PRACTICAL CODES FOR LOSSY COMPRESSION WHEN SIDE INFORMATION MAY BE ABSENT

Sivagnanasundaram Ramanan and John MacLaren Walsh

Drexel University
Dept. of Electrical and Computer Engineering
3141 Chestnut Street, Philadelphia, PA 19104, USA.

ABSTRACT

Practical codes are developed for quadratic Gaussian lossy compression when side information may be absent by hybridizing successively refinable trellis coded quantization (SR-TCQ) and low-density parity-check (LDPC) codes. A 2-refinement level SR-TCQ is used to generate a common description for both decoders and an individual description for the decoder with side information. Then, the common description is losslessly compressed while the individual description is compressed using a LDPC code which exploits the side information in a belief propagation decoder. Simulation results show that the practical codes require no more than 0.46 extra bits at moderate rates and no more than 0.7 extra bit at low rates from the theoretical bounds.

Index Terms— Successive refinement, SR-TCQ, LDPC codes

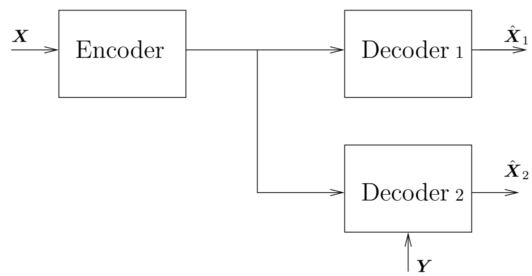

Fig. 1. Network model for the side information may be absent problem

1. INTRODUCTION

In [1], Heegard and Berger considered the class of lossy compression source coding problems "when side information may be absent". In these problems, a node needs to send a source sequence $\boldsymbol{X} = \{X(n)\}_{n=1}^{N}$ to another node which may or may not have a side information sequence $\boldsymbol{Y} = \{Y(n)\}_{n=1}^{N}$ that is correlated with the source sequence. The objective is to reconstruct the source sequence at the decoder with a distortion of D_1 when the side information is absent and with a distortion of D_2 when the side information is present. This problem can be modeled as shown in Fig. 1 with one encoder and two decoders. Heegard and Berger [1] studied the total rate required to achieve the distortions D_1 and D_2, and proved the rate distortion function for this sum rate.

This paper aims to develop practical low-complexity codes to approach the theoretical bounds derived in [1] for the quadratic Gaussian case of the problem. Although the achievability proof given in [1] cannot be directly applied in practice, the theoretical construction can provide insight into practical code design. In the theoretical construction, a single message is sent to both decoders, but decoder 1 ignores part of the message, which can be decoded by only decoder 2, and decodes the rest of the message. This can be viewed as two descriptions of the source being sent: one common description to both decoders and one description intended only for decoder 2. However, this is not a 2-multiple decription (MD) problem [2, 3], because the description intended only for decoder 2 cannot be decoded without the common description whereas in MD problem both descriptions can be decoded independently. Thus, the code construction to this problem is more closely related to the successive refinement problem [4, 5].

Since we are encoding a continuous source, the source should be quantized first which could be also used to generate successively refinable descriptions. One easy way to generate successively refinable descriptions is to apply a uniform nested scalar quantizer (NSQ) as applied in [6] for successive refinement for the Wyner-Ziv problem [7]. A uniform NSQ consists of a number of quantization levels (stages) equal to the number of refinement stages and each quantization level is nested within the upper level such that the quantization points in the lower level refine the interval between two quantization points in the upper level. At each stage, the source can be quantized using the respective quantization level and the bits required to represent the refinement can be sent to the decoder. It was shown in [6] that uniform NSQ followed by LDPC based Slepian-Wolf coding achieves distortions $1.29 - 3.45$dB away from the theoretical limits for the Gaussian successive refinement Wyner-Ziv problem (i.e. when side information is always present). A work more related to ours can be found in [8], which applies NSQ when the symbols in the side information sequence are degraded or erased with some probability.

In [9], trellis coded quantization (TCQ) [10] was shown to achieve better performance than the scalar quantizer for the Wyner-Ziv problem without successive refinement. This insight was further supported by Yang et al. [11] development of codes based on TCQ and LDPC codes [12] which are very close the the Wyner-Ziv theoretical limit. This suggests that TCQ may better suited to successive refinement with side information than NSQ. For this, TCQ must be adapted to successive refinability.

