The situation will only get worse as the peer to peer and broadband technologies become more popular.

Thus there is an urgent need to protect digital medium from illicit use or distribution. Traditional cryptographic techniques are inadequate in this case because the protection is lost after the user decrypts the medium. This gives rise to the development of the digital watermarking technology. To achieve copyright protection, a digital watermark for ownership identification is inserted into the digital medium by the copyright owner before the medium is distributed to the consumer. Besides the identification of the copyright owner, the watermark can also contain information of the consumer to track the source of illicit distribution and ultimately to prevent this from happening.[1] Ideally, the watermark should always be present unless serious damage is introduced to the medium; to the extend that it is completely useless. In addition, the watermark should be perceptually transparent (or imperceptible) unless it is "visible" in nature. This means that the watermarked signal does not contain any perceivable artifact and hence is perceptually indistinguishable from the original signal. Furthermore, without knowing the exact "location" of the watermark, an attacker can only remove the watermark by brute force (i.e., distorting all samples). Therefore, an imperceptible watermark is also more robust than a perceptible one in the sense that a malicious user needs to "locate" the watermark before he/she can remove it effectively.

Practical watermarking techniques have existed for a long time. For example, paper millers in the medieval time added watermarks to their products to distinguish them from others [49]. Before the "universal developer" [52, pp. 523-525][71] was invented, invisible ink such as fruit juice had been used extensively to hide information

[1]This type of watermark is usually referred to as fingerprinting.

inside an innocent-looking letter. Popular techniques for copyright protection did not appear until the early 1990s. One of them is the spread spectrum (SS) watermarking technique introduced by Cox *et al.* [34], which borrows ideas from spread spectrum communications. The scheme has good robustness in general, but its performance is limited by the fact that the host signal itself acts as an interference to watermark recovery. The simple yet important technique known as low-bit modulation was mentioned and reinvented in several papers [49, 71, 91]. In contrast to SS watermarking, the host signal in low-bit modulation does not appear as an interference. Realizing these, Chen and Wornell developed QIM [18] with significant performance gain, much of which stems from the fact that the host signal in QIM does not interfere with watermark extraction. However, QIM, being a variation of nested coding, suffers the same weakness that the resulting watermark can be easily destroyed by scaling attack. In this thesis, we introduce an enhanced SS watermarking scheme that attempts to diminish the host signal interference while the robustness against scaling attacks is maintained. The key idea is to embed watermark into a transformed host signal which has a smaller variance.

In contemporaneous independent work, Malvar and Florencio proposed an improved SS watermarking in which host interference cancellation is achieved by carefully adjusting the distortion level at each signal sample [62]. Although their goal of reducing host interference is the same as ours, the approaches taken are different. While Malvar and Florencio impose constraint on the magnitude of sample distortion, we impose constraint on watermarking keys. These two approaches are complementary to each other, as one may be more suitable than the other depending on the application.

B. Spread Spectrum Watermarking

The main idea of spread spectrum watermarking is to spread the distortion introduced in the watermarking process to many samples. This increases the robustness of the watermark because attacking a few watermarked samples will unlikely be able to destroy the watermark. Moreover, it is easier to make the watermark imperceptible because the distortion to each individual sample is small.

Consider the host signal sequence as $\mathbf{S} = S_1, S_2, ..., S_n$ and assume that a single bit $b \in \{1, -1\}$ will be embedded into $\mathbf{S}$ for simplicity. The watermark embedding process can be summarized as [34]

$$X_i = S_i + b\Delta_i \cdot \kappa_i, \quad i = 1, 2, ..., n, \tag{4.1}$$

where X_i is a sample of the watermarked sequence and κ_i is a sample of a watermarking key sequence, which is also provided to the watermark decoder. In order to keep the secrecy of the watermark, the embedded bit b can only be retrieved with the same key sequence. The non-negative factor Δ_i is used to control the amount of distortion introduced to each host sample.

It is more convenient to write (4.1) in vector form as

$$\mathbf{X} = \mathbf{S} + b\boldsymbol{\Delta}\boldsymbol{\kappa}, \tag{4.2}$$

where $\mathbf{S} = (S_1, S_2, ..., S_n)^T$, $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, ..., \kappa_n)^T$, and $\boldsymbol{\Delta}$ is an $n \times n$ diagonal matrix with $\Delta_1, \Delta_2, ..., \Delta_n$ as its diagonal elements. In conventional SS watermarking, $\boldsymbol{\Delta}$ and $\boldsymbol{\kappa}$ are assumed to be independent of $\mathbf{S}$.

Assume the watermarked signal $\mathbf{X}$ is attacked and let $\mathbf{Y} = (Y_1, Y_2, ..., Y_n)^T$ be the signal after an additive attack. Write $\mathbf{Y} = \mathbf{X} + \mathbf{Z}$, where $\mathbf{Z} = (Z_1, Z_2, ..., Z_n)^T$ is the attack. We will try to extract b from $\mathbf{Y}$. To do this, we estimate b from the

inner product $\Pi = \langle \mathbf{Y}, \boldsymbol{\kappa} \rangle \triangleq \frac{1}{n} \sum_{i=1}^{n} Y_i \kappa_i$ of $\mathbf{Y}$ and $\boldsymbol{\kappa}$. Assume $\mathbf{S}$ and $\mathbf{Z}$ are i.i.d. and independent from each other[2] with zero mean and variances σ_S^2 and σ_Z^2, respectively. Then, we have

$$E[\Pi|\boldsymbol{\kappa}] = E[\langle \mathbf{Y}, \boldsymbol{\kappa} \rangle] \;\; = \;\; E[\langle \mathbf{S}, \boldsymbol{\kappa} \rangle] + E[b\langle \boldsymbol{\Delta}\boldsymbol{\kappa}, \boldsymbol{\kappa} \rangle] + E[\langle \mathbf{Z}, \boldsymbol{\kappa} \rangle] = b\,\frac{\mathrm{tr}(\boldsymbol{\Delta})}{n}$$

and

$$\begin{aligned} var[\Pi|\boldsymbol{\kappa}] &= var[\langle \mathbf{S}, \boldsymbol{\kappa} \rangle] + var[\langle \mathbf{Z}, \boldsymbol{\kappa} \rangle] = E[\langle \mathbf{S}, \boldsymbol{\kappa} \rangle^2] + E[\langle \mathbf{Z}, \boldsymbol{\kappa} \rangle^2] \\ &= \frac{1}{n^2} E\left[\sum_{i=1}^{n} \sum_{j=1}^{n} S_i \kappa_i S_j \kappa_j \right] + \frac{1}{n^2} E\left[\sum_{i=1}^{n} \sum_{j=1}^{n} Z_i \kappa_i Z_j \kappa_j \right] \\ &= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_S^2 \delta_{ij} \kappa_i \kappa_j + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_Z^2 \delta_{ij} \kappa_i \kappa_j = \frac{\sigma_S^2 + \sigma_Z^2}{n}, \end{aligned}$$

where δ_{ij} is equal to 1 for $i = j$ and 0 otherwise. Since $E[\Pi|\boldsymbol{\kappa}]$ is proportional to b and the scaling factor $\mathrm{tr}(\boldsymbol{\Delta})$ does not depend on b, we can determine b from Π using 0 as a threshold. That is, the estimate $\hat{b} = \mathrm{sgn}(\Pi)$. The complete SS watermarking system is summarized in Fig. 26.

If samples of $\mathbf{Z}$ and $\mathbf{S}$ are Gaussian distributed, the error probability in estimating b is

$$Pr(\hat{b} \neq b) = \frac{1}{2}\,\mathrm{erfc}\left(\frac{\mathrm{tr}(\boldsymbol{\Delta})}{\sqrt{2n(\sigma_S^2 + \sigma_Z^2)}} \right). \tag{4.3}$$

However, according to the central limit theorem, Eqn. (4.3) gives accurate estimate of the error probability even for non-Gaussian cases as long as n is large.

---

[2]Unfortunately, this assumption does not hold for deliberate attacks because the attacker is likely to respond according to what he/she receives. Hence, $\mathbf{Z}$ depends on $\mathbf{X}$, and thus on $\mathbf{S}$ in this case.

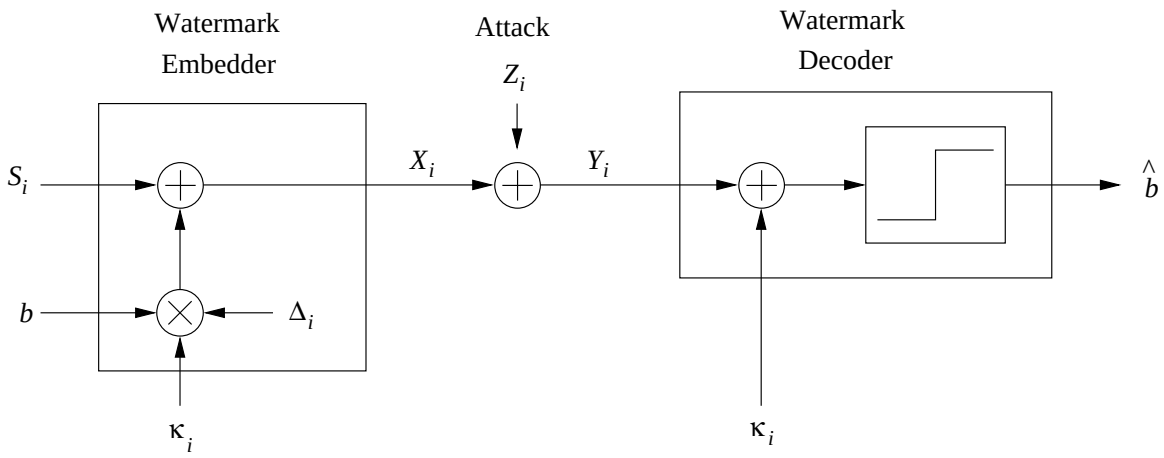

Fig. 26. Block diagram of a general SS watermarking system.

C. Enhanced Spread Spectrum Watermarking

1. Motivation

As we can see from (4.3), the probability of error depends on the original host signal σ_S^2. In particular, the larger the variance σ_S^2, the higher the probability of error. This means that the host signal sequence $\mathbf{S}$ appears as an interference to estimating b. When $\mathbf{S}$ is known to the decoder, this interference can be eliminated by subtracting $\langle \mathbf{S}, \boldsymbol{\kappa} \rangle$ from Π. This scenario is commonly known as private watermarking. However, it is much more realistic in practice to consider public watermarking when $\mathbf{S}$ is unknown to the decoder.

When both the noise and host signals are Gaussian, it is possible to construct a public watermarking scheme with equal embedding capability as the best private watermarking scheme [29]. Loosely speaking, this means that under the Gaussian assumption, it is possible to eliminate the host signal interference even for public watermarking. Stronger results when the host and the attack signals are not stationary

or ergodic and when the attack is arbitrary with constrained squared-error distortion are proved in [110] and [25], respectively. Practical approaches such as QIM and scalar Costa scheme are described in [18, 39]. The simplest example of these zero host interference schemes is low-bit modulation [71]. The watermark embedder simply replaces the least significant bit of the host signal with the embedded bit. The watermark decoder extracts the embedded bit directly from the least significant bit of the watermarked signal. Obviously, the error probability of decoding the embedded bit does not depend on the host signal. As pointed out in [62], the main weakness of these schemes is that the watermark can be easily destroyed by scaling.

