ON DISTRIBUTED QUANTIZATION IN SCALABLE AND PREDICTIVE CODING

Ankur Saxena and Kenneth Rose

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

ABSTRACT

The study of distributed coding system design for correlated sources has largely focused on the basic problem, with the implicit assumption that practically motivated extensions involving scalable or predictive coding may be obtained in a straightforward manner. In this paper, we show that naively extended approaches yield poor rate-distortion performance. In fact, inherent conflicts arise between distributed quantization and scalable or predictive coding. Distributed predictive coding is further plagued by design instability of closed loop predictors, whose impact has been recognized in the context of single source coding, but is greatly exacerbated in the case of distributed coding. We propose a general framework that allows for and controls mismatch between encoder and decoder estimates in the base-layer or prediction loop, respectively, in distributed scalable or predictive coding. Simulation results show substantial gains over single source (separate) scalable or predictive coding techniques, as well as over naive extensions to incorporate scalability or prediction in distributed coding.

***Index Terms*—** Distributed quantization, predictive coding, scalable coding.

1. INTRODUCTION

Distributed source coding (DSC) [1, 2] has witnessed a significant revival of interest with sensor networks as a prominent application area. Review papers, e.g., [3] and [4] describe a variety of strategies for distributed coding system design. However, most existing work is focused on distributed coding of memoryless sources, and neglects the temporal correlations of typical sources. Moreover, while scalable distributed coding has been considered theoretically and asymptotically [5, 6], its practical code design has not received much attention. The implicit assumption may have been that scalable (or predictive) coding per se is largely a solved problem in practice, and combining distributed coding with such compression paradigms would require a straightforward integration phase.

In this paper, we review two different coding strategies: distributed multi-stage coding (DMSC) and distributed predictive coding (DPC) from our previous work [7, 8, 9]. 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 (see e.g., [10, 11]) and multi-stage quantizer design ([12]), 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 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., Qualcomm Inc., and Sony Ericsson, Inc.

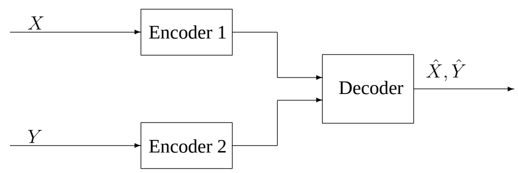

Fig. 1. Distributed coding of two correlated sources

Similarly the generalization from DSC to DPC is highly non-trivial due to conflicting objectives of distributed coding versus efficient prediction in DPC. In other words, optimal distributed coding (in terms of current reconstruction quality) may severely compromise the prediction loop at each source encoder. We review a controlled drift approach wherein the encoder and decoder estimates of the source reconstruction are allowed to be different such that the benefits in improved prediction overwhelm potential drift.

Another design difficulty whose origins are in standard predictive quantizer design [12] is exacerbated in the distributed setting. On the one hand, open-loop design is simple and stable but the quantizer is mismatched with the true prediction error statistics (as the system eventually operates in closed loop). On the other hand, if a distributed quantizer is designed in closed-loop, the effects of quantizer modifications are unpredictable as quantization errors are fed back through the prediction loop and can build up. Hence the procedure is unstable and may not converge. The effect is greatly exacerbated in the case of DPC. To circumvent these difficulties we use the technique of asymptotic closed loop (ACL) {[13, 14]} which we rederive for DPC system design. Within the ACL framework, the design is effectively in open-loop within iterations (eliminating issues of error buildup through the prediction loop), while ensuring that asymptotically, the prediction error statistics converge to closed loop statistics. In other words, the prediction loop is essentially closed asymptotically.

