Joint Entropy-Constrained Multiterminal Quantization

J. Cardinal and G. Van Assche

Abstract

As a rate-distortion extension to the Slepian-Wolf problem, we study the entropy-constrained design of a multiterminal quantizer for coding two correlated continuous sources. The designed quantizer can then be combined with a lossless encoder operating close to the Slepian-Wolf bound. Two design methods are presented, both optimizing a Lagrangian cost measure involving the distortion and the information rate. The first method is a simple descent algorithm, while the second is based on index reuse of a high-resolution quantizer. Numerical results are displayed. The soundness of the index reuse method is shown, while confirming the advantages of entropy constraints over simple entropy limitations.

I. Introduction

We study the design of a multiterminal quantizer $(\alpha_X, \alpha_Y, \beta)$ for coding of two correlated sources X and Y taking values in $\mathbb{R}^k$. We assume that the two sources are encoded separately by the encoders α_X and α_Y, and that the pair of output indices is jointly decoded by the decoder β. The reproduction values are denoted by $\hat{X}$ and $\hat{Y}$, and both corresponding distortions must be minimized. This problem is a rate-distortion generalization of the Slepian-Wolf (SW) distributed coding problem [1] and has to be distinguished from that of estimating a single source given two noisy observations X and Y of it. Following [2], we will refer to this latter problem as *remote coding* and to our problem as *direct coding*.

We further assume that the quantization step is followed by an ideal SW entropy coder $(\gamma_X, \gamma_Y, \gamma^{-1})$, as illustrated on Fig. 1. This assumption does make sense since recent works such as [3] propose practical coders operating close to the SW bounds. We propose to design a quantizer pair jointly minimizing the two distortions with constraints on the two bitrates predicted by the SW theorem.

Zamir and Berger [2] show that at the high-resolution limit there is no rate loss compared to the joint encoding of the two sources and that the optimal high-resolution performance can be achieved by the composition of "blind" separate quantizers with a SW coder. They suggest the use of lattice quantizers for the first stage. These results, however, only hold at high rates, and it can also be shown that in general there is a loss due to the separation of the encoders. We believe that this loss can be minimized by taking into account the cascaded SW coder during the quantizer design, just as entropy-constrained quantizers [4] perform better than lattice quantizers.

Fleming and Effros [5] recently proposed a unifying framework in which the fixed-rate version of the direct coding problem is treated. Pradhan and Ramchandran [6] propose a solution for the remote coding problem using linear codes. An early approach of fixed-rate quantizer design for remote coding can be found in a paper by Flynn and Gray [7].

II. Optimality Conditions

We use the notations $I = \alpha_X(X)$, $J = \alpha_Y(Y)$. The achievable rates for SW coding of these two indices satisfy:

$$R_X \geq H(I \mid J) \quad (1)$$

$$R_Y \geq H(J \mid I) \quad (2)$$

$$R_X + R_Y \geq H(I, J). \quad (3)$$

Using discrete Lagrangian optimization, we define the cost measure to be minimized by the triple $(\alpha_X, \alpha_Y, \beta)$:

$$J(\alpha_X, \alpha_Y, \beta) = D_X + \mu D_Y + \lambda_X R_X + \lambda_Y R_Y, \quad (4)$$

J. Cardinal is with the Faculté des Sciences, CP 212, Université Libre de Bruxelles, 1050 Brussels, Belgium (e-mail: jcardin@ulb.ac.be).
G. Van Assche is with the Ecole Polytechnique, CP 165, Université Libre de Bruxelles, 1050 Brussels, Belgium (e-mail: gvanassc@ulb.ac.be).

