Data Compression Conference

Distributed multi-stage coding of correlated sources *

Ankur Saxena and Kenneth Rose
Dept. of Electrical & Computer Engineering
University of California, Santa Barbara, CA 93106, USA
Email:{ankur, rose}@ece.ucsb.edu

Abstract

This paper considers the problem of distributed scalable coding of correlated sources that are communicated to a central unit. The general setting is typically encountered in sensor networks. The conditions of communication channels between the sensor sources and fusion center may be time-varying and it is often desirable to guarantee a base layer of coarse information during channel fades. Specifically, we consider a multi-stage coding system to perform such distributed scalable coding of correlated sources. This problem poses new challenges. We show that mere extensions of distributed coding ideas to include multi-stage coding yield poor rate-distortion performance, due to underlying conflicts between the objectives of scalable and distributed coding. An appropriate system paradigm is developed which allows such tradeoffs to be explicitly controlled within joint optimization of all the system components. We propose an iterative joint design technique and derive the necessary conditions for optimality which yield its update rules. Simulation results show substantial gains over single source (separate) multi-stage coding as well as naive extensions to incorporate scalability in distributed scalable coding schemes.

1 Introduction

Distributed source coding (DSC)[1, 2] has witnessed a significant revival of interest since the late nineties, with a growing focus on practical code design. The work of Pradhan and Ramchandran [3] was a notable precursor and the field has eventually seen the emergence of various distributed coding techniques, mostly with an eye towards sensor networks (see reviews in [4, 5]). The basic setting in DSC involves multiple correlated sources (e.g., spatially distributed sensors) transmitting information to a fusion center without any inter-communication amongst themselves, as shown in Fig. 1. The main objective in DSC is to exploit inter-source correlations despite the fact that each sensor source is encoded without access to other sources. The only

*The work is supported in part by the NSF under grants IIS-0329267 and CCF-0728986, the University of California MICRO program, Applied Signal Technology Inc., Cisco Systems Inc., Dolby Laboratories Inc., and Qualcomm Inc.

1068-0314/08 $25.00 © 2008 IEEE
DOI 10.1109/DCC.2008.64

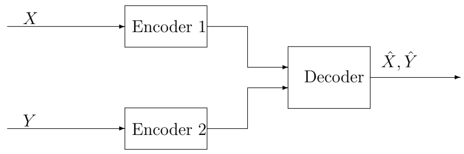

Figure 1: Distributed coding of two correlated sources

information available to a source encoder about the other sources is via joint statistics (typically extracted from training data).

The communication channels in a sensor field may vary in capacity due to the presence of obstacles or other phenomena such as fading. In such a scenario, it will be beneficial to convey a minimal amount of information even when the channel deteriorates. This motivates the problem of distributed scalable coding or distributed successive refinement, which generalizes the traditional problem of scalable coding of single source [6, 7, 8]. Successive refinement for Wyner-Ziv coding (side information at the decoder) was proposed in [9], and has been studied in [9, 10] from the information-theoretic perspective of characterizing achievable rate-distortion regions. In this paper, we derive practical iterative algorithms for the design of successive-refinable system within the multi-terminal (distributed) setting. It should be noted that distributed scalable coding is related to but differs from "robust distributed source coding" [11, 12] or multiple descriptions coding [13]. For example, base layer information is always required for reconstruction of the enhancement layer(s).

Various scalability structures may be implemented, such as tree-structured quantizers or multi-stage quantizers [14]. In practice, multi-stage structures are often preferred due to their reduced complexity and training data requirements. An example is speech coding applications where multi-stage vector quantizers are heavily used. In this work as well, we will primarily focus on the design of distributed multi-stage coding (DMSC) schemes.

DMSC can be considered as a generalization of traditional distributed source coding or traditional scalable coding. It may be tempting to assume that simple combination of algorithms for distributed coding ([15, 16, 12]) and multi-stage quantizer design ([14]), would yield a good DMSC coding scheme. However, as we will see, there exists a fundamental tradeoff between exploiting inter-source correlation at the base or intermediate layers, and better reconstruction in subsequent layers of the DMSC. Moreover, by allowing for a slight but controlled mismatch between encoder and decoder estimates and reconstructions, inter-source correlation can be exploited more effectively.