2. DISTRIBUTED SCALABLE CODING

Consider the scalable distributed coding scenario of Fig. 2. We restrict the analysis to the case of two sources and to two-layers for brevity, but without loss of generality. 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

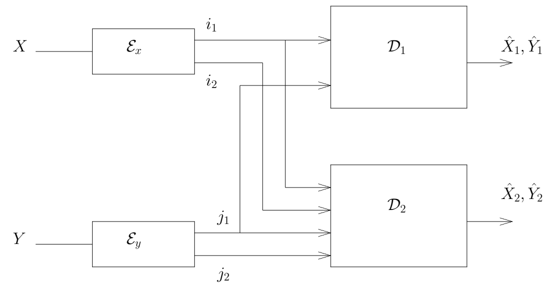

Fig. 2. Distributed multi-stage coding

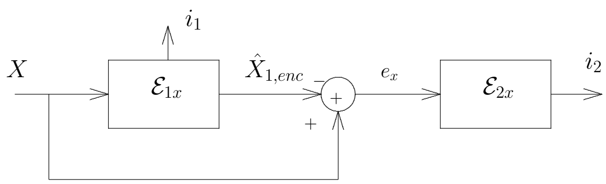

Fig. 3. DMSC Encoder

pair $\{i_1, i_2\}$ where $i_1 \in \{1..2^{R_{1x}}\}$ and $i_2 \in \{1..2^{R_{2x}}\}$. Similar notation is employed for source Y. R_{1x} and R_{1y} are the base layer rates, while R_{2x} and R_{2y} denote the incremental 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 $q \in [0, 1]$. 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 is to minimize the following overall distortion function given rate allocations R_{1x}, R_{2x}, R_{1y} and R_{2y}:

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

where $d(\cdot, \cdot)$ is a distortion measure and $\alpha \in [0, 1]$ governs the relative importance of the sources X and Y at the decoder. We next explain the functioning of various components of the DMSC (distributed multi stage coding) system.

2.1. Distributed Multi Stage Encoder

The DMSC encoder for source X is shown in Fig. 3. 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 second stage encoder $\mathcal{E}_{2x}$ has residual $e_x = X - \hat{X}_{1,enc}$ as input and outputs index i_2. To exploit inter-source correlation, the encoders $\mathcal{E}_{1x}$ and $\mathcal{E}_{2x}$ may 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 a Voronoi cell in the input space. 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 representing a union of Voronoi cells, 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.

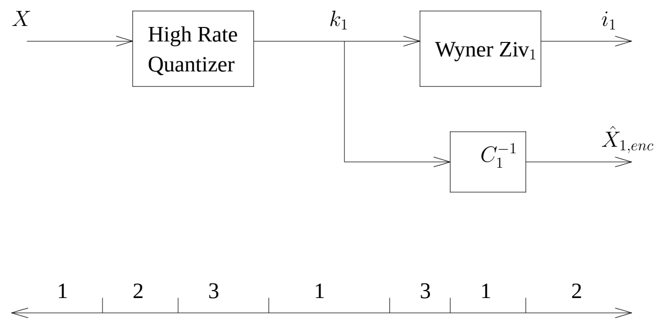

Fig. 4. DMSC base layer encoder and an example of Wyner Ziv mapping from Voronoi regions to (transmitted) indices

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. The error residuals e_x and e_y obtained by the first encoding stage are correlated as well and 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 $\mathcal{D}_{1x}$ and $\mathcal{D}_{2x}$ for source X are depicted in Fig. 5. Decoder $\mathcal{D}_{1x}$ takes indices i_1 and j_1 from base layer 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. The entities $\hat{X}_1$, $\hat{X}_{1,enc}$ and $\hat{X}_{1,dec}$ differ in general and 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.
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 $\mathcal{D}_{2y}$.