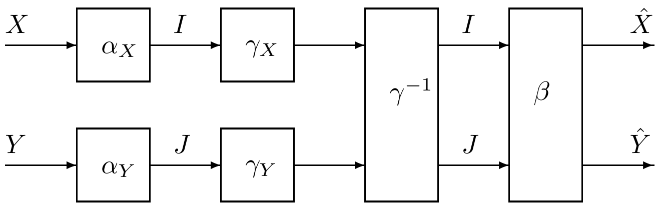

Fig. 1. Block diagram of the proposed multiterminal quantizer

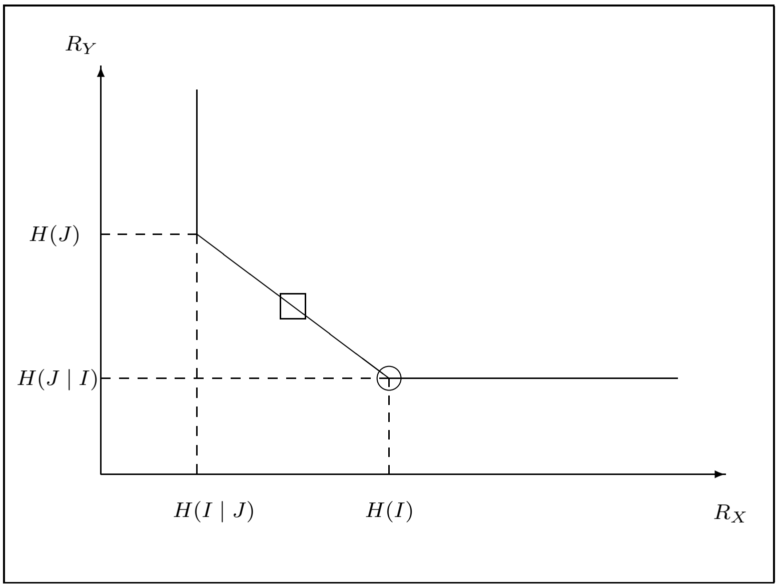

Fig. 2. The Slepian-Wolf achievable rate region and two possible choices for the definition of the entropy constraint: ○ corresponds to Eqn. (7-8) and □ to Eqn. (9-10)

where D_X and D_Y are the average distortions, R_X and R_Y the average rates, and μ, λ_X and λ_Y are positive Lagrangian multipliers.

We use the notation $\beta(i, j) = (\beta_X(i, j), \beta_Y(i, j))$. The distortions are written

$$D_X = E[d(X, \beta_X(I, J))] \quad (5)$$
$$D_Y = E[d(Y, \beta_Y(I, J))] \quad (6)$$

for a suitable distortion measure d. In the expression for J the number of rate constraints is two, while three inequalities define the achievable rate pairs. Hence one degree of freedom is left for the design of the SW coder and we have to choose an arbitrary operation point on the optimal rate pairs curve. In the following, we choose the simple definitions

$$R_X = H(I) \quad (7)$$
$$R_Y = H(J \mid I), \quad (8)$$

but nothing prevents the use of other values such as

$$R_X = \frac{H(I \mid J) + H(I)}{2} \quad (9)$$
$$R_Y = \frac{H(J \mid I) + H(J)}{2}. \quad (10)$$

One can simply design an encoder with the roles of X and Y reversed (i.e., so that $R_X = H(I \mid J)$ and $R_Y = H(J)$) and is then able to achieve any convex combination of the rates (7-8) and the reversed ones by time multiplexing.

Let us first consider the optimal encoder for X. The cost measure to minimize for α_X is derived from J by writing the distortion sum as a conditional mean and isolating the term depending on the value $I = i$ in R_X:

$$\alpha_X(x) = \arg\min_i E\left[d(x, \beta_X(i, J)) + \mu d(Y, \beta_Y(i, J)) \mid X = x\right] - \lambda_X \log P\left[I = i\right] - \lambda_Y \log P\left[J \mid I = i\right], \quad (11)$$

for all $x \in \mathbb{R}^k$, where i is taken in the set $\mathcal{I}_X$ of indices for α_X.

The optimal encoder for Y is similar, excepted that R_Y is defined differently:

$$\alpha_Y(y) = \arg\min_j E\left[d(X, \beta_X(I, j)) + \mu d(y, \beta_Y(I, j)) - \lambda_Y \log P\left[J = j \mid I\right] \mid Y = y\right], \quad (12)$$

for all $y \in \mathbb{R}^k$, where j is taken in the set $\mathcal{I}_Y$ of indices for α_Y.

The optimal decoder for the mean squared error is the classical Bayes estimator:

$$\beta(i, j) = E\left[X, Y \mid I = i \wedge J = j\right], \quad (13)$$

for all $(i,j) \in \mathcal{I}_X \times \mathcal{I}_Y$.