The rest of the paper is organized as follows. In Sec. 2, we state the problem formally, introduce notation and specify the components of the DMSC system. In Sec. 3, we discuss a naive design that simply combines distributed coding and multi-stage quantizer design algorithms. Sec. 4 describes the proposed iterative algorithm for DMSC system design, along with necessary conditions for optimality which determine its update rules. Simulation results are summarized in Sec. 5, followed by the conclusions in Sec. 6.

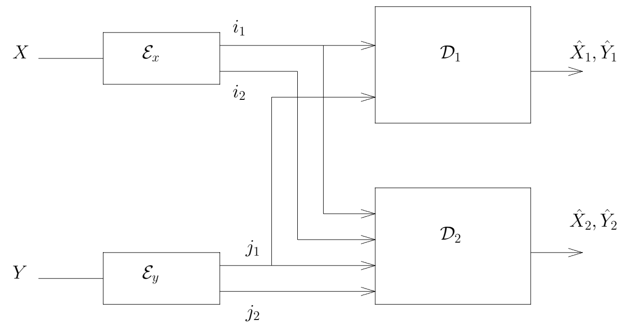

Figure 2: Distributed multi-stage coding

2 Problem Statement and System Paradigm

Consider the successively refinable distributed coding scenario in Fig. 2. For brevity, we will restrict the analysis to the case of two sources and to two-layers, but without loss of generality since the model is trivially extendible to an arbitrary number of sources or layers. Here (X, Y) are two continuous amplitude, i.i.d., correlated (scalar or vector) sources. The encoder $\mathcal{E}_x$ for source X compresses the data and transmits an index pair $\{i_1, i_2\}$ where $i_1 \in \{1..2^{R_{1x}}\}$ and $i_2 \in \{1..2^{R_{2x}}\}$. Similarly the encoder $\mathcal{E}_y$ for Y has an index pair $\{j_1, j_2\}$ as output where $j_1 \in \{1..2^{R_{1y}}\}$ and $j_2 \in \{1..2^{R_{2y}}\}$. R_{1x} and R_{1y} correspond to the first (base) layer rates while R_{2x} and R_{2y} denote the incremental second (enhancement) layer rates. We assume that the fusion center obtains full information from the base layer while data from the enhancement layer is lost with probability $p \in [0, 1]$ (for presentation simplicity, we assume enhancement layer information from both sources is lost simultaneously with probability p). Depending on whether or not the enhancement layer information is lost, the fusion center uses decoder $\mathcal{D}_1$ or $\mathcal{D}_2$ to reconstruct X as $\hat{X}_1$ or $\hat{X}_2$ (and similarly $\hat{Y}_1$ or $\hat{Y}_2$). The objective of the distributed multi-stage coding problem is to minimize the following overall distortion function given rate allocations R_{1x}, R_{2x}, R_{1y} and R_{2y}:

$$\begin{aligned} D_{net} &= E[p\{\alpha d(X, \hat{X}_1) + (1-\alpha) d(Y, \hat{Y}_1)\} \\ &+ (1-p)\{\alpha d(X, \hat{X}_2) + (1-\alpha) d(Y, \hat{Y}_2)]\}, \end{aligned} \tag{1}$$

where $d(\cdot, \cdot)$ is an appropriately defined distortion measure and $\alpha \in [0, 1]$ governs the relative importance of the sources X and Y at the decoder. The first two distortion terms in (1) account for base layer distortion per source, while the last two terms cover the case where enhancement layer information is also received. Note that the above simplifying assumptions can be eliminated by simple modification of the weight factors for these terms. Nevertheless, for simplicity, we use the cost (1) throughout the paper.

Similar to the design of a multi-stage vector quantizer, we here emphasize on an iterative design algorithm for distributed multi stage coder design for correlated sources. We next explain the functioning of various components of the DMSC (distributed multi stage coding) system.