In the analysis of the SS watermarking scheme (see (4.1)–(4.3)), we assume that both $\boldsymbol{\kappa}$ and $\boldsymbol{\Delta}$ are independent of $\mathbf{S}$. The improved scheme in [62] takes Δ_i as a function of $\mathbf{S}$ and constrains the average distortion by limiting the magnitude of $E[\Delta_i]$. This scheme is shown to be able to withstand 20 dB more of noise power, compared with the conventional SS scheme for high (host) signal to (attack) noise ratio. The main difficulty in applying this scheme is that one may not have the freedom of varying Δ_i. This happens when the distortion of individual samples rather than the average distortion is important. For instance, watermark embedding of audio signal is usually applied in the frequency domain. A masking function [68] is computed for each group of coefficients known as a bark band and the modification will be imperceptible as long as the distortion introduced to each coefficient has value smaller than the mask of the corresponding bark band it belongs to. Assume the mask function on a sample is $\mathfrak{m}_i$, the best Δ_i to allow maximum hiding capability is simply equal to $\mathfrak{m}_i$. That is, the maximum allowed watermarking power should be used. Therefore, we do not have the freedom to cancel the effect of $\mathbf{S}$ by varying Δ_i in this case. In our work, we reduce the interference by varying $\boldsymbol{\kappa}$ instead and our goal is to confine the choice of keys to those satisfying $var(\langle \mathbf{S}, \boldsymbol{\kappa} \rangle) \approx 0$.

2. System Setup

In this section, we propose an enhanced SS watermarking scheme by "transforming" the host sequence into another one before watermark embedding. As a result, the variance of the transformed host sequence will be much smaller than before, and so will be the probability of error according to (4.3). We show that the effect is the same as using a signal dependent key described in the previous Section.

In a nutshell, we perform the following: 1) Sort the original sequence; 2) construct a new host signal by taking the difference of every two consecutive samples in the sorted sequence; and 3) add a watermark to the new host signal.

In practice, we do not actually construct the new host signal during watermark embedding. Instead, we generate the watermarked signal directly from the original host signal using the sorting index obtained in step 1). Specifically, we first sort $\mathbf{S}$ in ascending order and obtain $S_{I_1} \leq S_{I_2} \leq S_{I_3} ... \leq S_{I_n}$. Then, assuming n is even for the sake of simplicity, we construct a hypothetical host signal sequence $\mathbf{S}' = (S'_1, S'_2, ..., S'_{n/2})^T$ with

$$S'_j = (-1)^j (S_{I_{2j}} - S_{I_{2j-1}}). \tag{4.4}$$

We explicitly make the two consecutive S'_j's have alternate signs to ensure that $\sum_{j=1}^{n/2} S'_j$ has approximately zero mean. Since adjacent sorted samples of $\mathbf{S}$ are close in their values, we conclude that with high probability, the sample values of $\mathbf{S}'$ will be much smaller than those of $\mathbf{S}$. Moreover, $\sigma^2_{S'}$ is much smaller than σ^2_S.

We "embed" bit b into $\mathbf{S}'$ as follows. We prepare a length-$n/2$ watermark key sequence $\kappa' = (\kappa'_1, \kappa'_2, ..., \kappa'_{n/2})^T$, $\kappa'_i = \{-1, 1\}$ and generate the watermarked sequence $\mathbf{X}' = (X'_1, X'_2, ..., X'_{n/2})^T$ as

$$X'_j = S'_j + b\Delta'_j \kappa'_j \tag{4.5}$$

just like in conventional SS watermarking, where Δ'_j controls the amount of distortion that can be added to S'_j. As mentioned previously, we do not actually construct $\mathbf{X}'$ but instead we modify $\mathbf{S}$ into sequence $\mathbf{X} = (X_1, X_2, ..., X_n)^T$ such that $\mathbf{X}'$, which satisfies (4.5), is a transform of $\mathbf{X}$ just as $\mathbf{S}'$ is a transform of $\mathbf{S}$. Since S'_j is related to both $S_{I_{2j-1}}$ and $S_{I_{2j}}$, the perturbation $b\Delta'_j\kappa'_j$ can be achieved by varying $S_{I_{2j-1}}$ and/or $S_{I_{2j}}$. This extra flexibility is useful because $S_{I_{2j-1}}$ and $S_{I_{2j}}$ may have unequal susceptibility to noise. For simplicity, we split the distortion evenly among the pair. Thus we construct

$$X_{I_{2j}} = S_{I_{2j}} + (-1)^j b\frac{\Delta'_j}{2}\kappa'_j; \qquad X_{I_{2j-1}} = S_{I_{2j-1}} - (-1)^j b\frac{\Delta'_j}{2}\kappa'_j, \tag{4.6}$$

and obtain $X'_j = (-1)^j(X_{I_{2j}} - X_{I_{2j-1}}) = S'_j + b\Delta'_j\kappa'_j$ as desired.

Recall that $\mathbf{Y}$ is the distorted $\mathbf{X}$ received by the watermark decoder. To decode b, we first transform $\mathbf{Y}$ into another sequence $\mathbf{Y}' = (Y'_1, Y'_2, ..., Y'_{n/2})^T$ with the help of the sorting indices $\mathbf{I} \triangleq (I_1, I_2...I_n)^T$ as in (4.4). The embedded bit $\hat{b}$ is then estimated to be $\text{sgn}\left(\frac{2}{n}\sum_{j=1}^{n/2}\kappa'_j Y'_j\right)$ as in conventional SS watermarking. Therefore, both the sorting indices $\mathbf{I}$ and the watermark key $\boldsymbol{\kappa}'$ are required for decoding.

If we let

$$\kappa_{I_i} = \begin{cases} (-1)^{\frac{i}{2}}\kappa'_{\frac{i}{2}}, & i \text{ is even} \\ -(-1)^{\frac{i+1}{2}}\kappa'_{\frac{i+1}{2}}, & i \text{ is odd} \end{cases} \tag{4.7}$$

and

$$\Delta_{I_i} = \begin{cases} \Delta'_{\frac{i}{2}}/2, & i \text{ is even} \\ \Delta'_{\frac{i+1}{2}}/2, & i \text{ is odd,} \end{cases} \tag{4.8}$$

we obtain $X_i = S_i + b\Delta_i\kappa_i$ from (4.6). The expression is the same as that of the embedding process in conventional SS watermarking. But unlike the traditional scheme,

$\boldsymbol{\kappa}$, which can be viewed as a combined key of $\mathbf{I}$ and $\boldsymbol{\kappa}'$, does depend on the host signal $\mathbf{S}$. Therefore, we can consider our enhanced SS scheme as conventional SS scheme but with a signal dependent key. This dependency reduces the degree of freedom of $\boldsymbol{\kappa}$ from n to $n/2$. In other words, the number of valid keys decreases from 2^n to $2^{n/2}$. Note that $var(\langle \mathbf{S}, \boldsymbol{\kappa} \rangle) = \frac{1}{n} \sum_{j=1}^{n/2} \kappa'_j (-1)^j var(S_{I_{2j}} - S_{I_{2j-1}}) \approx 0$ as $S_{I_{2j-1}} \approx S_{I_{2j}}$. Constructing $\boldsymbol{\kappa}$ can be considered as selecting good keys out of all possible keys that satisfy $var(\langle \mathbf{S}, \boldsymbol{\kappa} \rangle) \approx 0$.

It seems that the reduction in the number of valid keys may result in lesser security, because the smaller the size of the key set implies the easier the embedded information to be extracted by an unauthorized person using brute force. However, if we use a different key for each embedding bit,[3] it is easy to show that the resulting enhanced SS watermarking scheme has perfect secrecy [90, ch. 2]. In practice, these keys can be constructed as consecutive sections cutting from a pseudo-random sequence generated by a single seed.

3. Performance Analysis

Define $\Pi' \triangleq \langle \boldsymbol{\kappa}', \mathbf{Y}' \rangle$, the estimate of embedded bit $\hat{b} = \text{sgn}(\Pi')$. Since $Y'_j = (-1)^i (Y_{I_{2j}} - Y_{I_{2j-1]}}) = S'_j + b\Delta'_j \kappa'_j + (-1)^j (Z_{I_{2j}} - Z_{I_{2j-1}})$, then

$$E[\Pi' | \boldsymbol{\kappa}'] = E[\langle \mathbf{S}', \boldsymbol{\kappa}' \rangle | \boldsymbol{\kappa}'] + b \left(\frac{2}{n} \right) \text{tr}(\boldsymbol{\Delta}') = E[\langle \mathbf{S}', \boldsymbol{\kappa}' \rangle | \boldsymbol{\kappa}'] + b \left(\frac{2}{n} \right) \text{tr}(\boldsymbol{\Delta})$$

because $\text{tr}(\boldsymbol{\Delta}) = \sum_{i=1}^{n} \Delta_i = \sum_{i=1}^{n} \Delta_{I_i} = \sum_{j=1}^{n/2} \Delta_{I_{2j}} + \Delta_{I_{2j-1}} = \sum_{j=1}^{n/2} \Delta'_j/2 + \Delta'_j/2 = \text{tr}(\boldsymbol{\Delta}')$. In addition,

$$var[\Pi' | \boldsymbol{\kappa}'] = var[\langle \mathbf{S}', \boldsymbol{\kappa}' \rangle | \boldsymbol{\kappa}'] + \left(\frac{2}{n} \right)^2 n\sigma_Z^2.$$

[3] This is, of course, just the main idea of one-time pad.

The transform of $\mathbf{S}$ to $\mathbf{S}'$ is non-linear. This makes it almost impossible to compute the exact $var[\langle \mathbf{S}', \boldsymbol{\kappa}' \rangle | \boldsymbol{\kappa}']$. Therefore, we further assume that elements of $\boldsymbol{\kappa}'$ are i.i.d. with zero mean and that $\mathbf{S}'$ and $\boldsymbol{\kappa}'$ are independent, then

$$E_{\boldsymbol{\kappa}'}[E[\langle \mathbf{S}', \boldsymbol{\kappa}' \rangle | \boldsymbol{\kappa}']] = \frac{2}{n} E_{\boldsymbol{\kappa}'} \left[\sum_{j=1}^{n/2} E[S'_j] \kappa'_j \right] = 0$$

and

$$\begin{aligned} E_{\boldsymbol{\kappa}'}[var[\langle \mathbf{S}', \boldsymbol{\kappa}' \rangle | \boldsymbol{\kappa}']] &\approx E_{\boldsymbol{\kappa}'}[E[\langle \mathbf{S}', \boldsymbol{\kappa}' \rangle^2 | \boldsymbol{\kappa}']] \\ &= \frac{4}{n^2} \left(\sum_{j=1}^{n/2} E[S'^2_j] + E_{\boldsymbol{\kappa}'} \left[\sum_{j=1}^{n/2} \sum_{i \neq j} E[S'_i S'_j] \kappa'_i \kappa'_j \right] \right) \\ &= \frac{4}{n^2} \sum_{j=1}^{n/2} \sigma^2_{S'_j}. \end{aligned}$$

By the central limit theorem, the probability of error can be approximated as[4]

$$Pr(\hat{b} \neq b) \approx \frac{1}{2} \text{erfc} \left(\frac{E_{\boldsymbol{\kappa}'}[E[\Pi' | \boldsymbol{\kappa}']]}{\sqrt{2 E_{\boldsymbol{\kappa}'}[var[\Pi' | \boldsymbol{\kappa}']]}} \right) = \frac{1}{2} \text{erfc} \left(\frac{\text{tr}(\boldsymbol{\Delta})}{\sqrt{2(\sum_{j=1}^{n/2} \sigma^2_{S'_j} + n\sigma^2_Z)}} \right). \quad (4.9)$$