A simple descent algorithm for the design of $(\alpha_X, \alpha_Y, \beta)$ is as follows:
1. define α_X as in Eqn. (11) while holding α_Y and the probabilities $P[I=i]$ fixed,
2. define α_Y as in Eqn. (12) while holding α_X and the probabilities $P[J=j \mid I=i]$ fixed,
3. define β as in Eqn. (13) while holding α_X and α_Y fixed,
4. update the probabilities $P[I=i]$ and $P[J=j \mid I=i]$ while holding α_X and α_Y fixed.

The cost $J(\alpha_X, \alpha_Y, \beta)$ decreases at each iteration and is bounded, hence the algorithm converges to a local optimum.

The optimal design equations are straightforwardly derived but their implementation is not simple. The estimation of the conditional distribution of one source given another using a training set, in particular, can lead to estimation errors and instability in the iterative process. Moreover, if we restrict the scheme to scalar quantizers ($k=1$), we do not have any simple formula for the bounds of the quantization cells.

In [5], the probability density estimation problem is solved by using piecewise constant approximations of the probability distributions. This essentially reduces to operating on quantized versions of the sources. We propose a related, but simpler, approach. It is also inspired from the quantization based on index reuse proposed in [7].

III. Alternative Design

We define α_X as the composition of a primary quantizer Q_X and an index assignment (IA) function δ_X:

$$\delta_X : \mathcal{K}_X \rightarrow \mathcal{I}_X, \tag{14}$$

where $\mathcal{K}_X$ is the index set for Q_X. Similarly, we set $\alpha_Y = \delta_Y \circ Q_Y$. Note that an optimal quantizer $\alpha_.$ can be approximated with arbitrary precision by this composition for a primary quantizer $Q_.$ with sufficiently high resolution. This approach has already been applied by Flynn and Gray [7] to the remote coding problem with fixed-rate quantization.

In the following, merging two quantization cells i and i' in $\mathcal{I}_X$ consists in creating a new IA function δ'_X identical to δ_X excepted that $\delta'_X(i) = \delta'_X(i') = \delta_X(i)$. Similarly, we can merge two quantization cells $j, j' \in \mathcal{I}_Y$.

We design the IA function by iteratively merging quantization cells until the current rate is equal to the target rate. The initial IA function is the identity. We denote by $\Delta_{(i,i')}(.)$ (resp. $\Delta_{(j,j')}(.)$) the variation of the argument when i and i' (resp. j and j') are merged in δ_X (resp. δ_Y).

We define the following *marginal returns*

$$\Lambda_X(i,i') = \frac{-\Delta_{(i,i')}(D_X) - \mu\Delta_{(i,i')}(D_Y)}{\Delta_{(i,i')}(R_X)} \tag{15}$$

$$\Lambda_Y(j,j') = \frac{-\Delta_{(j,j')}(D_X) - \mu\Delta_{(j,j')}(D_Y)}{\lambda\Delta_{(j,j')}(R_Y)}, \tag{16}$$

where μ and λ are positive Lagrangian multipliers.

We can find a good sequence of mergings by choosing at each step the pair of indices minimizing the corresponding marginal return. It amounts to choosing the merging that minimizes the distortion increase per bit. A geometrical illustration is given in Fig. 3.

This approach is advantageous for several reasons. First, it can be implemented easily with an empirical knowledge of the input sources, i.e. a training set. Second, it requires less Lagrangian multipliers than in the previous approach. Finally, the method directly gives the whole rate-distortion curve while the previous algorithm was defined for a specific target rate. The drawback of the technique is that it gives no guarantee of optimality.

As noted in [7], the marginal return method allows many simplifications in the case of mean squared error. First, the centroids can be easily recomputed when cells i and i' are merged (and similarly for j and j').

$$\beta'_{(i,i')}(i,j) = \frac{P[I=i \wedge J=j]\beta(i,j) + P[I=i' \wedge J=j]\beta(i',j)}{P[I=i \wedge J=j] + P[I=i' \wedge J=j]} \tag{17}$$

Second, the increase in distortion is a simple function of the centroids and masses of probabilities before merging.

$$\Delta_{(i,i')}(D_X) = \sum_j \frac{P[I=i \wedge J=j]P[I=i' \wedge J=j]}{P[I=i \wedge J=j] + P[I=i' \wedge J=j]}(\beta_X(i,j) - \beta_X(i',j))^2 \tag{18}$$

Third, the decrease in rate can also be calculated quite simply. For this case, merging i and i' must be distinguished from merging j and j'.

$$\Delta_{(i,i')}(R_X) = f(P[I=i] + P[I=i']) - f(P[I=i]) - f(P[I=i']) \quad \text{with } f(x) = x\log_2 x \tag{19}$$

$$\Delta_{(j,j')}(R_Y) = \sum_i f(P[I=i \wedge J=j] + P[I=i \wedge J=j']) - f(P[I=i \wedge J=j]) - f(P[I=i \wedge J=j']). \tag{20}$$

Finally, two cells merged imply to recompute only the marginal returns for which one of the two merged cells indexes is involved.