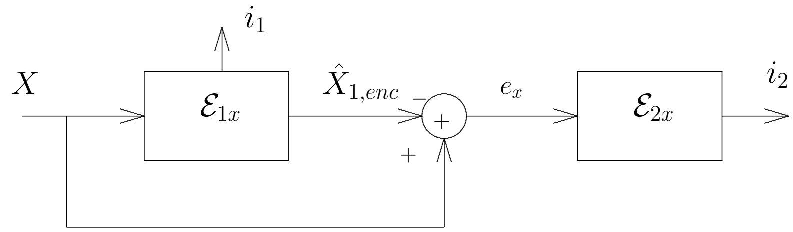

Figure 3: DMSC Encoder

2.1 Distributed Multi Stage Encoder

The DMSC encoder for source X is shown in Fig. 3. The overall encoder $\mathcal{E}_x$ consists of two stage encoders $\mathcal{E}_{1x}$ and $\mathcal{E}_{2x}$. Input X is fed to the first stage encoder $\mathcal{E}_{1x}$ whose output is an index i_1 and an *encoder* reconstruction value $\hat{X}_{1,enc}$. The residual, $e_x = X - \hat{X}_{1,enc}$ is input to the second stage encoder $\mathcal{E}_{2x}$, whose output is an index i_2. Since the sources X and Y are correlated, the encoders $\mathcal{E}_{1x}$ and $\mathcal{E}_{2x}$ will differ from the nearest-neighbor quantizers encountered in single-source multi-stage quantization. A block diagram for $\mathcal{E}_{1x}$ is shown in Fig. 4. High resolution quantization maps source X to index k_1 representing Voronoi region $C^x_{k_1}$. Next, a *lossy* mapping which we refer to as Wyner-Ziv (WZ) mapping is employed (the name loosely accounts for the fact that the scenario involves lossy coding with side information whose asymptotic performance bound was given in [2]). The WZ mapping block takes in k_1 and outputs index $i_1 = v_1(k_1)$ representing region $R^x_{i_1} = \bigcup_{k_1; v_1(k_1)=i_1} C^x_{k_1}$, to be transmitted over the channel. An example of WZ mapping for a scalar source with $\mathcal{K}_1 = 7$ and $\mathcal{I}_1 = 3$, is also given in Fig. 4.

The *encoder* codebook C_1^{-1} takes index k_1 as input and outputs $\hat{X}_{1,enc}$ which is used to compute the residual e_x. Base layer encoder $\mathcal{E}_{1y}$ for source Y is defined similarly. Since the error residuals e_x and e_y obtained by the first encoding stage are correlated, a distributed coder should be designed to exploit inter-source correlations. The second stage encoders $\mathcal{E}_{2x}$ and $\mathcal{E}_{2y}$ similarly consist of a high rate quantizer followed by WZ mapping. Since the second stage is the last stage in our setting here, no encoder codebook is needed in $\mathcal{E}_{2x}$ or $\mathcal{E}_{2y}$ (in general all except the last DMSC stage encoders contain an encoder codebook as in $\mathcal{E}_{1x}$).

2.2 Distributed Multi Stage Decoder

The DMSC base and enhancement layer decoders for source X are depicted in Fig. 5. Decoder $\mathcal{D}_{1x}$ takes indices i_1 and j_1 from the first layer of the sources to reconstruct $\hat{X}_1$ while $\mathcal{D}_{2x}$ reconstructs $\hat{X}_2$ based on i_1,i_2,j_1 and j_2. Decoder $\mathcal{D}_{1x}$ simply consists of codebook C_2^{-1} as shown, while $\mathcal{D}_{2x}$ consists of codebooks C_3^{-1} and C_4^{-1} which output $\hat{X}_{1,dec}$ and $\hat{e}_x$, respectively. Note that $\hat{X}_1$, $\hat{X}_{1,enc}$ and $\hat{X}_{1,dec}$ differ in general. In brief, these entities can be interpreted as follows:

1. $\hat{X}_1$ is constructed using i_1 and j_1 to minimize the distortion of the base layer reconstruction.

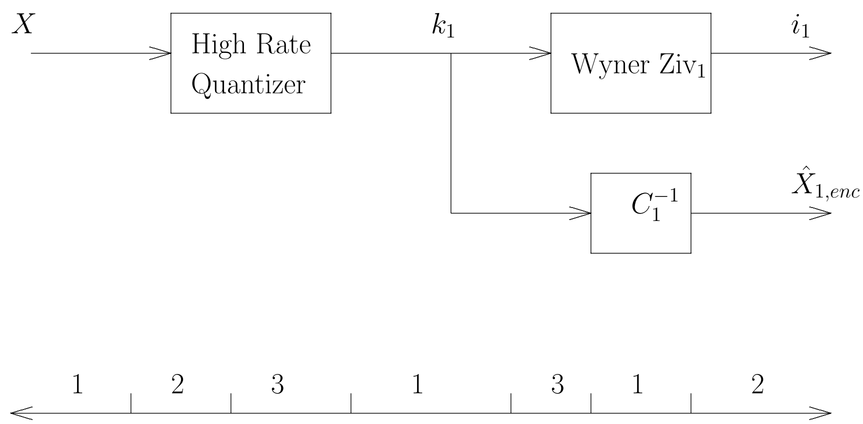

Figure 4: DMSC base layer encoder and an example of Wyner Ziv mapping from Voronoi regions to (transmitted) indices