3. DISTRIBUTED PREDICTIVE CODING

Consider the generic distributed coding scenario (Fig.1), where in addition to inter-source correlation, X and Y also have memory. The objective is to exploit both the temporal and inter-source correlations to minimize the following expected distortion:

$$D_{net} = E\{\alpha d(X, \hat{X}) + (1-\alpha) d(Y, \hat{Y})\}, \quad (2)$$

where $\alpha \in [0, 1]$ is a weighing factor. We employ linear prediction to exploit temporal redundancies within the sources. The prediction errors e_x (for X) and e_y (for Y) (to be defined in following subsection) are likely correlated. Hence, instead of the standard predictive quantizer, a distributed quantizer needs to be designed to exploit

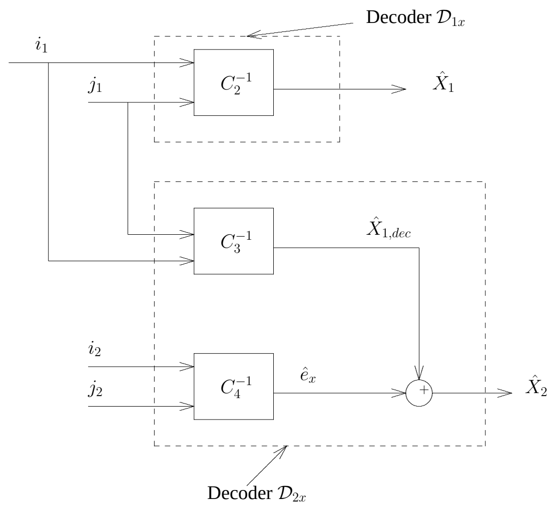

Fig. 5. DMSC decoder

inter-source correlations. We next describe the DPC controlled-drift approach.

3.1. Distributed Predictive Encoder

The encoder for distributed predictive coding is shown in Fig.6. The error residual is $e_x = X - \tilde{X}_{enc}$, where $\tilde{X}_{enc}$ is the predicted value of X. Encoder codebook C_k^{-1} is used to obtain $\hat{e}_{x,enc}$, the error reconstruction at the encoder. $\hat{X}_{enc}$ denotes the source reconstruction at the encoder and P_x denotes the linear predictor.

3.2. Distributed Predictive Decoder

The DPC decoder is depicted in Fig.7. Here codebooks $C_{ij,loop}^{-1}$ and $C_{ij,dec}^{-1}$ denote the loop and decoder codebooks respectively and aid in reconstruction of $\hat{X}_{dec}$ at the decoder. Note that $\hat{X}_{enc}$ and $\hat{X}_{loop}$ are, in general, different from $\hat{X}_{dec}$.

4. DESIGN STRATEGIES FOR DMSC AND DPC SYSTEMS

Here we outline the design strategies for a DMSC (the DPC counterpart will appear parenthetically) system. The most fundamental observation to be made in DMSC (DPC) design is to use different codebooks for constructing $\hat{X}_{1,enc}$ and $\hat{X}_{1,dec}$ ($\hat{e}_{x,enc}$ and $\hat{e}_{x,loop}$ in DPC). At the decoder, both indices i_1 and j_1 can be utilized to construct $\hat{X}_{1,dec}$ (i, j for reconstructing $\hat{e}_{x,loop}$ in DPC). However, the encoder for source X only has access to index i_1 (or i) to construct $\hat{X}_{1,enc}$ (or $\hat{e}_{x,enc}$), and does not know j_1 (or j). Obviously, there will be a mismatch between $\hat{X}_{1,enc}$ and $\hat{X}_{1,dec}$ (or $\hat{e}_{x,enc}$ and $\hat{e}_{x,loop}$). A possible way to match $\hat{X}_{1,dec}$ with $\hat{X}_{1,enc}$ (or $\hat{e}_{x,enc}$ with $\hat{e}_{x,loop}$) will be to make $\hat{X}_{1,dec}$ (or $\hat{e}_{x,loop}$) a function of i_1 (or i) alone. Such a strategy is referred to as the naive scheme in DMSC (or zero-drift scheme in DPC), and is analyzed in [9, 7]). Unfortunately, it defeats the main objective of efficiently exploiting inter-source correlations in the base layer of DMSC (or distributed coding for prediction errors in DPC).