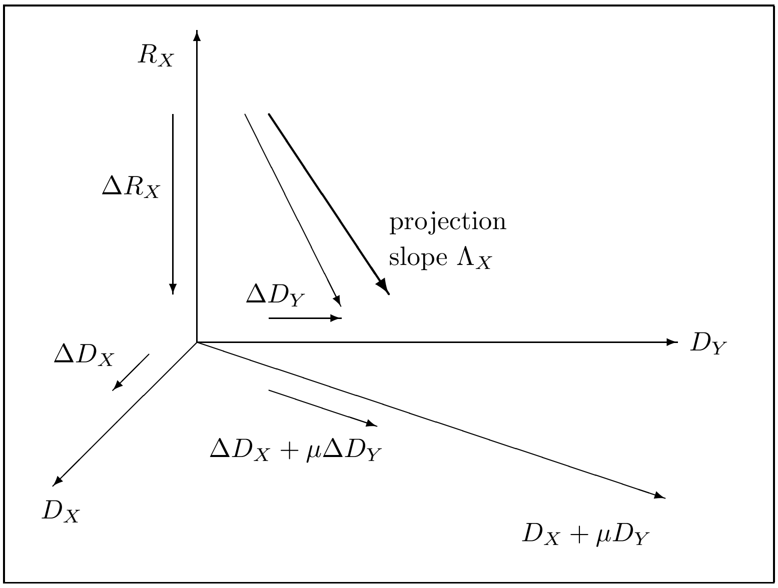

Fig. 3. Computation of the marginal return for a merging in δ_X

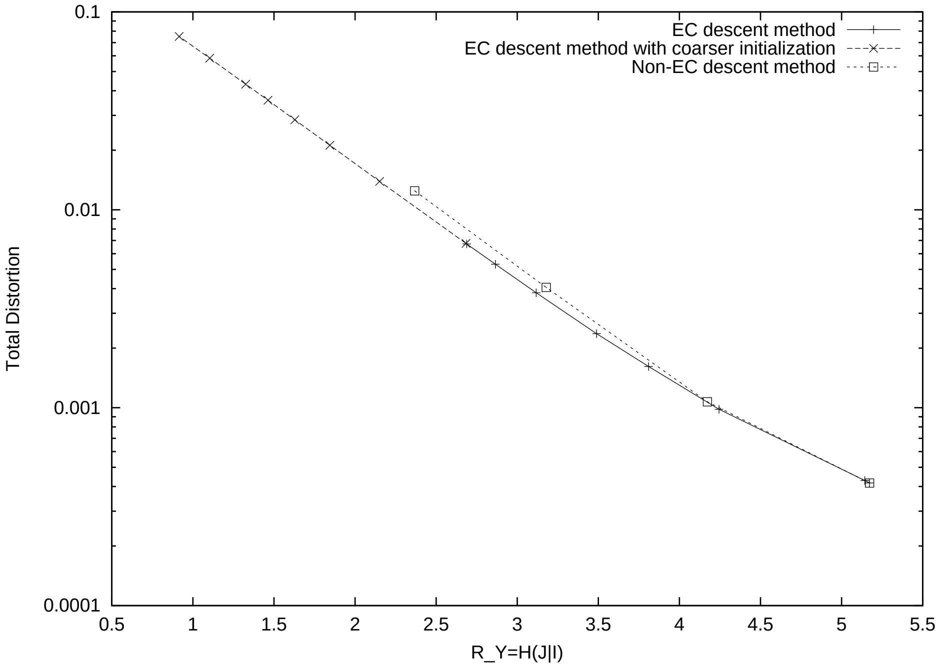

Fig. 4. The rate-distortion curves obtained with the descent method

IV. Experiments

In these experiments, we defined X and Y as scalar Gaussian sources ($k = 1$) with unit variance and correlation factor $\rho = 0.9$. The probability density function is thus

$$f_\rho(x, y) = \frac{e^{-\frac{1}{2}\frac{x^2 - 2\rho xy + y^2}{1-\rho^2}}}{2\pi\sqrt{1-\rho^2}}. \tag{21}$$

We first implemented the descent algorithm of Section II using numerical integration. We chose to train the encoder α_Y and the decoder β using Eqn. (12) with α_X defined as a uniform quantizer.