2. $\hat{X}_{1,dec}$ is also based on i_1 and j_1 and its sole objective is to aid the second layer reconstruction, $\hat{X}_2$.

3. $\hat{X}_{1,enc}$, based on k_1 is an encoder estimate of X at the base layer in order to derive the residual for the enhancement layer.

For source Y, we have similar decoders $\mathcal{D}_{1y}$ and $\mathcal{D}_{2y}$. Note that the base layer decoder $\mathcal{D}_1$ of Fig.2, actually contains $\mathcal{D}_{1x}$ and $\mathcal{D}_{1y}$. Similarly, $\mathcal{D}_2$ contains $\mathcal{D}_{2x}$ and $\mathcal{D}_{2y}$.

2.3 Components to Optimize

Distributed multi-stage coding design optimizes the high rate quantizers, WZ mappings, encoder and decoder codebooks for all layers and all sources. We will restrict the scope here to the design of all codebooks and WZ mappings. (For simplicity, we will assume that high rate quantizers are independently designed using standard Lloyd's algorithm [17]. Additional gains due to their joint optimization with the rest of the system are expected to be small).

3 Naive Design Scheme

We first discuss the design scheme which emerges when distributed coding is directly combined with multi-stage coding. As it ignores the potential conflict in objectives we refer to it as "naive" design. In the naive scheme, a first-layer distributed coder is designed to minimize the distortion for the base layer using a distributed coder design algorithm such as in [12], while ignoring the enhancement layer and the role of p. Consequently, the estimates $\hat{X}_{1,enc}$ and $\hat{X}_{1,dec}$ are calculated only based on index i_1. (Note that there is no encoder-decoder mismatch in this scheme and $\hat{X}_{1,enc}(i_1) = \hat{X}_{1,dec}(i_1) = E[X|X \in R^x_{i_1}])$. The residual e_x is calculated as $e_x = X - \hat{X}_{1,enc}$ and

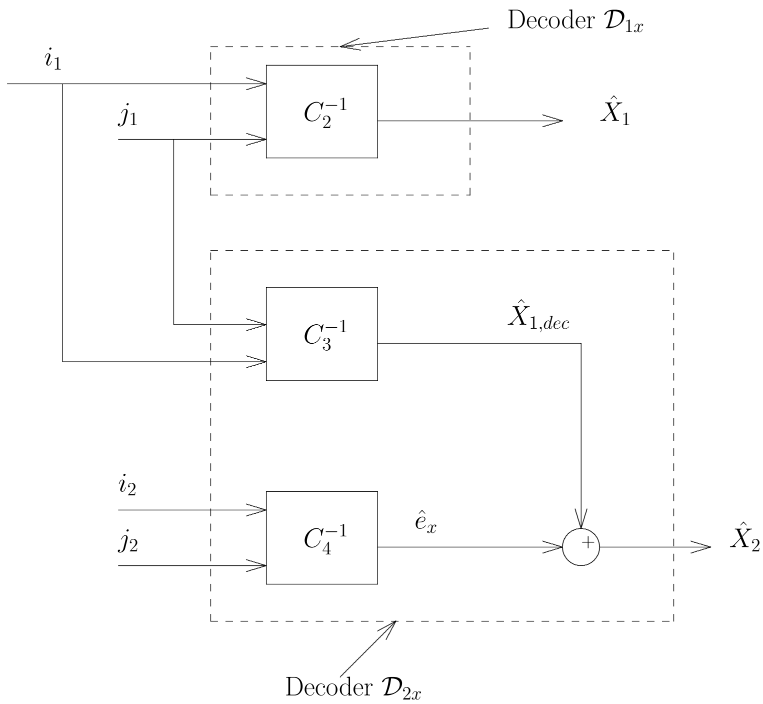

Figure 5: DMSC decoder

similarly for e_y. The resulting training set for $\{e_x, e_y\}$ is used to design a distributed coder for the enhancement layer to minimize the enhancement layer distortion given the fixed base layer coder. Note that the naive scheme will approach optimality only at high probability of loss at the enhancement layer (in which case, only the base layer is of interest).