From (4.9), we need to find the statistics of S'_j's in order to compute the error probability. Recall $\mathbf{S_I} = (S_{I_1}, S_{I_2} ... S_{I_n})^T$ is the sorted sequence of $\mathbf{S}$ in ascending order. Assume S_i has a cumulative distribution function $F_S(s)$ and a probability density function $f_S(s)$. The joint probability density function of the two consecutive

---

[4]Strictly speaking, the probability of error should be $E_{\boldsymbol{\kappa}'} \left[\frac{1}{2} \text{erfc} \left(\frac{E[\Pi' | \boldsymbol{\kappa}']}{\sqrt{2var[\Pi' | \boldsymbol{\kappa}']}} \right) \right]$. However, (4.9) gives a reasonable approximation as $E[\Pi' | \boldsymbol{\kappa}']$ and $var[\Pi' | \boldsymbol{\kappa}']$ are approximately constants with respect to $\boldsymbol{\kappa}'$. Note that $E[\Pi' | \boldsymbol{\kappa}']$ and $var[\Pi' | \boldsymbol{\kappa}']$ are related to $\boldsymbol{\kappa}'$ via the terms $\sum_{j=1}^{n/2} E[S'_j] \kappa'_j$ and $\sum_{j=1}^{n/2} \sum_{i \neq j} E[S'_i S'_j] \kappa'_i \kappa'_j$, respectively. Since S'_i's have alternating signs and are independent of $\boldsymbol{\kappa}'$, the summands of $\sum_{j=1}^{n/2} E[S'_j] \kappa'_j$ and $\sum_{j=1}^{n/2} \sum_{i \neq j} E[S'_i S'_j] \kappa'_i \kappa'_j$ will tend to cancel out and hence the sums will be close to 0.

ordered random variables S_{I_l} and $S_{I_{l+1}}$ is [69, pp. 246]

$$f_{S_{I_l}S_{I_{l+1}}}(S_{I_l}, S_{I_{l+1}})$$
$$= \begin{cases} \frac{n!}{(l-1)!(n-l)!} F_S(S_{I_l})^{l-1} f_S(S_{I_l}) f_S(S_{I_{l+1}})(1 - F_S(S_{I_{l+1}}))^{n-l-1}, & S_{I_l} \leq S_{I_{l+1}} \\ 0, & S_{I_l} > S_{I_{l+1}}. \end{cases}$$

For $S_{I_l} > S_{I_{l+1}}$, the density is obviously 0 because $\mathbf{S_I}$ are ordered ascendingly. For $S_{I_l} \leq S_{I_{l+1}}$, we want $l-1$ S's to be no bigger than S_{I_l}, $n-l-1$ S's to be no smaller than $S_{I_{l+1}}$, and have the remaining two S's take values S_{I_l} and $S_{I_{l+1}}$.

Let $d_j = S_{I_{2j}} - S_{I_{2j-1}}$, $j = 1, 2, ...\frac{n}{2}$, then

$$f_{d_j}(d) = \begin{cases} \frac{n!}{(2j-2)!(n-2j)!} \int_{\mathbb{R}} F_S(s)^{2j-2} f_S(s) f_S(d+s)(1 - F_S(d+s))^{n-2j} ds, & d > 0 \\ 0, & \text{otherwise.} \end{cases}$$

Note that $S'_j = (-1)^j d_j$, hence $\sigma^2_{S'_j} = \sigma^2_{d_j}$.

Uniform Host Signal

We can find $\sigma^2_{S'_j}$ exactly in only a few cases. One of them is when the host signal is uniformly distributed. Assume that S_i is uniformly distributed with support ω, i.e.,

$$f_{S_i}(s) = \begin{cases} \frac{1}{\omega} & s \in [0, \omega] \\ 0 & \text{otherwise,} \end{cases}$$

then for $d > 0$,

$$\begin{aligned} f_{d_j}(d) &= \frac{n!}{(2j-2)!(n-2j)!} \int_0^{\omega - d} \left(\frac{S}{\omega}\right)^{2j-2} \left(\frac{1}{\omega}\right)^2 \left(1 - \frac{S+d}{\omega}\right)^{n-2j} ds \\ &= \frac{n}{\omega^n}(\omega - d)^{n-1}. \end{aligned}$$

Hence $E[d_j] = \frac{\omega}{n+1}$ and $E[d_j^2] = \frac{2\omega^2}{(n+1)(n+2)}$. We have $\sigma^2_{S'_j} = \sigma^2_{d_j} = E[d_j^2] - E[d_j]^2 =$

$\frac{12n\sigma_S^2}{(n+1)^2(n+2)}$. Thus,

$$Pr(\hat{b} \neq b) \approx \frac{1}{2}\operatorname{erfc}\left(\frac{\operatorname{tr}(\mathbf{\Delta})}{\sqrt{2(\frac{6n^2\sigma_S^2}{(n+1)^2(n+2)} + n\sigma_Z^2)}}\right). \tag{4.10}$$

The host signal interference tends to disappear as $\sum_i^{n/2} \sigma_{S_j'}^2 = \frac{6n^2\sigma_S^2}{(n+1)^2(n+2)}$ tends to 0 for large n.

We compare enhanced SS watermarking with conventional SS watermarking, the spread-transform dither modulation (STDM), and an ideal case when host signal is known to the decoder and hence does not act as an interference. STDM is a combination of QIM and conventional SS watermarking [18]. It happens to have the same weakness as QIM in the sense that its watermark can be easily destroyed by signal scaling. Chen and Wornell show the watermarking power of STDM is 1.25 dB higher than that in the ideal case in order to achieve the same probability of error [18].

The performances of the four different cases are shown in Fig. 27. We plot the log error probability against $\operatorname{tr}(\mathbf{\Delta})/n\sigma_S = \frac{1}{n}\sum_{i=1}^{n} \Delta_i/\sigma_S$, which is the average magnitude of the watermark to that of the host signal. Results for several n's and (host) signal to (attack) noise ratios are computed. For all combinations, enhanced SS watermarking has performance very close to that of the ideal case and achieves lower error probability than that of STDM for the same allowed distortion. Although there are other QIM variations [39] that offer better performance than STDM, their watermarks have high susceptibility to signal scaling as in the QIM case.

For large n, $\sum_j^{n/2} \sigma_{S_j'}^2$ is practically 0. The error probability $Pr(\hat{b} \neq b)$ is then approximately $\frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{\operatorname{tr}^2(\mathbf{\Delta})}{2(n\sigma_Z^2)}}\right)$, and depends on $\operatorname{tr}(\mathbf{\Delta})^2/n$ as a whole. Therefore, like conventional SS watermarking, we can tradeoff the embedding rate and the distortion introduced to the watermarked signal. For example, to reduce the absolute distortion

by half, one needs approximately quadruple n to keep the same probability of error.

It is interesting to look at the actual robustness gain of enhanced SS watermarking. We want to see how much more noise an enhanced SS watermark can withstand comparing to a conventional SS watermark. From (4.10) and (4.3), we have $\left(\frac{\sigma_Z}{\sigma_{Z_0}}\right)^2 = 1 + \left(1 - \frac{6n}{(n+1)^2(n+2)}\right)\left(\frac{\sigma_S}{\sigma_{Z_0}}\right)^2$, where $\sigma_{Z_0}^2$ and σ_Z^2 are the noise variances introduced to the conventional SS watermark and the enhanced SS watermark that result in the same amount of error probability. Fig. 28 plots $\left(\frac{\sigma_Z}{\sigma_{Z_0}}\right)^2$ against $\left(\frac{\sigma_S}{\sigma_{Z_0}}\right)^2$ to show the robustness gain as the signal to noise ratio increases for different n. Enhanced SS watermarking offers almost the same gain as the ideal case (without host signal interference), which is also included in Fig. 28. We expect the enhanced scheme to approach the ideal case as $n \to \infty$. In fact, enhanced SS watermarking realizes most of the gain even when n is as small as two.[5] This is because the host signal does not need to be eliminated completely to achieve a huge gain. This last point is verified in Fig. 29 which shows the theoretical robustness gain obtained by reducing the host signal variance. The robustness gain is merely 3 dB away from the ideal case even when the host signal variance is only reduced by half.

Another observation from Fig. 28 is that $\left(\frac{\sigma_Z}{\sigma_{Z_0}}\right)^2 \approx \left(\frac{\sigma_S}{\sigma_{Z_0}}\right)^2$ for large σ_S/σ_{Z_0}. This is intuitive as the signal to noise ratio increases, the error probability is dominated by the host signal interference in conventional SS watermarking. Since there is almost no host interference in enhanced SS watermarking, its watermark can withstand as much noise as the combined noise power in conventional SS watermarking, which is approximately equal to the host signal power.

Gaussian Host Signal

When the host signal is Gaussian, it is difficult to find $\sigma_{S_j'}$ as in the uniform case.

[5] A caveat is that the analysis here only give a crude estimation as the central limit theorem will no longer be accurate for such a small n.

Instead we perform Monte Carlo simulations to obtain an estimate of $\sum_{j=1}^{n/2} \sigma_{S'_j}^2$ as shown in Table IX. 100000 sequences are generated to get the estimate for each n.

Table IX. Estimates of the sums of $\sigma_{S'_j}$ for the Gaussian distributed host signal.

n	10	50	200	500
$\sum_{j=1}^{n/2} \sigma_{S'_j}^2$	$0.61\sigma_S^2$	$0.40\sigma_S^2$	$0.28\sigma_S^2$	$0.23\sigma_S^2$

Using Table IX and (4.9), we can calculate the probability of error for enhanced SS watermarking. The results are shown in Fig. 30. Results for conventional SS watermarking, STDM, and the ideal case are also included for comparison. As in the uniform case, enhanced SS watermarking has error probability smaller than STDM and very close to the ideal case.

The robustness gain of enhanced SS watermarking, i.e., the amount of additional noise that can be withstood comparing to conventional SS watermarking, is computed and shown in Fig. 31. The gain for the ideal case is also shown for comparison. Just as in the uniform case, enhanced SS watermarking with Gaussian host signals has its robustness gain quite close to the ideal case even for an n as small as two.

4. Discussion

The basic idea of enhanced SS watermarking is to impose constraint on the valid key set so that host interference cancellation is achieved. Our "sorting and sample differentiation" scheme is only one way to accomplish this goal. Thus one interesting further direction is to look for an optimal scheme that satisfies either 1) the constraint is weakest (the number of valid keys is largest) given the same level of host interference; or 2) the host interference is minimized given the same level of constraint is imposed (the number of valid keys is the same).

Although the watermarking key $\boldsymbol{\kappa}$ can be arbitrarily chosen in theory, it is usually implemented as a pseudo-random sequence for the ease of transmission. Due to the dependency on the host signal, $\boldsymbol{\kappa}$ can no longer be reproduced as a pseudo-random sequence with a single seed. In practice, the user key $\boldsymbol{\kappa}'$ can still be generated from a seed, and the sorting indices $I_1, I_2, ..., I_n$ are needed separately in order to reconstruct the watermarking key $\boldsymbol{\kappa}$. It is possible to pad these sorting indices directly into the user key but these uncoded indices can contribute significant overhead when n is large. There are at least two compatible approaches in tackling this issue. The first one is by taking advantage between the correlation of the host signal and the received watermarked signal. For example, we can transmit a check sum that contributes a smaller overhead instead of the entire sorting indices. In the decoder, we can estimate the sorting indices from the watermarked signal and compare the checksum of the estimate with the received one. If the checksums do not match, the next best estimate is chosen[6] and the same test is performed again. This search is repeated for a maximum number of times or until the checksum is satisfied.