In this vein, Jafarkhani and Tarokh introduced a rate-scalable trellis quantization called successively refinable TCQ (SR-TCQ) in [13]. Similar to NSQ, SR-TCQ also uses nested quantization levels, however SR-TCQ differs in that the consecutive quantized values of a sequence are influenced by a trellis structure which reduces the number of bits required to represent the quantization points.

Here, we apply SR-TCQ with 2 refinement stages to get one common description and one individual description of the source for our problem. As it must be capable of being decoded without side information, the common (base) description is losslessly compressed down to its entropy. To compress the bit planes of the refinement (individual) description, we send the syndrome of a powerful LDPC channel code. This syndrome is then decoded to regain the refinement bit planes at the receiver by exploiting the correlation between the quantized source and the side information in a belief propagation decoder as in [11].

In the next section, we discuss how SR-TCQ can be applied to generate the descriptions.

2. GENERATING DESCRIPTIONS USING SR-TCQ

Suppose that the source X and the side information Y are jointly distributed such that $Y = X + Z$, where $Z \sim \mathcal{N}(0, \sigma_Z^2)$ is independent of $X \sim \mathcal{N}(0, \sigma_X^2)$. Decoders 1 and 2 are expected to reproduce X such that $E[d_1(X, \hat{X}_1(U))] \leq D_1$ and $E[d_2(X, \hat{X}_2(U, V, Y))] \leq D_2$ respectively, where U is the common description and V is the description intended only for decoder 2. In our work, we use the squared distortion measure for both d_1 and d_2.

We apply SR-TCQ to generate the descriptions U and V from the source X. Before we get into the details of how the descriptions are generated, we briefly describe the SR-TCQ technique in Section 2.1.

2.1. SR-TCQ

In this section, we describe SR-TCQ [13] process for only 2 refinement stages since that is what we need for our work. Suppose that in the first stage the description (U) is sent at rate r_1 and in the second stage the description (V) is sent at rate r_2. Then, we need to have two sets of quantization points Q_1 and Q_2, one for each stage. The set Q_1 consists of 2^{r_1+1} quantization points, $Q_1 = \{q_i : i \in \{1, \ldots, 2^{r_1+1}\}\}$ and the set Q_2 consists of 2^{r_2+1} quantization points for each one of the quantization points $i \in Q_1$, i.e. $Q_2 = \{q_{i,j} : j \in \{1, \ldots, 2^{r_2+1}\},\ i \in \{1, \ldots, 2^{r_1+1}\}\}$. At both stages, the quantization points are partitioned into 4 cosets (C^0, C^1, C^2, C^3) in the same way it is done in the TCQ. Next, consider the construction of the trellis used in SR-TCQ.

The trellis for SR-TCQ is constructed by taking the tensor product of the trellises of the two refinement stages. In particular, the trellis for our problem is constructed by taking the tensor product $T_1 \otimes T_2$ of the first stage trellis T_1 and the second stage trellis T_2. Suppose that T_1 and T_2 have states $v_1, v_2, \ldots, v_{2^{n_1}}$ and $w_1, w_2, \ldots, w_{2^{n_2}}$, respectively. Then $T_1 \otimes T_2$ consists of $2^{n_1+n_2}$ states $v_i \otimes w_j, (i, j) \in \{1, 2, \ldots, 2^{n_1}\} \times \{1, 2, \ldots, 2^{n_2}\}$. There is a transition between states $v_i \otimes w_j$ and $v_k \otimes w_\ell$ in $T_1 \otimes T_2$ if and only if there is a transition between v_i and v_k in T_1, and there is a transition between w_j and w_ℓ in T_2. Denote the trellis constructed by the tensor product by $T \triangleq T_1 \otimes T_2$.

Having described how the trellis for SR-TCQ is constructed, we next discuss the quantization process in SR-TCQ. To quantize the source sequence, we break it into blocks of N symbols and apply SR-TCQ for each block. In SR-TCQ, the Viterbi algorithm is used to find the quantization sequence which minimizes the quantization error. Unlike in the single description case, here the quantization errors (distortions) at both stages should be taken into account when the distortion is minimized. Depending on the application, different weights can be given to the distortions D_1 and D_2 when the distortion to be minimized D is selected. Thus, for a block