4 Distributed Multi Stage Coding Algorithm

4.1 Motivation and Design

The most fundamental deviation of this work from the "natural" approach to DMSC is in the use of different codebooks for constructing $\hat{X}_{1,enc}$ and $\hat{X}_{1,dec}$. At the decoder, both indices i_1 and j_1 can be utilized to construct $\hat{X}_{1,dec}$. However, the encoder for source X only has access to index i_1 to construct $\hat{X}_{1,enc}$, and does not know j_1. Obviously, there will be a mismatch between $\hat{X}_{1,enc}$ and $\hat{X}_{1,dec}$. A possible way to match $\hat{X}_{1,dec}$ with $\hat{X}_{1,enc}$ will be to make $\hat{X}_{1,dec}$ a function of i_1 alone, but this will defeat the purpose of distributed coding (utilizing inter-source correlation) in the first layer.

The idea, is therefore to allow for some mismatch between the first layer estimates at the encoder and decoder and optimize so that efficient distributed coding at first layer will more than compensate for any allowed mismatch. Another crucial point to note is that, the source encoder has complete knowledge of the source itself or effectively index k_1 (which is the output of the high resolution quantizer used primarily to discretize the source), while the decoder has additional knowledge from the correlated source Y, in the form of index j_1. This implies that there may exist some (elusive) additional information at both ends that could be exploited, if an appropriate means were devised.

We therefore use different codebooks for calculating $\hat{X}_{1,dec}$ and $\hat{X}_{1,enc}$ at the decoder versus encoder. The encoder codebook (C_1^{-1}) can have k_1 as input, and the *decoder helper* codebook C_3^{-1} has inputs i_1 and j_1. This flexibility enables optimization of the tradeoff between better exploitation of inter-source correlations, and the cost of some mismatch in the system. Appropriate design of encoder and decoder codebooks (as well as WZ mappings) will optimize the precise overall performance while accounting for the mismatch.

Note that the scheme subsumes single source multi-stage quantizer design as a special case. Also, when the sources X and Y are uncorrelated, then WZ mappings for the base layer will converge to a union of contiguous cells (the encoder $\mathcal{E}_{1x}$ will act as a fine-coarse quantizer) and both the encoder and decoder codebooks will effectively be the same and depend on i_1 only.

4.2 Update Rules

Herein we assume mean-squared error distortion for simplicity. The notation in what follows is heavy due to the multiple indexing involved. But in a nutshell, we alternate between optimization of the various codebooks and WZ mappings at different layers while fixing all other parameters.

The following necessary conditions for optimality determine the update rules in terms of the subset of distortion terms to be minimized (i.e., those that depend on the parameters being updated) while avoiding detailed notation.

1. **First Layer Decoder Codebook** (C_2^{-1}): Entry (i_1, j_1), $i_1 = 1 : \mathcal{I}_1$ and $j_1 = 1 : \mathcal{J}_1$ is obtained by minimizing

$$\hat{x}_1(i_1, j_1) = \arg\min_{\phi} \sum_{(x,y)\in R_{i_1}\times R_{j_1}} d(x, \phi). \tag{2}$$

2. **Second Layer Decoder Codebook** (C_4^{-1}) (for residuals): Entry (i_2, j_2), $i_2 = 1 : \mathcal{I}_2$ and $j_2 = 1 : \mathcal{J}_2$ is obtained by minimizing

$$\hat{e}_x(i_2, j_2) = \arg\min_{\phi} \sum_{(e_x,e_y)\in R_{i_2}\times R_{j_2}} d(x, \hat{x}_{1,dec} + \phi). \tag{3}$$

3. **Encoder Codebook** (C_1^{-1}): Entry k_1, $k_1 = 1 : \mathcal{K}_1$ is given by minimizing:

$$\hat{x}_{1,enc}(k_1) = \arg\min_{\phi} \sum_{x\in C_{k_1}} d(x, \hat{x}_{1,dec} + \hat{e}_x), \tag{4}$$

where the dependence on ϕ comes from $\hat{e}_x$, which is the reconstructed value of e_x at the second layer and $e_x = x - \phi$.