Another approach is by imposing constraint on the choice of the indices. Note that sorting used in our setup is just one way of permuting the host signal samples. Moreover, no restriction has been imposed on the choice of this permutation. We can reduce the amount of the overhead by restriction this choice, or equivalently by reducing the number of allowed permutationsi. For example, instead of sorting the whole n-sample sequence, we can first divide it into m subblocks and perform sorting individually. When this number is reduced to 1, i.e., only the identity permutation is allowed, this degenerates back to the conventional SS scheme with no overhead at all.

[6]For our setup, this can be done effectively by swapping the two neighboring indices with the smallest sample difference in the current permutation. Each newly generated permutation is recorded to avoid any repetition.

On the other hand, our current scheme corresponds to the other extreme case when the maximum number of permutations is allowed. There should exist an optimal number of allowed permutation that gives the highest robustness of the watermark with a fixed amount of overhead. While we do not know what this optimum is, it is reasonable to believe that this optimum lies between the two extreme cases (our enhanced SS scheme and the conventional SS scheme).

D. Applications

In this section, we will describe two applications relating to digital watermarking: AAC audio watermarking and AAC audio error concealment.

1. AAC Audio Watermarking

In the past decade, advances in audio compression have made distribution of digital audio easy and convenient. Many commercial and non-commercial techniques have been invented. The highly successful MPEG-1 layer 3 (MP3) [15] audio coder and its successor AAC [3] are two examples. However, as mentioned previously in this chapter, these compression techniques also pose a serious threat to the record and film companies because they make illegal distribution of digital audio easy and convenient. As a result, several watermarking schemes have been proposed to address this problem [53, 66, 8].

The AAC encoding procedure consists of four steps: frequency transform, quantization, entropy coding, and bitstream multiplexing [3, 14, 1, 2]. AAC employs the modified discrete cosine transform (MDCT) [61] typically with 1024 samples per frame. Perceptual modeling [68] is applied to estimate for each Bark band the maximum amount of distortion that can be withstood. The quantization step size is

iteratively[7] adjusted until both the bit rate is below the target and the distortion is below the maximum acceptable perceptual threshold. Huffmann coding is then used to encode the quantized coefficients and the step size information. Finally, the encoded indices are multiplexed into one single bitstream.

In [66], an AAC audio watermarking scheme is proposed, which partially decodes the compressed audio in the frequency domain and requantizes[8] it after embedding a perceptually imperceivable watermark. The distortion introduced to each frequency coefficient is determined by the perceptual threshold, which is assumed to be recorded during the original compression process and passed onto the compressed audio.

a. Proposed AAC Watermarking System

A drawback of the approach in [66] is that the perceptual modeling information is usually not available when the watermark is added. It is unlikely that the compressed audio clip stores this extra information. Although there is watermarking system that estimates the perceptual information from the compressed audio, this results in estimation error and increased complexity [67]. Therefore, we assume that no such information is available during watermark embedding and take a heuristic approach instead. In addition, we embed the watermark directly into the quantization indices rather than the MDCT coefficients. This speeds up the overall watermarking process since no dequantization and requantization is necessary. Furthermore, this avoids any tandem coding distortions (i.e., distortions accumulation due to repeated quantization). We propose an AAC audio watermarking scheme based on our enhanced SS

[7]Note that there exists more sophisticated encoding system that employs forward estimation of quantization step sizes without needs of iterations.

[8]Same quantization step sizes are used to avoid tandem coding distortions (i.e., distortions accumulation due to repeated quantization).

approach. A user key $\boldsymbol{\kappa}'$ and the host signal samples $S_1, S_2, ..., S_n$, which are the quantization indices in this case, are given to the watermark embedder. The host signal samples are then sorted and the sorting indices $I_1, I_2, ..., I_n$ are generated. A combined key $\boldsymbol{\kappa}$, which depends on both the user key and the sorting indices, is constructed as in (4.7) and sent to the decoder for watermark decoding. Alternatively, the user key and the sorting indices can be transmitted separately to regenerate the combined key $\boldsymbol{\kappa}$ at the decoder. Block diagrams for watermark embedding and decoding in our proposed system are shown in Fig. 32 (a) and (b), respectively.

The distortion control in Fig. 32 determines Δ_i, which controls the amount of distortion imposed to the i^{th} quantization index. Ideally, this can be done by applying perceptual modeling to the original audio. For example, if one coefficient can tolerate a distortion of 10 units and its current quantization step size is two. Then we can approximately vary the corresponding index by five steps without affecting the quality.[9] However, as mentioned before, this information is not easily accessible during watermark embedding. Therefore, a heuristic scheme is employed as follows:

1. Pick indices corresponding to a frequency range in which the human ears are more sensitive to distortion (to prevent destruction of the watermark by frequency truncation attack).

2. Set Δ_i to 0 for zero indices (to avoid having distortion during silent period).

3. Set Δ_i to be 1 for the remaining indices (to minimize distortion).

The modified quantization indices after watermarking are compressed with Huffman coding using the original codebook. It is possible to search for the optimum

[9]Uniform quantization is assumed in this idealized example.

Table X. Noise-to-mask ratio (NMR) of watermarked audio.

Audio	clip1	clip2	clip3	clip4	clip5	clip6	clip7
NMR	-1.40	-9.44	-8.87	-7.21	-3.20	-6.01	-8.38

codebook again as in AAC encoding. However, we do not take this approach due to complexity concerns.

We decode the watermark from the MDCT coefficients instead of the quantization indices directly. This is because the quantization indices are relatively vulnerable to digital/analog/digital conversion as quantization step sizes may change.

b. Experimental Results

Perceptual quality

Although we use a heuristic estimate on the perceptual model for the watermark embedding, test results show the perceptual quality were acceptable under office or lab environments where the tests are conducted. We provide in Table X the noise-to-mask ratio (NMR) [16] for the watermarked audio as an objective measure of the audio quality. Samples of watermarked music clips can also be found at http://samuel.ee.tamu.edu/research/aactest.asp.

Information hiding capacity

In general, the robustness of the watermark drops naturally with the increase of information hiding rate. Information hiding capacity is defined as the maximum amount of information that can be embedded into the host signal while guaranteeing correct retrieval. We estimate this information hiding capacity by measuring the watermark bit error rate (WBER) for different embedding rates and different audio clips without noise (Table XI). The information hiding capacity of our system is

approximately 30 bps. The source of error is mainly due to imperfect host interference elimination as the sorting indices are obtained from ordering the quantization indices instead of the MDCT coefficients, where the latters are actually used for watermark recovery. This is a trade-off with the complexity as MDCT transform is not necessary in our encoding scheme.

Table XI. Watermark bit error rate at different embedding rate.

Audio	Embedding Rate (bps)				
	10	20	30	50	80
clip1	0	0	0	0.001	0.0037
clip2	0	0	0	0	0
clip3	0	0	0	0	0
clip4	0	0	0	0.013	0.0014
clip5	0	0	0	0	0
clip6	0	0	0	0	0
clip7	0	0	0	0	0

Robustness against transcoding

To estimate the robustness of our watermark against transcoding, we first decode the watermarked AAC audio to WAV format and then convert it to MP3 format. It is then decoded to WAV format again and encoded back to AAC format. The transcoded AAC audio is inputted to the decoder for the watermark retrieval after resynchronization. The resulting WBER is shown in Table XII. Note that there is some drop in the WBER even when the information hiding rate increases, this is probably due to statistical error since the test audio sequence is pretty short (< 20 sec on average) for good perceptual quality test. Despite the increase in the WBER

Table XII. Watermark bit error rate at different embedding rate after MP3 transcoding.

Audio	Embedding Rate (bps)				
	10	20	30	50	80
clip1	0	0.003	0.003	0.014	0.010
clip2	0	0	0	0	0
clip3	0	0	0	0.001	0.001
clip4	0	0.002	0.004	0.007	0.025
clip5	0	0	0.002	0	0
clip6	0	0	0	0	0
clip7	0	0	0	0.002	0.002

after transcoding, the watermark is still shown to be robust at a relatively high rate of 10 bps.

Change in file size after watermarking

An increase of audio clip size after watermarking is expected due to data embedding. Table XIII lists the percentage increase in file size for different test clips after watermarking. The increase in file size is below 5% for all test clips and about 1% higher than that obtained in [66]. The extra 1% increase could be due to the fact that our simplified approach does not perform any AAC encoding steps except Huffmann encoding. Performing those AAC encoding steps (e.g., searching for optimum Huffmann tables) will increase the compression ratio at the cost of increased complexity.

Table XIII. Percentage size change after watermarking.

Audio	clip1	clip2	clip3	clip4	clip5	clip6	clip7
Size Increase	3.9%	3.8%	2.0%	3.0%	3.8%	4.0%	1.7%

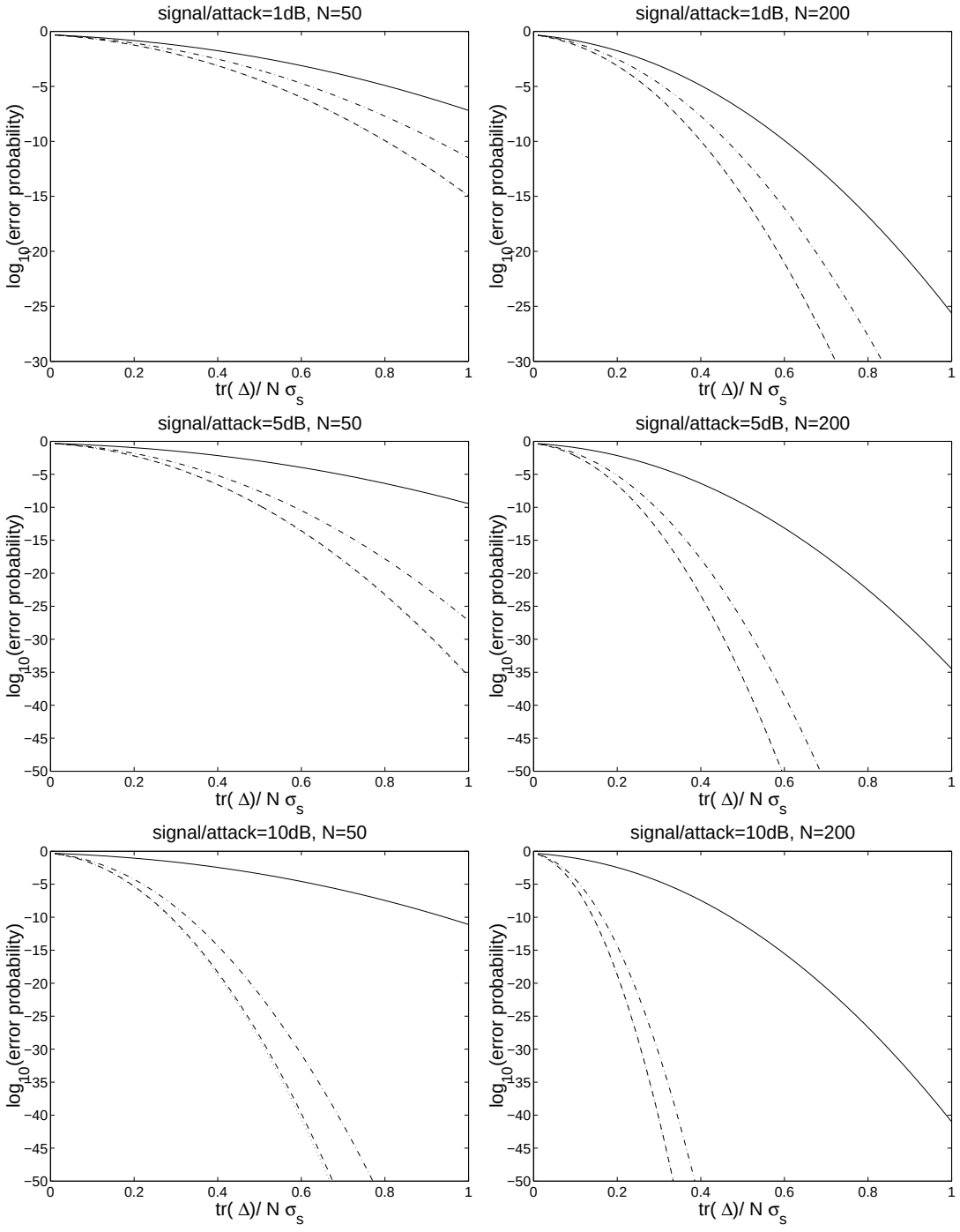

Fig. 27. The figures compare the performance of enhanced SS watermarking (dashed lines) from conventional SS watermarking (solid lines) and STDM (dash-dot lines) for uniform host signal. The ideal case that without host signal interference (dotted lines) is also shown.