The choice of numerical integration results from the need to evaluate conditional probabilities, for which training sets are not well suited. All quantities based on (21) such as probability mass, average and distortion can be symbolically

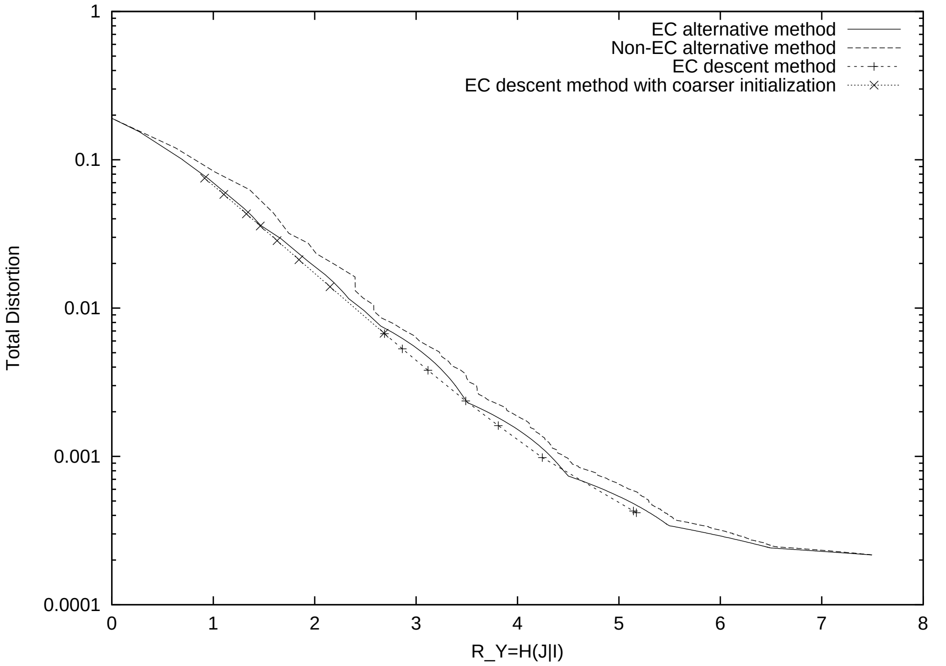

Fig. 5. The rate-distortion curves obtained with the alternative method

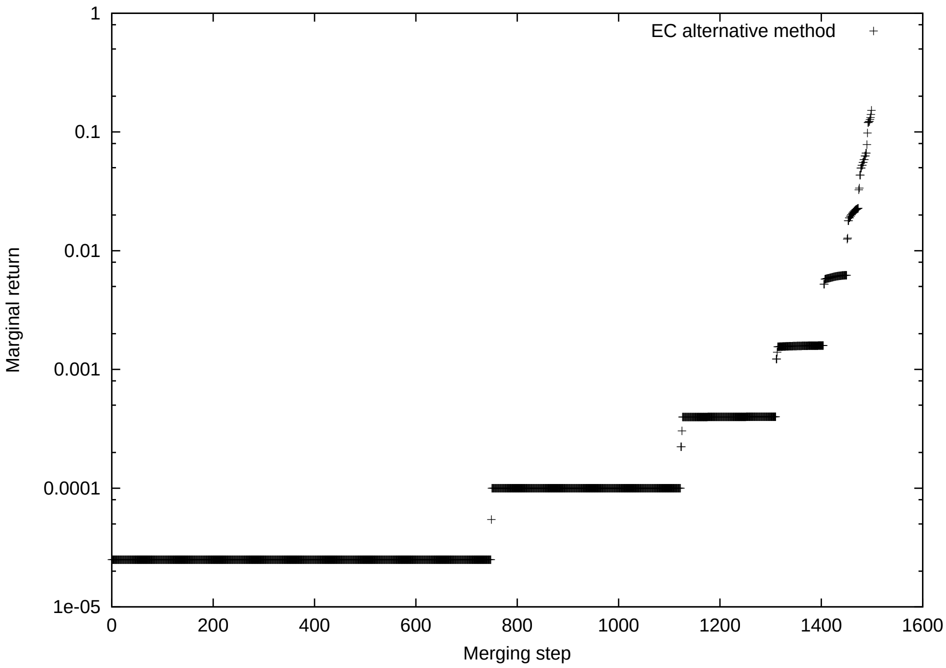

Fig. 6. Marginal return as a function of the merging step

integrated in one dimension and expressed in terms of error functions. However, the second dimension must be integrated numerically, and we used Romberg's algorithm [8].

The uniform encoder chosen for α_X is a uniform quantizer with interval length 0.05, corresponding to a rate of almost 6.37 bits. We neglected areas such that $|x| > 6$ or $|y| > 6$, for which the mass of probability is insignificant. The encoder α_Y is initialized with an encoder identical to α_X. In some cases, we initialized α_Y with a uniform quantizer with interval length 0.2 to allow quicker convergence – see Fig. 4. Eqn. (12) is evaluated for y varying in discrete steps. The difference between successive values is chosen to be significantly smaller than the length of the smallest interval. When two successive y result in two different minima j, a bisection is performed in order to find the interval bound with increased accuracy. The resulting rate-distortion pairs are show in Fig. 4, for which we fixed $\mu = 1$, $R_X \approx 6.37$ and spanned a wide range of achievable rates R_Y. *Non* entropy-constrained optimization results are also displayed in the same figure, as they turned out to be less satisfactory.

We also implemented the alternative method. Since all the necessary routines were already created, we also used numerical integration to evaluate it. Furthermore, this allows us to compare more objectively with the descent algorithm. Note, however, that the alternative method could be simply implemented with training sets as no conditional probability needs to be evaluated during the iterations.

The encoder chosen for α_X is the same as above. For α_Y, we started with a uniform encoder with thinner intervals. The chosen interval length is 0.01, thus 5 times smaller than for α_X. This makes the work of the merging algorithm sufficiently fine-grained while remaining tractable on a PC. Moreover, if training sets were used, it would require a large but yet reasonable number of training samples. The results are displayed in Fig. 5.