4. **First Layer Helper Decoder Codebook** (C_3^{-1}): Entry (i_1, j_1), $i_1 = 1 : \mathcal{I}_1$ and $j_1 = 1 : \mathcal{J}_1$ is obtained by minimizing

$$\hat{x}_{1,dec}(i_1, j_1) = \arg\min_{\psi} \sum_{(x,y) \in R_{i_1} \times R_{j_1}} d(x, \hat{e}_x + \psi). \tag{5}$$

5. **WZ Mappings (Layer 2)**: For $k_2 = 1 : \mathcal{K}_2$, assign region k_2 to index $i_2 = v_2(k_2)$ such that:

$$v_2(k_2) = \arg\min_{i_2 \in \{1..I_2\}} \sum_{e_x \in C_{k_2}} d(x, \hat{x}_{1,dec} + \hat{e}_x(i_2, j_2)). \tag{6}$$

6. **WZ Mappings (Layer 1)**: For $k_1 = 1 : \mathcal{K}_1$, assign region k_1 to index $i_1 = v_1(k_1)$ such that:

$$v_1(k_1) = \arg\min_{i_1 \in \{1..I_1\}} \sum_{x \in C_{k_1}} [pd(x, \hat{x}_1(i_1, j_1)) + (1-p)d(x, \hat{x}_{1,dec}(i_1, j_1) + \hat{e}_x(i_2, j_2))]. \tag{7}$$

The update rules for the second source Y are straightforward to specify from the above. Also, to reduce clutter, superscripts and arguments were omitted where obvious, e.g., R_{i_1} rather than $R_{i_1}^x$; $\hat{e}_x$ rather than $\hat{e}_x(i_2, j_2)$.

5 Simulation Results

Two examples are provided to demonstrate the gains of the proposed distributed multi-stage coding. In our simulations, the sources X and Y are assumed to be jointly gaussian with zero means, unit variances and correlation coefficient $\rho = 0.95$. A training set of 5000 scalars is generated. Simulation results are depicted in Fig. 6. In the simulations, the weighting coefficient of (1) is set to $\alpha = 0.5$ so that equal importance is given to both sources at the decoder. The number of prototypes is 60 for the high rate quantizers which are designed using Lloyd's algorithm [17].

In the first experiment, the same transmission rate is allocated to each layer of each source, i.e., $R_{1x} = R_{2x} = R_{1y} = R_{2y} = R$. The probability of enhancement layer loss is $p = 0.2$. The weighted distortion (1) at the decoder is plotted versus the rate R. We compare: (a) non-distributed multi-stage coding, i.e., each source is compressed using a standard multi-stage scalar quantizer; (b) naive distributed multi-stage coding, and (c) proposed distributed multi-stage coding (DMSC). DMSC clearly outperforms the other compression schemes and gains of up to ~ 2.2 dB are achieved (e.g., at rate of 2 bits). Note that the naive distributed scalable coding underperforms separate scalable coding, which is evidence for the significance of the underlying conflict between objectives. These must be explicitly resolved as is indeed done by the proposed approach.

In the second experiment, the transmission rates for the different layers and sources are all fixed at 2 bits/sample. The weighted distortion is plotted vs. the probability of enhancement layer loss, p. Here when p is high, the naive scheme performs better than separate multi-stage coding since it simply minimizes the expected weighted distortion at the base layer before proceeding to the enhancement layer. Note that in the naive scheme, p only plays the role of a weighing factor in computing the total distortion (1). The enhancement layer distortion in naive scheme is considerably higher than the base layer distortion (since scalability is ignored in order to do efficient distributed coding at base layer) and this is an evidence of the inherent conflict between distributed and scalable coding. Hence, the performance of naive scheme approaches DMSC only for higher values of p. However, the DMSC scheme consistently outperforms both the naive scheme and separate multi-stage coding for all p. It should also be mentioned that the various algorithms were run multiple times to mitigate concerns about susceptibility to local minimum traps depending on initialization. (Optimization using global variants of the technique is beyond the scope of this paper).