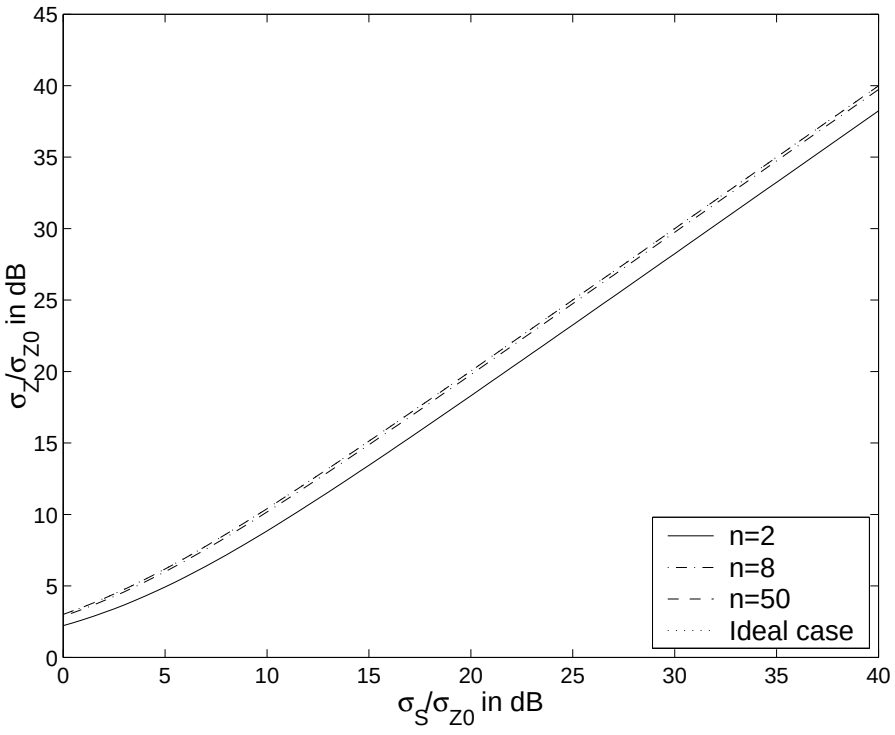

Fig. 28. The figures shows the robustness gain against the (host) signal to (attack) noise ratio for $n = 2, 8, 50$ when the host signal is uniformly distributed. An ideal case with no host signal interference is also shown for comparison.

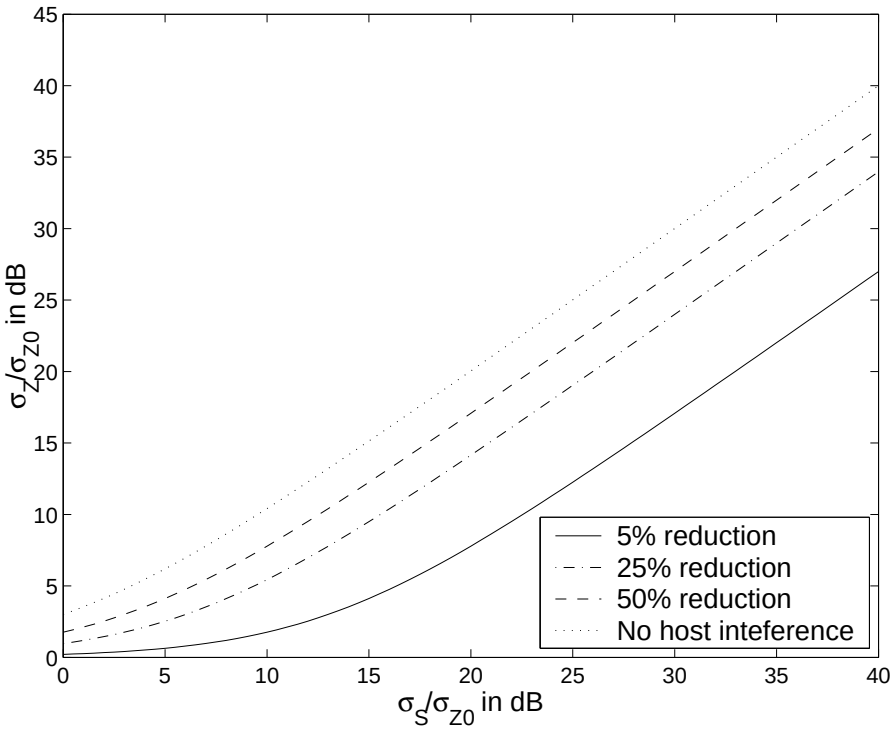

Fig. 29. The figure shows the robustness gain against the (host) signal to (attack) noise ratio for different reductions of host signal variance. An ideal case with no host signal interference is also shown for comparison.

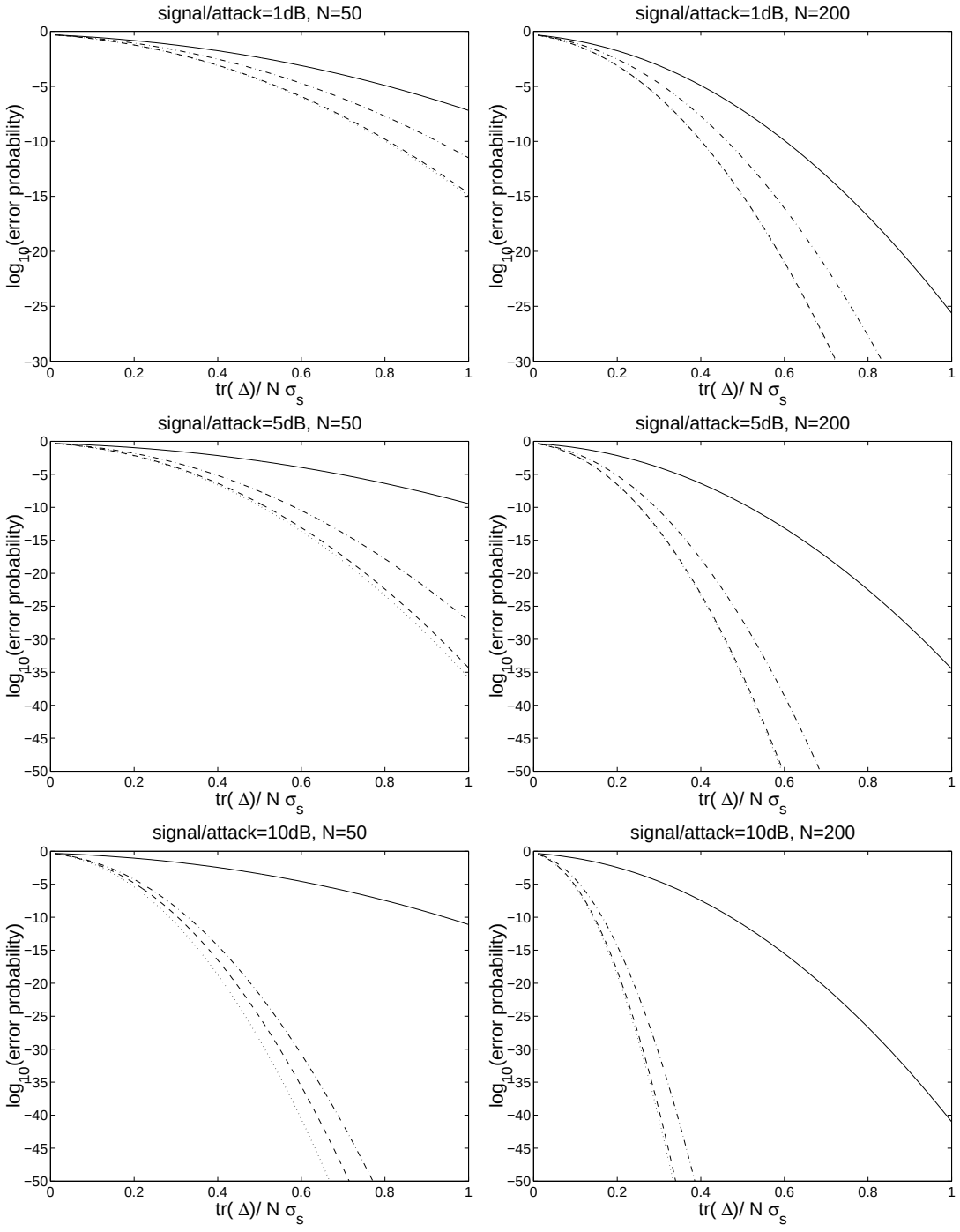

Fig. 30. The figures compare the performance of enhanced SS watermarking (dashed lines) from conventional SS watermarking (solid lines) and STDM (dash-dot lines) for Gaussian distributed host signal. The ideal case that without host signal interference (dotted lines) is also shown.

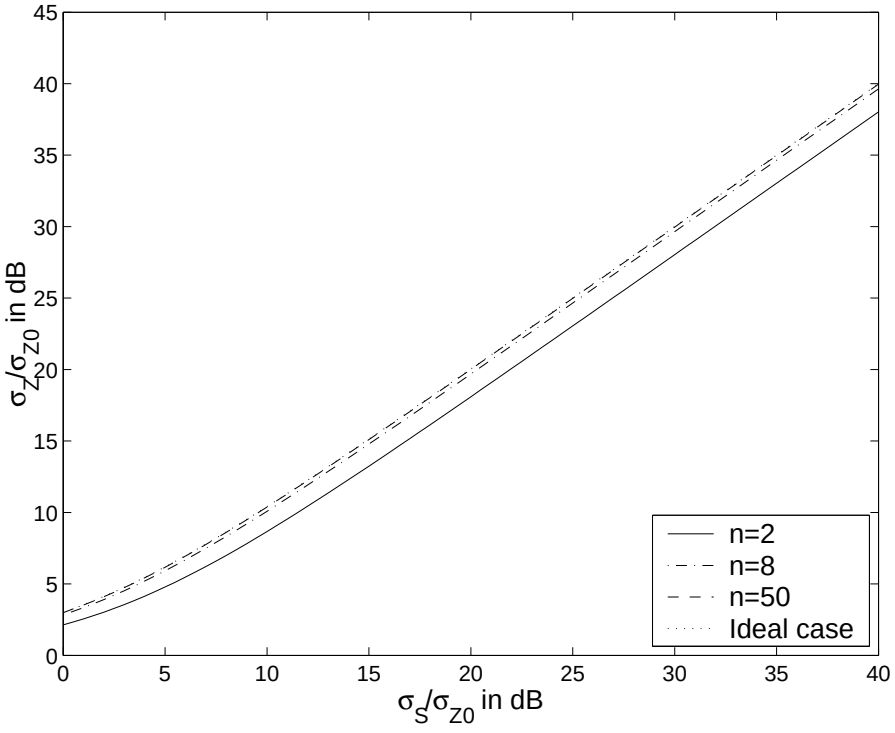

Fig. 31. The figure shows the robustness gain against the (host) signal to (attack) noise ratio for $n = 2, 8, 50$ when the host signal is Gaussian distributed. An ideal case with no host signal interference is also shown for comparison.