While the descent algorithm requires us to vary λ_Y to get several rate-distortion pairs, a single run of the merging method gives a curve of rate-distortion pairs. It is interesting to note that the merging algorithm seems to work in stages. The marginal return stays about constant for many mergings in a row, then jumps to a higher level, then stays again about constant, and so on (see Fig. 6). For rate-distortion pairs taken as the marginal return jumps from one step to another, the method is really competitive to the more computationally intensive descent algorithm. Also, it is clear that the marginal return method works best when entropy constraints are taken into account.

V. Conclusion

We described two novel multiterminal quantizer design algorithms assuming the existence of an optimal SW coder at the quantizer output. The soundness and efficiency of these algorithms have been shown experimentally on correlated scalar Gaussian sources.

Future work on this topic could include the combination of the quantizer with a practical SW coder, and the inclusion of the SW coder update in the design loop. The merging method can also be applied to vector quantizers for real sources, such as quantization of LPC coefficients in speech coding or wavelet coefficients in correlated image sources. Finally, one can easily generalize the technique to more than two correlated sources.

References

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

[2] R. Zamir and T. Berger, "Multiterminal source coding with high resolution," *IEEE Trans. Inform. Theory*, vol. 45, no. 1, pp. 106–117, Jan. 1999.

[3] Q. Zhao and M. Effros, "Optimal code design for lossless and near lossless source coding in multiple access networks," in *Proc. IEEE Data Compression Conf.*, 2001, pp. 263–272.

[4] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 1, pp. 31, 1989.

[5] M. Fleming and M. Effros, "Network vector quantization," in *Proc. IEEE Data Compression Conf.*, 2001.

[6] S. S. Pradhan and K. Ramchandran, "Distributed source coding: Symmetric rates and applications to sensor networks," in *Proc. IEEE Data Compression Conf.*, 2000, pp. 363–372.

[7] T. J. Flynn and R. M. Gray, "Encoding of correlated observations," *IEEE Trans. Inform. Theory*, pp. 773–787, Nov. 1987.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, second edition, 1992.