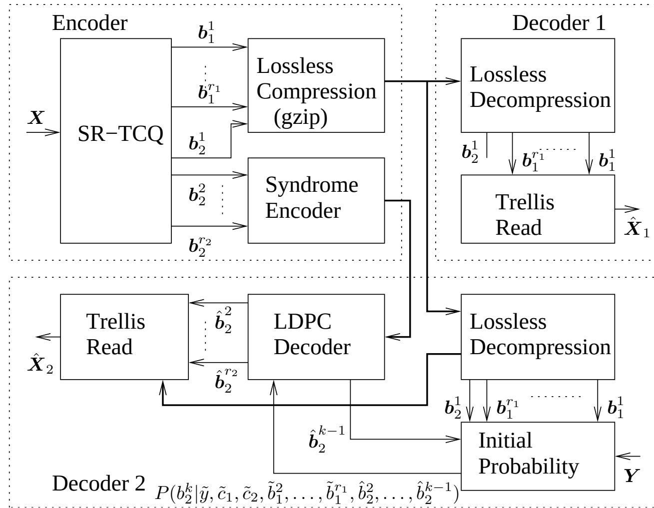

Fig. 2. The system architecture of the practical code design we propose.

$x(1), \ldots, x(N)$, D can be defined as [13]

$$D = \frac{1}{N}\{\alpha \sum_{n=1}^{N}[(x(n) - \hat{x}_1(n)]^2 + (1-\alpha)\sum_{n=1}^{N}[(x(n) - \hat{x}_2(n)]^2\} \tag{1}$$

where α is the weighting factor.

2.2. Generating Descriptions

We are now ready to explain how the descriptions U and V are generated. For each block, SR-TCQ outputs a sequence of quantized values $\{q_{i,j}(n)\}_{n=1}^N$. Given $q_{i,j}(n) \in Q_2$, one can easily find the quantization point $q_i(n) \in Q_1$ selected for the first stage and this is the common description $U(n)$ of the source $X(n)$ which is sent to both decoders. To describe (before compression) U we need only r_1 bits, because only 2 cosets leave each state in trellis T_1 [10, 13]. The refinement description $V(n)$ describes which of the quantization points in Q_2 nested within $q_i(n)$ is selected. Again, because of the same argument made for the first stage, only r_2 bits (before compression) are required to represent the decription V.

For each symbol $n \in \{1, \ldots, N\}$, denote the bits that represent the description $U(n)$ by $b_1^1(n), \ldots, b_1^{r_1}(n)$ and denote the bits that represent the description $V(n)$ by $b_2^1(n), \ldots, b_2^{r_2}(n)$. Provided the current state in the trellis T_1, the bit $b_1^1(n)$ determines which one of the two branches and cosets is selected, while the remaining bits $b_1^2(n), \ldots, b_1^{r_1}(n)$ determine which one of the quantization points within that coset is selected. Similarly, given the common quantization level, $b_2^1(n)$ determines which one of the two branches in T_2 is selected while $b_2^2(n), \ldots, b_2^{r_2}(n)$ determine which one of the quantization points within that coset is selected. For each $m \in \{1, 2\}$ and $k \in \{1, \ldots, r_m\}$, we call the vector $\boldsymbol{b}_m^k = [b_m^k(1), \ldots, b_m^k(N)]$ a bit-plane. Also, denote the collection of the bit-planes by $\boldsymbol{B}_m = [\boldsymbol{b}_m^k : k \in \{1, \ldots, r_m\}],\ m \in \{1, 2\}$.

We next turn our attention to lossless compression of the bit-planes discussed above.

3. LOSSLESS COMPRESSION OF THE BIT-PLANES

There exists significant redundancy in the SR-TCQ bit-planes in $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ that can be exploited to compress them losslessly as shown in Fig. 2. This lossless compression allows the same distortions to be achieved with a lower rate.

First consider the bit-planes $\boldsymbol{b}_1^1$ and $\boldsymbol{b}_2^1$. These two bit-planes have memory in them and should be compressed with an universal source coding technique (e.g. Lempel-Ziv). Given these two bit-planes the bits in the other bit-planes are independent of one another and as a result we have the following.