The idea, is therefore to allow for some mismatch between the first layer estimates in DMSC (prediction error estimates in DPC)

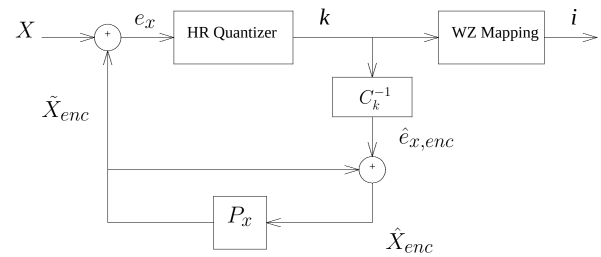

Fig. 6. Controlled-Drift DPC encoder

at the encoder and decoder, and jointly optimize so that efficient distributed coding 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, while the decoder has additional knowledge from the correlated source Y, in the form of index j_1 (or j). This implies that there may exist some (elusive) overlapping 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 in a DMSC system. The encoder codebook (C_1^{-1}) has 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. We refer the reader to [9] for further details on a locally optimal iterative technique along with necessary rules for optimality (update rules) for DMSC coder design. Similarly in DPC, we use different encoder and loop codebooks to efficiently utilize both inter-source and temporal correlation.

Also note that in DPC, the quantized error sample $\hat{e}_{x,enc}$ at time n impacts $\tilde{X}_{enc}$ and $\hat{X}_{dec}$ from time $n+1$ onwards due to the presence of prediction loop. On the other hand, $\hat{e}_{x,dec}$ at time n only impacts $\hat{X}_{dec}$ at time n. Hence, if one tries to directly design a *distributed quantizer* for the quantities being quantized, namely, the pair of prediction errors $\{e_x, e_y\}$, while ignoring effects on the prediction loop, i.e., to minimize the following distortion:

$$E[\alpha d(e_x, \hat{e}_{x,dec}) + (1-\alpha) d(e_y, \hat{e}_{y,dec})] \quad (3)$$

(see e.g., DSC in [10]), the ultimate distortion in (2) will not be minimized.

It is evident that there are conflicting design objectives for the distributed quantizer in terms of current reconstruction versus prediction performance. This motivates the need for an asymptotic closed-loop (ACL) approach [13, 14], that allows the design iteration to be performed in open-loop, but with essential closing of the loop asymptotically. For a complete discussion on DPC system design algorithms via ACL, the reader is referred to [7, 8].

5. SIMULATION RESULTS

The simulation results are shown in Fig. 8. In both the experiments, a training size of 5000 scalars is used and the weighing factor α is set to 0.5. The number of prototypes is 60 for the high rate quantizers.

In the DMSC experiment, the sources are jointly gaussian with zero mean, unit variance and correlation coefficient 0.95. The probability of enhancement layer loss is $q = 0.2$. The transmission rate