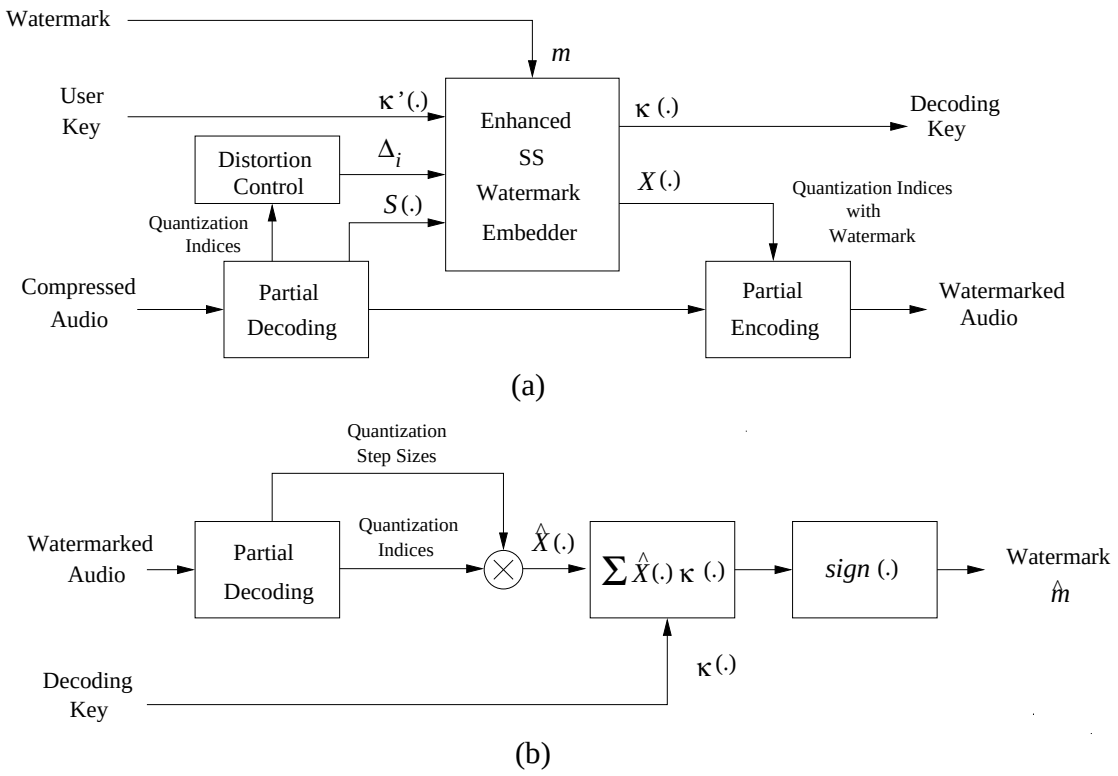

Fig. 32. Block diagrams of our proposed AAC watermarking system: (a) Encoding; (b) Decoding.

2. AAC Audio Error Concealment

Effective transmission of digital audio over noisy channels is a challenging task for researchers. Although channel coding can be used to protect the audio from network errors, it usually requires extra payload. Since most users are only concerned with the perceptual quality of the received audio, error free transmission is not necessary. Therefore, error concealment [109] [95] [20], which typically extracts features from the received audio and uses them to recover the lost data, is very attractive in audio transmission as it improves the perceptual quality without the need of additional payload.

There are two issues in error concealment: complexity of the receiver and inaccurate extraction of enhancement features at the decoder. Both can be addressed by extracting the features at the encoder and transmitting them to the decoder along with the audio. However, this method has the same disadvantage as using channel coding in that an extra payload is also required. This extra payload not only uses up more bandwidth, but necessarily modifies the audio format if neither a common area nor a user data area is available. This format change makes the audio no longer decodable by an ordinary decoder.

In this work, we apply data hiding technique to embed these enhancement features for error concealment of MPEG-2 AAC audio. Specifically, a novel modulo watermarking technique is deployed in our scheme. Modulo watermarking, which extracts hidden data as the modulo of the sum of a watermarked integer signal samples, is an example of one-to-many embedding schemes. In other words, several different watermarked signals can contain the same hidden data. This property gives the watermark encoder freedom in selecting a watermarked signal with small perceptual distortion.

Portions of the AAC encoded audio such as audio headers are naturally more important than the others. When the encoded audio is transmitted via a noisy channel, unequal error protection (UEP) is usually applied to ensure almost no corruption on these portions. In this work, we will assume the headers are very well protected and can be fully recovered. However, frequency coefficients, which are much less important and not protected substantially, may be lost during transmission. When this happens, we extract the enhancement features from embedded watermarks and use them for error concealment.

As will be shown afterward, the file size increase due to our watermark embedding scheme is negligible (< 0.1 %). This file size increase corresponds to a relatively small drop in SNR (< 0.7 dB) under noiseless conditions. However, under noisy network conditions, our experimental results show a consistent SNR gain of our scheme over the zero replacement and the frame duplication schemes at a packet loss ratio of 0.01, and the gain is even more conspicuous as the network conditions get worse.

a. Proposed Error Concealment Scheme

Since a coefficient is most effectively estimated by its nearest neighbors, ideally, adjacent coefficients along both time and frequency axes should not be packed together, because the sources of estimation will be lost as well when the packet is dropped. However, we do not impose this as a requirement of our scheme, because we target at overlaying our scheme on any other protecting scheme.

As coefficients inside a frequency band share similar perceptual behavior, we choose to group them together for estimation.

Denote (i, j)-band as the the j^{th} band at the i^{th} time frame and assume coefficients $c[i, l]$ in (i, j)-band are lost, where $l \in \mathcal{L}_j$; $\mathcal{L}_j$ is the index set of the j^{th} band. We estimate $c[i, l]$ as $\hat{c}_0[i, l] = 0, \hat{c}_1[i, l] = c[i-1, l], \hat{c}_2[i, l] = c[i+1, l]$, or

$\hat{c}_3[i,l] = \frac{1}{2}(c[i-1,l] + c[i+1,l])$.

For each of the four choices, we define $\tilde{m}[i,j]$ as the index that minimizes the mean square error. That is,

$$\tilde{m}[i,j] = \text{argmin}_{\tilde{m}\in\{0,1,2,3\}} \sum_{l\in\mathcal{L}_j} (c[i,l] - \hat{c}_{\tilde{m}}[i,l])^2.$$

$\tilde{m}[i,j]$ is pre-computed and embedded into the original AAC audio. Embedding $\tilde{m}[i,j]$ into the (i,j)-band itself will not work because when we need this information, the band is lost and $\tilde{m}[i,j]$ cannot be recovered as well. We split $\tilde{m}[i,j]$ into two bits and embed them separately into the two neighboring bands.

Define

$$m[i,j] = \begin{cases} 0, & \text{if } \tilde{m}[i-1,j] \in \{0,1\} \wedge \tilde{m}[i+1,j] \in \{0,2\}, \\ 1, & \text{if } \tilde{m}[i-1,j] \in \{2,3\} \wedge \tilde{m}[i+1,j] \in \{0,2\}, \\ 2, & \text{if } \tilde{m}[i-1,j] \in \{0,1\} \wedge \tilde{m}[i+1,j] \in \{1,3\}, \\ 3, & \text{if } \tilde{m}[i-1,j] \in \{2,3\} \wedge \tilde{m}[i+1,j] \in \{1,3\}. \end{cases}$$

The higher and the lower bit of $m[i,j]$ tell whether the current band is suitable for estimating the band in the next time frame ($(i+1,j)$-band) and in the last time frame ($(i-1,j)$-band), respectively.

For example, suppose the (i,j)-band is lost, from the lower bit of $m[i+1,j]$ and the higher bit of $m[i-1,j]$, we can determine whether the current band should be estimated from any of its neighbors. When it is estimated from both sides, it is scaled by $1/2$. If one of its neighbors is lost, we estimate the current band from the remaining neighbor. If both neighbors are loss, then we assume $\tilde{m}[i,j] = 0$ and replace the coefficients by zeros.

Upon deciding the enhancement information $m[i,j]$, we need to determine what

watermarking scheme should be used to embed the information. Watermarking schemes can be roughly categorized into two classes: fragile and robust watermarking. Fragile watermarking trades robustness with information embedding rate, and vice versa for robust watermarking.

Since there are two bits for each $m[i,j]$ and one $m[i,j]$ per band, the embedding rate is about $44100/1024 \times 49 \times 2 \times 2 \doteq 8$kbits/sec for a dual channel audio with sampling rate 44100 Hz. This is a very high embedding rate for robust watermark, which typically has a rate less than 10 bits/sec. Therefore, fragile watermark is the only possible choice.

One typical fragile watermarking scheme is least bit modulation (LBM). We can embed a bit into a host signal sequence by simply replacing the least significant bit of one signal sample by the embedding bit. The information embedding rate of LBM can be very high. For example, if we embed a bit into each sample of dual channel audio with sampling rate 44100 Hz, the embedding rate is up to $44100 \times 2 \doteq 80$kbits/sec in theory. However, since only the least significant bit is modified, the watermark can be removed easily by truncating the embedded bit. Fortunately, unlike dealing with copyright protection application, deliberate attacks to our watermark is not likely.

Since different signal samples may have different susceptibilities to distortion, we should adaptively select the embedding locations. However, for LBM, both the encoder and the decoder have to agree with a predefined embedding locations, because there is no side-information in telling the decoder the embedding locations. Note that it may not be a problem for some other applications in which a key is available for decoding, because the key itself can serve as the side-information. However, for the error concealment problem, it is not reasonable to require a user to provide a "key" before enhancement is performed.

To enable flexible encoding, we propose a novel fragile watermarking technique

that does not require the decoder to have the knowledge of the exact embedding locations. Let $\mathbf{a} = a_1, a_2, ...a_N$ be an arbitrary integer host signal sequence. We embed an integer $\hat{m} \in [0, M]$ by enforcing the following:

$$\sum_{j=1}^{N} a_j \equiv \hat{m} \mod M.$$

Note that LBM is a special case of modulo watermarking when $N = 1$ and $K = 2$.

There is more than one possible watermarked signal containing with the same embedded information. The encoder has the freedom of choosing locations of modifications that give a watermarked signal perceptually closest to the original. Despite that, the decoder does not really need to know these locations where modifications have been made.

One limitation in applying our fragile watermarking is that it can only be deployed after quantization, otherwise the watermark will be destroyed. Moreover, since it is very hard to embed watermark into a Huffman coded signal, we embed the enhancement features into the quantization indices, which are obtained after partial decoding. After watermarking, the modified indices will be encoded using Huffman coding with the original codebook.

With the freedom of embedding given by modulo watermarking, the question left is what indices and by how much they should be modified. Ideally, this can be done by applying perceptual modeling to the original audio. For example, if we know one coefficient can afford a distortion of 10 units and its current quantization step size is 2 unit. Then we know that we can approximately vary the corresponding index by 5 steps without affecting the quality. [10]

However, the perceptual model may not be accessible, because the file can be

[10]Linear quantization is assumed in this simplified example.

already compressed. Although we can also estimate the model from the decompressed audio, the estimation is not accurate in general. Therefore, we employ a heuristic approach as follows without using the perceptual model:

To embed $m[i,j]$ into the quantization indices $q[i,l]$ of (i,j)-band, $l \in \mathcal{L}_j$, let $\eta \equiv \sum_{l\in\mathcal{L}_j} q[i,l] - m[i,j] \mod M$, where M is the number of different values that can be embedded and hence is 4 in this case. Let's first assume $0 \leq \eta < M/2 = 2$,

1. Among all indices lie within range $[q_{\min}, q_{\max}]$, select η of them with largest magnitudes.

2. Declare embedding failure and leave indices unchanged if less than η indices can be found in step 1.

3. Subtract each of those indices by 1.

If $4 > \eta \geq 2$, replace η as $4 - \eta$ and proceed all steps except modifying the last one with addition instead of subtraction.