$$\begin{aligned} & H(\boldsymbol{b}_1^2,\ldots,\boldsymbol{b}_1^{r_1}|\boldsymbol{b}_1^1,\boldsymbol{b}_2^1) \\ = & \sum_{n=1}^{N}\Big(H(b_1^2(n)|\boldsymbol{b}_1^1,\boldsymbol{b}_2^1)+\ldots \\ & +H(b_1^{r_1}(n)|\boldsymbol{b}_1^1,\boldsymbol{b}_2^1,b_1^2(n),\ldots,b_1^{r_1-1}(n))\Big) \\ = & \sum_{n=1}^{N}\Big(H(b_1^2(n)|c_1(n),c_2(n))+\ldots \\ & +H(b_1^{r_1}(n)|c_1(n),c_2(n),b_1^2(n),\ldots,b_1^{r_1-1}(n))\Big) \end{aligned} \quad (2)$$

where $c_1(n)$ and $c_2(n)$ are the cosets selected for symbol $x(n)$ in the first and second levels, respectively. During the decompression of the bit-planes in $\boldsymbol{B}_2 \setminus \{\boldsymbol{b}_2^1\}$, both the side information $\boldsymbol{Y}$ and $\{\boldsymbol{B}_1,\boldsymbol{b}_2^1\}$ can be exploited, and thus for these bit-planes we have

$$\begin{aligned} & H(\boldsymbol{b}_2^2,\ldots,\boldsymbol{b}_2^{r_2}|\boldsymbol{Y},\boldsymbol{B}_1,\boldsymbol{b}_2^1) \\ = & \sum_{n=1}^{N}\Big(H(b_2^2(n)|y(n),c_1(n),c_2(n),b_1^2(n),\ldots,b_1^{r_1})+\ldots \\ & +H(b_2^{r_2}(n)|y(n),c_1(n),c_2(n),b_1^2(n),\ldots,b_1^{r_1}(n), \\ & b_2^2(n),\ldots,b_2^{r_2-1}(n))\Big) \end{aligned} \quad (3)$$

Thus, the goal now is to compress these bit-planes successively to their conditional entropies. We defer our discussion on the computation of $H(\boldsymbol{b}_2^2,\ldots,\boldsymbol{b}_2^{r_2}|\boldsymbol{Y},\boldsymbol{B}_1,\boldsymbol{b}_2^1)$ until Section 4 and now discuss how the bit-planes can be successively compressed.

The *Lossless Compression* block in Fig. 2, takes the bit-planes $\{\boldsymbol{B}_1,\boldsymbol{b}_2^1\}$ and applies an universal source coding technique to $\boldsymbol{b}_1^1,\boldsymbol{b}_2^1$ and applies any lossless source coding technique to the rest of the bit-planes. Note that since the bits in the other bit-planes are independent given $\boldsymbol{b}_1^1,\boldsymbol{b}_2^1$, a Huffman code could be efficiently used with the conditional probabilities. For maximal simplicity, we applied *gzip* program to compress all bit-planes in $\{\boldsymbol{B}_1,\boldsymbol{b}_2^1\}$ for our simulations. These compressed sequences are then sent over a noiseless channel to both decoders.

In recent years, many people have successfully applied powerful channel codes (Turbo [14] and LDPC codes [12]) to compress the sources when highly correlated side information is available at the decoder [11]. Following these constructions, the *Syndrome Encoder* applies r_2-1 LDPC codes, one for each bit-plane, to compress $\boldsymbol{B}_2 \setminus \{\boldsymbol{b}_2^1\}$ and generates syndrome sequences $\boldsymbol{S}=[\boldsymbol{s}^k : k\in\{2,\ldots,r_2\}]$ [11] which are then sent to decoder 2. The rates (compression ratio) of the codes used for different bit-planes differ depending on the conditional entropies of the bit planes. Normally, the compression increases as the bit-plane number increases.

At the decoders, first the *Lossless Decompresssion* blocks losslessly decode the compressed sequences of $\{\boldsymbol{B}_1,\boldsymbol{b}_2^1\}$. The *Trellis Read* module at decoder 1 uses the decoded $\boldsymbol{B}_1$ to reconstruct the source as $\hat{\boldsymbol{X}}_1$. Note that $\boldsymbol{b}_2^1$ is not used during the reconstruction at decoder 1, because knowing the first bit of V alone is not enough to predict the best quantization point in Q_2. However, $\boldsymbol{b}_2^1$ can be used for compression purposes as in (2).