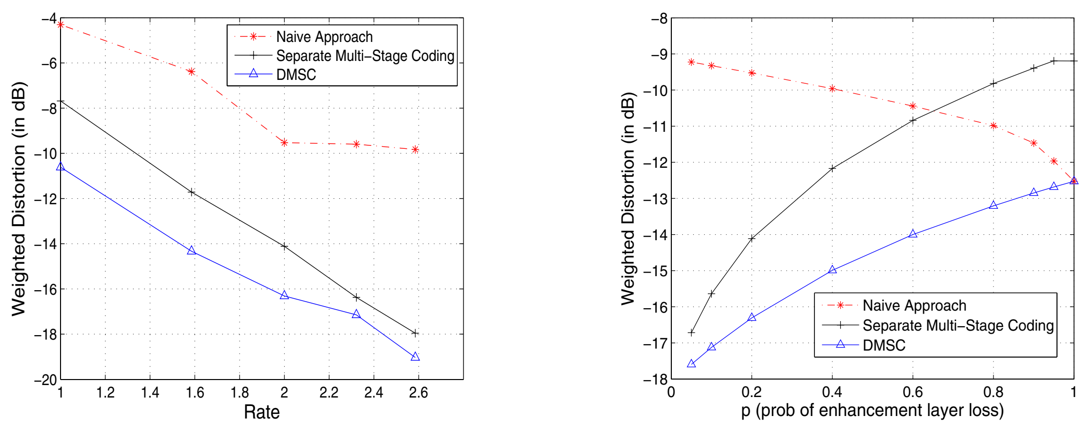

Figure 6: Performance comparison: distributed multi stage coding (DMSC), separate, non-distributed multi-stage coding and naive design scheme. In the left figure, the x-axis represents transmission rates ($R_{1x} = R_{2x} = R_{1y} = R_{2y} = R$); while in the right figure, the x-axis denotes the probability of enhancement layer loss

6 Conclusions

In this paper, we have proposed an iterative algorithm for the design of multi-stage distributed coders. Our scheme allows a controlled mismatch between the encoder and decoder reconstruction for estimating the enhancement layer residual and jointly optimizes all the components in the DMSC system. Simulation results show that the DMSC scheme consistently outperforms other naive schemes and single source (separate) distributed multi stage coding schemes.

References

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Information Theory*, vol. 19, no. 4, pp. 471–480, Jul 1973.

[2] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side-information at the decoder," *IEEE Trans. on Information Theory*, vol. 22, pp. 1–10, Jan 1976.

[3] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction." *IEEE Trans. on Information Theory*, vol. 49, no. 3, pp. 626–643, Mar 2003.

[4] 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, Mar 2002.

[5] Z. Xiong, A. Liveris, and S.Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, Sep 2004.

[6] B. Rimoldi, "Successive refinement of information: characterization of the achievable rates ," *IEEE Trans. on Information Theory*, vol. 40, no. 1, pp. 253–259, Jan 1994.

[7] W. Equitz and T. M. Cover, "Successive refinement of information ," *IEEE Trans. on Information Theory*, vol. 37, no. 2, pp. 269–275, Nov 1991.

[8] V. Koshelev, "An evaluation of the average distortion for discrete schemes of sequential approximation," *Probl. Pered. Inform.*, vol. 17, no. 3, pp. 20–33, Jan 1981.

[9] Y. Steinberg and N. Merhav, "On successive refinement for the wyner-ziv problem," *IEEE Trans. on Information Theory*, vol. 50, no. 8, pp. 1636–1654, Aug 2004.

[10] C. Tian and S. Diggavi, "Multistage successive refinement for Wyner-Ziv source coding with degraded side informations," in *IEEE ISIT*, July 2006, pp. 1594–1598.

[11] J. Chen and T. Berger, "Robust distributed source coding," *to appear in IEEE Trans. on Information Theory.*

[12] A. Saxena, J. Nayak, and K. Rose, "On efficient quantizer design for robust distributed source coding," in *IEEE DCC*, Mar 2006, pp. 63–72.

[13] L.Ozarow, "Source-coding problem with two channels and three receivers," *Bell Sys. Tech. J.*, vol. 59, no. 10, pp. 1909–1921, 1980.

[14] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.

[15] M. Fleming, Q. Zhao, and M. Effros, "Network vector quantization," *IEEE Trans. on Information Theory*, vol. 50, no. 8, pp. 1584–1604, Aug 2004.

[16] D. Rebollo-Monedero, R. Zhang, and B. Girod, "Design of optimal quantizers for distributed source coding," in *IEEE DCC*, Mar 2003, pp. 13–22.

[17] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar 1982.