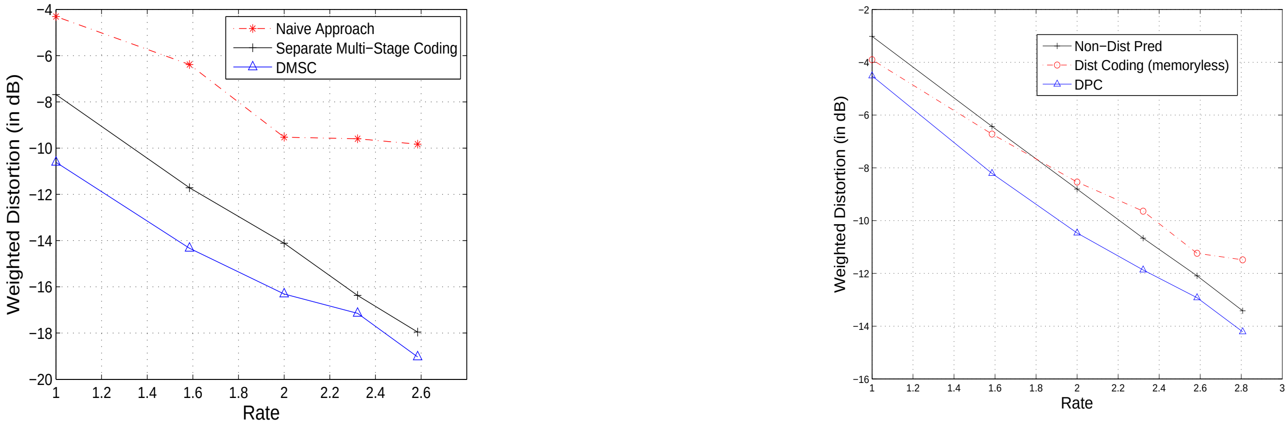

Fig. 8. Performance comparison: (a) distributed multi stage coding (DMSC); separate, non-distributed multi-stage coding; and naive design scheme (b) controlled-drift distributed predictive coding (DPC); separate, non distributed predictive coding; memoryless distributed coding

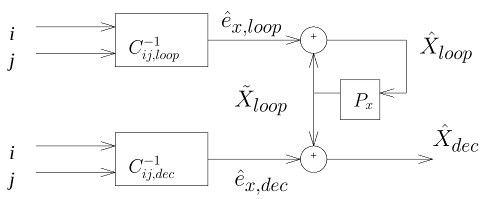

Fig. 7. Controlled-Drift DPC decoder

is uniformly distributed across all layers and sources. The weighted distortion (1) at the decoder is plotted versus the rate R. Three different strategies are compared: (i) non-distributed multi-stage coding, i.e., each source is compressed using a standard multi-stage scalar quantizer; (ii) naive distributed multi-stage coding; and (iii) proposed distributed multi-stage coding (DMSC). DMSC clearly outperforms the other compression schemes with typical gains of ~ 2.2 dB 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.

The DPC experiment employs a Gauss-Markov source model $X_n = \beta X_{n-1} + w_n$ and $Y_n = \gamma Y_{n-1} + u_n$, where w_n, u_n are i.i.d., zero-mean, unit variance, jointly Gaussian scalar sources with correlation coefficient $\rho = 0.97$ and $\beta = \gamma = 0.8$. First order linear predictors were designed for the sources. The weighted distortion at the decoder is plotted versus rate. We compare: (i) non-distributed predictive coding, i.e., each source is compressed using standard predictive coding; (ii) memoryless distributed coding, i.e., no prediction is performed; and (iii) controlled-drift based distributed predictive coding (DPC). The DPC scheme clearly outperform the other two compression schemes and gains of $\sim$ 1.8 dB are achieved (e.g., at $R_1 = R_2 = 2$ bits/sample).

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

[4] S.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] 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.

[6] Chao Tian and Suhas Diggavi, “Multistage successive refinement for Wyner-Ziv source coding with degraded side informations,” in *IEEE ISIT*, July 2006, pp. 1594–1598.

[7] A. Saxena and K. Rose, “Distributed predictive coding for spatio-temporally correlated sources,” in *IEEE International Symposium on Information Theory*, June 2007, pp. 1506–1510.

[8] A. Saxena and K. Rose, “Challenges and recent advances in distributed predictive coding,” in *IEEE Information Theory Workshop*, Sep 2007, pp. 448–453.

[9] A. Saxena and K. Rose, “Distributed multi-stage coding of correlated sources,” in *IEEE DCC*, Mar 2008, pp. 312–321.

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

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

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

[13] H. Khalil, K. Rose, and S. L. Regunathan, “The asymptotic closed-loop approach to predictive vector quantizer design with application in video coding,” *IEEE Trans. on Image Processing*, vol. 10, no. 1, pp. 15–23, Jan 2001.

[14] H. Khalil and K. Rose, “Predictive vector quantizer design using deterministic annealing,” *IEEE Trans. on Signal Processing*, vol. 51, no. 1, pp. 244–254, Jan 2003.