At decoder 2, the *Initial Probability* block takes $\boldsymbol{Y},\boldsymbol{B}_1,\boldsymbol{b}_2^1$ and previously decoded bits $\hat{\boldsymbol{b}}_2^2,\ldots,\hat{\boldsymbol{b}}_2^{k-1}$ as input and calculates the probability $P(b_2^k|\tilde{y},\tilde{c}_1,\tilde{c}_2,\tilde{b}_1^2,\ldots,\tilde{b}_1^{r_1},\hat{b}_2^2,\ldots,\hat{b}_2^{k-1})$ required to decode $\boldsymbol{s}^k$ at the *LDPC Decoder*, where $\tilde{x}$ denotes the realization of x. The *LDPC Decoder* successively decodes the syndrome sequences $\boldsymbol{S}$ by running a message-passing algorithm which uses the probability $P(b_2^k|\tilde{y},\tilde{c}_1,\tilde{c}_2,\tilde{b}_1^2,\ldots,\tilde{b}_1^{r_1},\hat{b}_2^2,\ldots,\hat{b}_2^{k-1})$. The *LDPC Decoder* and the *Initial Probability* block communicate with each other every time a bit-plane is decoded. The losslessly decoded bit-planes $\{\boldsymbol{B}_1,\boldsymbol{b}_2^1\}$ and the bit-planes $\hat{\boldsymbol{b}}_2^2,\ldots,\hat{\boldsymbol{b}}_2^{r_2}$ are then used to reconstruct the source as $\hat{\boldsymbol{X}}_2$.

4. COMPUTATION OF THE ENTROPIES

It remains to explain, extending [11], how to compute the conditional entropies $H(b_2^k|y,c_1,c_2,b_1^2,\ldots,b_1^{r_1},b_2^2,\ldots,b_2^{k-1})$ and the probabilities $P(b_2^k|\tilde{y},\tilde{c}_1,\tilde{c}_2,\tilde{b}_1^2,\ldots,\tilde{b}_1^{r_1},\hat{b}_2^2,\ldots,\hat{b}_2^{k-1})$. The entropy $H(\boldsymbol{B}_1,\boldsymbol{b}_2^1)$ is calculated by generating long sequences of the bit-planes $\{\boldsymbol{B}_1,\boldsymbol{b}_2^1\}$, compressing them using *gzip* and calculating the compression ratio achieved by *gzip* under the assumption that it achieves a compression ratio close to the entropy. For $\boldsymbol{b}_2^k,\ k\in\{2,\ldots,r_2\}$, the rates for the LDPC codes are selected by calculating the entropies $H(b_2^k|y,c_1,c_2,b_1^2,\ldots,b_1^{r_1},b_2^2,\ldots,b_2^{k-1})$, where we assumed that the entropy is equal for all $n\in\{1,\ldots,N\}$ in (3). We use the same technique used in ([11]-equation 11) to calculate the entropies, but we need to use the probabilty, $P(b_2^k|y,c_1,c_2,b_1^2,\ldots,b_1^{r_1},b_2^2,\ldots,b_2^{k-1})$, to calculate the entropy.

The probabilities required at the LDPC decoder are also computed by using the technique used in ([11]-equation 12) where we use the probability $P(b_2^k|\tilde{y},\tilde{c}_1,\tilde{c}_2,\tilde{b}_1^2,\ldots,\tilde{b}_1^{r_1},\hat{b}_2^2,\ldots,\hat{b}_2^{k-1})$.

5. SIMULATION RESULTS

We implemented the proposed practical coding scheme for lossy compression when side information may be absent and tested it via simulation. We selected $\sigma_X^2=1$ and $\sigma_Z^2=0.1225$ to generate i.i.d. source and side information sequences. The source sequence is broken into blocks of length $N=1000$ before SR-TCQ is applied to each block. In SR-TCQ, Ungerboeck's [15] 4-state trellis is used for both T_1 and T_2. The quantization points used in SR-TCQ are optimized using the algorithm provided in [13] where the distortion to be minimized D is selected by setting $\alpha=0.5$. Note that a series a points can be obtained in the rate distortion region by selecting a series of values for α.

We use *gzip* program to losslessly compress the bit-planes $\{\boldsymbol{B}_1,\boldsymbol{b}_2^1\}$ and to compute the entropy $H(\boldsymbol{B}_1,\boldsymbol{b}_2^1)$ that is reported in this section. For the compression of $\boldsymbol{B}_2 \setminus \{\boldsymbol{b}_2^1\}$, we apply LDPC codes of length 10^5 which means that we group together the SR-TCQ output of 100 blocks before we apply LDPC. The rates of the codes are selected according to the entropies calculated. In particular, for a bit-plane with entropy H_k we select the code rate such that the compression achieved by the code is