Since the enhancement features ($m[i,j]$) are independently stored, they are useful even when only a fraction of them is retrieved correctly. Therefore, embedding failure in the scheme is acceptable.

The lower limit $q_{\min}$ in the first step restrains modification of small value indices, because they are more probable to have high susceptibility to distortion. In particular, no distortion should be imposed on zero indices. $q_{\min}$ also serves as a design parameter in trading error free distortion with error concealment capability. As $q_{\min}$ increases, it is more probable that the embedding of $m[i,j]$ fails and leaves the indices with no distortion. However, the inaccurate $m[i,j]$ will make the error concealment process less efficient. In our experiment, $q_{\min}$ is simply set to be 1.

$q_{\max}$ is equal to the maximum possible value available in the Huffman table less

Table XIV. Percentage change in audio clip size after watermarking.

clip1	clip2	clip3	clip4	clip5	clip6	clip7
0.02%	0.02%	0.06%	0.01%	0.03%	0.06%	0.06%

1. This prevents indices from being out of bound after modification. Large indices are selected for modification because they can withstand a larger distortion.

b. Experimental Results

The Huffman codebook used in the original audio is optimized in the AAC encoder. Since we modify the indices but keep the old codebook, it is expected the size of the audio will increase. However, the increase is small because we only change relatively few indices. Table XIV indeed confirms this–the size increase is less than 0.1 % over all test audio clips.

In contrast, we need 8 kbits/sec if an explicit overhead is written to the audio. This corresponds to 8/256=3 % of total file size for an audio encoded at 256 kbits/sec.

In the case of no error, we expect the embedded watermark to deteriorate the audio quality. However, our test shows that the perceptual quality of the watermarked audio clips is acceptable in office or lab environment. As a objective measure, we compare the SNR difference of each AAC coded audio clip before and after the watermarking. The SNR decrease due to watermarking is between 0.03 dB and 0.68 dB (Table XV).

We assume the AAC audio coefficients are packetized and transmitted via a noisy channel. Each packet consists of coefficients from one time frame. Packet is either correctly received or lost. A periodic packet loss is assumed in our simulation with a fixed packet loss ratio. We compare our scheme with two reference schemes (Ref.1

Table XV. SNR change (in dB) after embedding enhancement information.

	AAC audio	After watermarking	SNR changes
clip1	32.87	32.69	0.18
clip2	18.18	17.95	0.23
clip3	17.13	17.10	0.03
clip4	31.50	31.29	0.21
clip5	28.66	27.99	0.67
clip6	24.47	23.79	0.68
clip7	26.73	26.69	0.04

and Ref.2). In Ref.1, all lost coefficients are set to 0, In Ref.2, the previous adjacent time frame is copied to the current lost one (Table XVI).

Our enhanced audio results in higher SNR than the control audio in all cases. The slight drop in SNR due to watermark embedding is quickly exceeded by the gain obtained from our enhancement even at a small error rate of 0.01. Moreover, the gain is more conspicuous as the error rate increases.

Table XVI. SNR comparison (in dB) of three different error concealment schemes: our scheme (upper), zero replacement scheme (middle), blindly duplication from previous time frame (lower).

		Packet loss ratio				
		0.01	0.02	0.05	0.1	0.2
clip1	Ours	22.79	20.99	15.80	13.25	9.91
	Ref.1	20.92	16.99	12.90	10.01	6.74
	Ref.2	18.60	15.06	10.63	7.61	4.24
clip2	Ours	16.93	15.94	13.92	11.80	9.49
	Ref.1	16.01	14.56	11.87	9.47	6.82
	Ref.2	15.02	13.01	9.90	7.20	4.42
clip3	Ours	16.12	15.23	13.06	11.16	8.65
	Ref.1	15.73	14.39	11.81	9.50	6.87
	Ref.2	14.41	12.49	9.36	6.71	3.92
clip4	Ours	23.74	19.62	15.27	12.42	9.55
	Ref.1	20.64	17.37	12.88	9.99	6.98
	Ref.2	17.18	14.22	10.15	7.25	4.09
clip5	Ours	23.93	21.20	14.91	12.63	9.30
	Ref.1	22.17	18.75	12.73	10.35	6.92
	Ref.2	19.35	15.08	10.13	7.67	4.53
clip6	Ours	20.73	18.82	16.81	13.62	10.59
	Ref.1	19.99	17.06	13.17	10.57	7.19
	Ref.2	16.73	14.19	9.18	6.61	3.19
clip7	Ours	23.33	21.10	15.19	13.26	9.87
	Ref.1	20.07	17.46	12.16	9.97	7.05
	Ref.2	18.82	15.87	8.59	6.26	3.36

CHAPTER V

CONCLUSION

A. Summary

Source coding and channel coding are two main components in point-to-point communications. Although side information naturally exists in many scenarios, the effect of side information is not taken into account in the conventional setups. While side information can be given to the encoder and/or decoder and thus yields several different cases, two problems that worth particular attention are source coding with side information at the decoder (Wyner-Ziv coding) and channel coding with side information at the encoder (Gel'fand-Pinsker coding) since they require completely different design strategies from the conventional source and channel coding problems.

In Chapter II, we briefly review the theories of WZC and GPC and describe a new result regarding sucessive refinement for WZC. Although the theoretical limits of WZC and GPC are known in the literatures, they cannot be obtained in close forms in general. For problems with discrete alphabets, we present an iterative algorithm in computing the theoretical limits of coding problems with side information in general.

In Chapter III, we discuss issues in WZC design. We split our discussion into the two cases when the distortion of the reconstructed source is zero and when it is not. We review that the first case, which is commonly called SWC, can be implemented using conventional channel coding. Then, we detail the SWC design using the low-density parity-check (LDPC) code. To facilitate efficient SWC design, a necessary requirement is that the SWC performance is needed to be independent of the input source. We show that a sufficient condition of this requirement is that the hypothetical channel between the source and the side information satisfies a symme-

try condition dubbed *dual symmetry*. Moreover, when dual symmetry is satisfied, the LDPC code performance over the hypothetical channel precisely translates to the SWC performance. Therefore, under that dual symmetry condition, SWC design problem can be simply treated as LDPC coding design over the hypothetical channel.

When the distortion of the reconstructed source is non-zero, we propose a practical WZC paradigm called Slepian-Wolf coded quantization (SWCQ) by combining SWC and nested lattice quantization. We point out an interesting analogy between SWCQ and entropy coded quantization in classic source coding. Furthermore, a practical scheme of SWCQ using 1-D nested lattice quantization and LDPC is implemeted, where detail design issues are discussed.

In Chapter IV, we focus on the design of GPC. However, GPC is a rather general problem that actual design procedure relies on the more precise setting of the problem. We choose to investigate the design of GPC as the form of a digital watermarking problem since digital watermarking is the precise dual of WZC as is shown in Section IIB. Although the nested coding approach described in Section IIB is applicable in theory for the digital watermarking problem, a common scaling attack can easily destroy the watermark generated by nested coding. Hence, we instead introduce an improved version of the well-known spread spectrum watermarking technique. Finally, two applications related to digital watermarking are depicted.

B. Future Directions

Our proposed SWCQ paradigm performs very well for Gaussian WZC problem. Actually, it is only 0.5 dB away from the Wyner-Ziv limit if LDPC code based SWC and trellis coded quantization are used [108]. However, this good performance is restricted to the case when the source statistics is well-defined and known. In classic source

coding, there exists the so-called universal source coding schemes such as Lempel-Ziv coding [114], which do not require the knowledge of the precise statistics of the source. It is an interesting direction to search for "universal Wyner-Ziv coding" that is defined in a similar manner.

In our digital watermarking design, we discard nested coding as an option since it is not robust against scaling attack. However, nested coding is supposed to be efficient for other applications such as broadcast channel coding. Efficient design of GPC using nested code is another difficult but rewarding research direction.

REFERENCES

[1] "MPEG. Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s, part 3: Audio. International Standard IS 11172-3," ISO/IEC JTC1/SC29 WG11, 1992.

[2] "MPEG. Information technology–generic coding of moving pictures and associated audio, part 3: Audio. International Standard IS 13818-3," ISO/IEC JTC1/SC29 WG11, 1994.

[3] "MPEG. MPEG-2 advanced audio coding, AAC. International Standard IS 13818-7," ISO/IEC JTC1/SC29 WG11, 1997.

[4] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. DCC'02*, Snowbird, UT, Mar. 2002.

[5] R. Ahlswede and J. Körner, "Source coding with side information and a converse for the degraded broadcast channel," *IEEE Trans. Inform. Theory*, vol. 21, no. 6, pp. 629–637, Nov. 1975.

[6] T. Ancheta, "Syndrome source coding and its universal generalization," *IEEE Trans. Inform. Theory*, vol. 22, pp. 432–436, July 1976.

[7] S. Arimoto, "An algorithm for calculating the capacity of an arbitrary discrete memoryless channel," *IEEE Trans. Inform. Theory*, vol. 18, pp. 14–20, Jan. 1972.

[8] M. Arnold and S. Kanka, "MP3 robust audio watermarking," in *DFG VIIDII Watermarking Workshop*, Erlangen, Germany, 1999.

[9] J. Bajcsy and P. Mitran, "Coding for the slepian-wolf problem with turbo codes," in *Proc. GlobeCom'01*, San Antonio, TX, Nov. 2001.

[10] R. J. Barron, B. Chen, and G. W. Wornell, "The duality between information embedding and source coding with side information and some applications," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1159–1180, May 2003.

[11] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes," in *Proc. IEEE International Conference on Communications*, Geneva, Switzerland, May 1993.

[13] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inform. Theory*, vol. 18, pp. 460–472, July 1972.

[14] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa, "ISO/IEC MPEG-2 advanced audio coding," *Journal of the Audio Engineering Society*, vol. 45, no. 10, pp. 789–814, Oct. 1997.

[15] K. Brandenburg and H. Popp, "An introduction to MPEG Layer-3," [Online]. Available: http://citeseer.ist.psu.com/brandenburg00introduction.html, accessed on 22 July 2004.

[16] K. Brandenburg and T. Sporer, "NMR and masking flag: Evaluation of quality using perceptual criteria," in *Proc. 11th inter. AES Conf.*, Portland, OR, May 1992.

[17] G. Caire, S. Shamai, and S. Verdu, "Lossless data compression with error correcting codes," in *Proc. ISIT'03*, Yokohama, Japan, July 2003.

[18] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1423–1443, May 2001.

[19] B. Chen and G. Wornell, "Achievable performance of digital watermarking systems," in *Proc. ICMCS99*, Florence, Italy, July 1999.

[20] Y. L. Chen and B. S. Chen, "Model-based multirate representation of speech signals and its application to recovery of missing speech packets," *IEEE Trans. Speech and Audio Processing*, vol. 5, pp. 220–231, May 1997.

[21] J. Chou, S. Pradhan, and K. Ramchandran, "Turbo and trellis-based constructions for source coding with side information," in *Proc. DCC'03*, Snowbird, UT, Mar. 2003.

[22] S.-Y. Chung, D. Forney, T. J. Richardson, and R. L. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Comm. Letters*, vol. 5, pp. 58–60, Feb. 2001.

[23] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.

[24] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Trans. Communications*, vol. 32, pp. 396–402, Apr. 1984.

[25] A. S. Cohen and A. Lapidoth, "The Gaussian watermarking game," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1639–1667, June 2002.

[26] T. Coleman, A. Lee, M. Medard, and M. Effros, "On some new approaches to practical Slepian-Wolf compression inspired by channel coding," in *Proc. DCC'04*, Snowbird, UT, Mar. 2004.

[27] J. Conway, E. Rains, and N. Sloane, "On the existence of similar sublattices," *Canad. J. Math.*, vol. 51, pp. 1300–1306, 1999.

[28] J. Conway and N. Sloane, "A lower bound on the average error of vector quantizers," *IEEE Trans. Inform. Theory*, vol. 31, pp. 106–109, Jan. 1985.

[29] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inform. Theory*, vol. 29, pp. 439–441, May 1983.

[30] T. M. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[31] T. M. Cover, "Broadcast channels," *IEEE Trans. on Inform. Theory*, vol. 18, no. 1, pp. 2–14, 1972.

[32] T. M. Cover and M. Chiang, "Duality between channel capacity and rate distortion with two-sided state information," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1629–1638, June 2002.

[33] T. Cover, "Comments on broadcast channels," *IEEE Trans. on Inform. Theory*, vol. 44, no. 6, pp. 2524–30, 1998.

[34] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, 1995.

[35] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proc. IEEE*, vol. 87, pp. 1127–1141, 1999.

[36] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedures," *Statistics and Decisions*, Supplement Issue 1, pp. 205-237, 1984.

[37] L. A. Dalton, "Analysis of 1-D nested lattice quantization and Slepian-Wolf coding for Wyner-Ziv coding of i.i.d. sources," Texas A&M University, College Station, TX, Tech. Rep., 2003.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. Ser. B*, vol. 39, pp. 1–22, 1977.

[39] J. J. Eggers, J. K. Su, and B. Girod, "A blind watermarking scheme based on structured codebooks," in *IEE Coll.: Secure Images and Image Authentication*, London, UK, Apr. 2000.

[40] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory*, vol. 37, pp. 269–275, Mar. 1991.

[41] U. Erez and S. ten Brink, "Approaching the dirty paper limit for canceling known interference," in *Proc. of the 41st Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, 2003.

[42] P. Gács and J. Körner, "Common information is far less than mutual information," *PCIT*, vol. 2, pp. 149–162, 1973.

[43] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21–28, Jan. 1962.

[44] ——, *Low-Density Parity-Check Codes.* Cambridge, MA: MIT Press, 1963.

[45] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Comm. Letters*, pp. 417–419, Oct. 2001.

[46] ——, "Compression of binary memoryless sources using punctured turbo codes," *IEEE Comm. Letters*, pp. 394–396, Sept. 2002.

[47] S. I. Gel'fand and M. S. Pinsker, "Coding for channel with random parameters," *Problems of Control and Information Theory*, no. 1, pp. 19–31, 1980.

[48] R. Gray and D. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2382, Oct. 1998.

[49] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, pp. 1079–1107, July 1999.

[50] C. Heegard and A. A. E. Gamal, "On the capacity of computer memory with defects," *IEEE Trans. Inform. Theory*, vol. 29, pp. 731–739, Sept. 1983.

[51] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 23, pp. 371–377, May 1977.

[52] D. Kahn, *The Codebreakers—The Story of Secret Writing.* New York: Scribner, 1996.

[53] D. Kirovski and H. Malvar, "Spread-spectrum watermarking of audio signals," *IEEE Trans. Signal Processing*, vol. 51, pp. 1020–1033, Apr. 2003.

[54] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[55] A. Kusnetsov and B. Tsybakov, "Coding in a memory with defective cells," *Probl. Pered. Inform.*, vol. 10, no. 2, pp. 52–60, 1974.

[56] Z. Liu, S. Cheng, A. Liveris, and Z. Xiong, "Slepian-Wolf coded nested quantization (SWC-NQ) for Wyner-Ziv coding: Performance analysis and code design," in *Proc. DCC'04*, Snowbird, UT, Mar. 2004.

[57] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, pp. 440–442, Oct. 2002.

[58] ——, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," in *Proc. DCC'03*, Snowbird, UT, Mar. 2003.

[59] D. J. C. Mackay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[60] B. MacMillan, "The basic theorems of information theory," *Ann. Math. Statist.*, vol. 24, pp. 196–219, June 1953.

[61] H. S. Malvar, *Signal Processing with Lapped Transforms.* Boston, MA: Artech House, 1992.

[62] H. S. Malvar and D. A. Florencio, "An improved spread spectrum technique for robust watermark," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, May 2002.

[63] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Communications*, vol. 38, pp. 82–93, Jan. 1990.

[64] Y. Matsunaga and H. Yamamoto, "A coding theorem for lossy data compression by ldpc codes," *IEEE Trans. Inform. Theory*, vol. 49, pp. 2225–2229, Sept. 2003.

[65] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Inform. Theory*, vol. 49, pp. 563–593, Mar. 2003.

[66] C. Neubauer and J. Herre, "Audio watermarking of MPEG-2 AAC bit streams," in *108th AES Convention*, Paris, Feb. 2000, preprint 5101.

[67] C. Neubauer, R. Kulessa, and J. Herre, "A compatible family of bitstream watermarking schemes," in *110th AES Convention*, Amsterdam, Netherlands, May 2001, preprint 5346.

[68] T. Painter and A. S. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, pp. 451–513, Apr. 2000.

[69] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. Boston: McGraw-Hill, 2002.

[70] J. Pearl, *Probability Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, California: Morgan Kaufmann Publishers, Inc., 1988.

[71] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding–a survey," *Proc. IEEE*, vol. 87, pp. 1062–1078, July 1999.

[72] T. Philosof, "Precoding for interference cancellation at low SNR," Masters thesis, Tel-Aviv University, Tel-Aviv, Israel, Jan. 2003.

[73] T. Philosof, U. Erez, and R. Zamir, "Combined shaping and precoding for interference cancellation at low SNR," in *Proc. ISIT'03*, Yokohama, Japan, July 2003.

[74] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread-spectrum communications - a tutorial." *IEEE Trans. Comm.*, vol. 30, pp. 855–884, 1982.

[75] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," in *Proc. DCC'99*, Snowbird, UT, Mar. 1999.

[76] ——, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inform. Theory*, vol. 49, pp. 626–643, Mar. 2003.

[77] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1181–1203, May 2003.

[78] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51–60, 2002.

[79] D. Rebollo-Monedero, R. Zhang, and B. Girod, "Design of optimal quantizers for distributed source coding," in *Proc. DCC'03*, Snowbird, UT, Mar. 2003.

[80] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[81] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[82] A. Roumy, S. Guemghar, G. Caire, and S. Verdu, "Design methods for irregular repeat-accumulate codes," to appear in *IEEE Trans. Inform. Theory*, 2004.

[83] K. Sayood, *Introduction to Data Compression.* San Francisco: Morgan Kaufmann Publishers, 1996.

[84] D. Schonberg, S. Pradhan, and K. Ramchandran, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," in *Proc. DCC'04*, Snowbird, UT, Mar. 2004.

[85] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, July 1948.

[86] ——, "Channels with side information at the transmitter," *IBM J. Res. & Dev.*, vol. 2, pp. 289–293, 1958.

[87] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, July 1973.

[88] V. Stankovic, A. Liveris, Z. Xiong, and C. Georghiades, "Design of Slepian-Wolf codes by channel code partitioning," in *Proc. DCC'04*, Snowbird, UT, Mar. 2004.

[89] Y. Steinberg and N. Merhav, "On successive refinement for the Wyner-Ziv problem," *IEEE Trans. Inform. Theory*, vol. 50, pp. 1636–1654, Aug. 2004.

[90] D. R. Stinson, *Cryptography: Theory and Practice.* Boca Raton: CRC Press, 1995.

[91] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE*, vol. 86, pp. 1064–1087, June 1998.

[92] M. R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sept. 1981.

[93] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Boston: Kluwer, 2001.

[94] S. Vembu and S. Verdú, "The source-channel separation theorem revisited," *IEEE Trans. Inform. Theory*, vol. 41, pp. 44–54, Jan. 1995.

[95] B. W. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the internet," in *Proc. Int'l Symposium on Multimedia Software Engineering*, Taipei, Taiwan, Dec. 2000.

[96] X. Wang and M. Orchard, "Design of trellis codes for source coding with side information at the decoder," in *Proc. DCC'01*, Snowbird, UT, Mar. 2001.

[97] E. Weiss, "Compression and coding," *IRE Trans. Inform. Theory*, vol. 8, pp. 256–257, Apr. 1962.

[98] F. Willems, Y. Shtarkov, and T. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inform. Theory*, vol. 41, pp. 653–664, May 1995.

[99] A. Wyner, "The common information of two dependent random variables," *IEEE Trans. Inform. Theory*, vol. 21, pp. 163–179, Mar. 1975.

[100] ——, "On source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 21, pp. 294–300, May 1975.

[101] A. D. Wyner, "A theorem on the entropy of certain binary sequences and applications II," *IEEE Trans. Inform. Theory*, vol. 19, pp. 772–777, Nov. 1973.

[102] ——, "The rate-distortion function for source coding with side information at the decoder–II: general sources," *Inform. and Control*, vol. 38, pp. 60–80, July 1978.

[103] A. D. Wyner and J. Ziv, "A theorem on the entropy of certain binary sequences and applications I," *IEEE Trans. Inform. Theory*, vol. 19, pp. 769–772, Nov. 1973.

[104] ——, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, pp. 1–10, Jan. 1976.

[105] A. D. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. 20, pp. 2–10, Jan. 1974.

[106] Z. Xiong, A. Liveris, S. Cheng, and Z. Liu, "Nested quantization and Slepian-Wolf coding: A Wyner-Ziv coding paradigm for i.i.d. sources," in *Proc. IEEE Workshop on Statistical Signal Processing*, St. Louis, MO, Sept. 2003.

[107] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding: Channel coding for compression," to appear in *IEEE Signal Processing Magazine*, Sept. 2004.

[108] Y. Yang, S. Cheng, Z. Xiong, and W. Zhao, "Wyner-Ziv coding based on TCQ and LDPC codes," in *Proc. 37th Asilomar Conf.*, Pacific Grove, CA, 2003.

[109] P. Yin, H. Yu, and B. Liu, "Error concealment using data hiding," in *Proc. IEEE ICASSP'01*, Salt Lake City, Utah, May 2001.

[110] W. Yu, A. Sutivong, D. Julian, T. M. Cover, and M. Chiang, "Writing on colored paper," in *ISIT 2001*, Washington, DC, June 2001.

[111] R. W. Yueng, *A First Course in Information Theory.* New York: Kluwer Academic / Plenum Publishers, 2002.

[112] R. Zamir and S. Shamai, "Nested linear/lattice codes for Wyner-Ziv encoding," in *Proc. IEEE Inform. Theory Workshop*, Killarney, Ireland, June 1998.

[113] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1250–1276, June 2002.

[114] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, vol. 23, pp. 337–343, May 1977.

VITA

Sze Ming Cheng (Samuel) received the B.S. degree in electrical and electronic engineering from the University of Hong Kong in 1995, and the M.S. degree in physics and the M.S. degree in electrical engineering from the Hong Kong University of Science and Technology and the University of Hawaii, Honolulu, in 1997 and 2000, respectively.

The typist for this thesis was Sze Ming Cheng (Samuel).