$$r=\frac{H_k+\delta(1+H_k)}{1-\delta(1+H_k)} \quad (4)$$

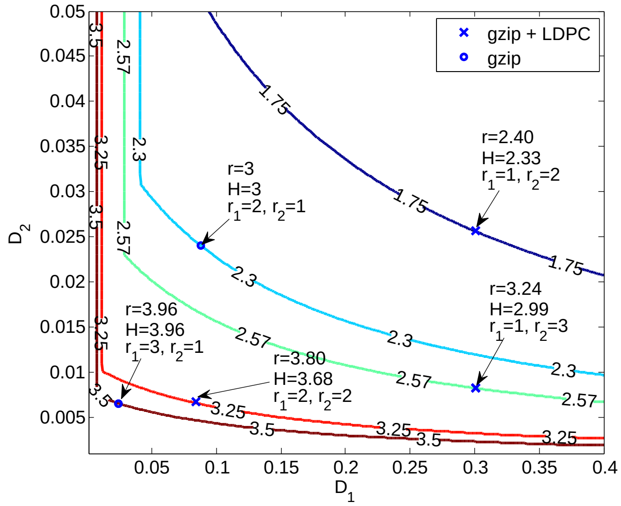

Fig. 3. Comparison of the rate distortion points obtained using our practical code design with the theoretical bounds. For SR-TCQ, a 4-state trellis for both T_1 and T_1 was used, and a code length of 10^4 was used for the LDPC codes

where δ is selected such that the bit error rate of the code is almost 0. We use a degree distribution of $\lambda(x) = x^2$ and $\rho(x) = (1-\rho_m)x^{m-2} + \rho_m x^{m-1}$ for our LDPC codes [16], where $m = \lceil 3/H_k \rceil$.

Fig. 3 compares the rate distortion points we obtained with our scheme with the theoretical bounds derived by Heegard and Berger [1]. In the plot, H denotes the entropy of each symbol ($b/symb$), r denotes the average number of bits required to encode each symbol, and r_1 and r_2 denote the rates defined in 2.1. For all 5 points, we were able to achieve the distortions D_1 and D_2 with no more than 0.7 extra bits from the theoretical bound.

Note that the difference between our result and the theoretical bounds is only 0.46 bits when $r_1 = 3$ and $r_2 = 1$ bits. Also note that, among all the codes with the same total rate $r_1 + r_2$, the one with the higher rate for the first refinement achieves performance closer to the theoretical limits.

6. CONCLUSION

We proposed a practical code design for quadratic Gaussian lossy compression when side information may be absent by using SR-TCQ and LDPC codes. In this code design we generated two descriptions using 2 refinement level SR-TCQ, compressed the common description using standard universal lossless compression, while compressing the individual description using LDPC codes for decoding with side information. Our simulations show that on average we require only 0.46 extra bits than the theoretical bounds at moderate rates. Simulations also showed that, among the proposed codes, the ones with the higher rate for the first refinement achieves the best performance among all the codes with the same total TCQ rate.

7. REFERENCES

[1] Chris Heegard and Toby Berger, "Rate distortion when side information may be absent," *IEEE Transactions on Information Theory*, vol. 31, no. 6, pp. 727–734, November 1985.

[2] A. A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Transactions on Information Theory*, vol. IT-28, no. 6, pp. 851–857, November 1982.

[3] L. Ozarow, "On a source-coding problem with two channels and three receivers," *The Bell System Technical Journal*, vol. 59, no. 10, pp. 1909–1921, December 1980.

[4] William H. R. Equitz and Thomas M. Cover, "Successive Refinement of Information," *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 269–275, Mar. 1991.

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

[6] Samuel Cheng and Zixiang Xiong, "Successive refinement for the wyner-ziv problem and layered code design," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3269–3281, Aug. 2005.

[7] Aaron D. Wyner and Jacob Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. IT-22, no. 1, pp. 1–10, January 1976.

[8] Francesca Bassi, Michel Kieffer, and Claudio Weidmann, "Source coding with intermittent and degraded side information at the decoder," in *International Conference on Acoustics, Speech and Signal Processing*, 2008.

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

[10] Michael W. Marcellin and Thomas R. Fischer, "Trellis coded quantization of memoryless and gauss-markov sources," *IEEE Transactions on Communications*, vol. 38, no. 1, pp. 82–93, Jan. 1990.

[11] Yang Yang, Samuel Cheng, Zixiang Xiong, and Wei Zhao, "Wyner-ziv coding based on tcq and ldpc codes," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 376–387, Feb. 2009.

[12] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[13] Hamid Jafarkhani and Vahid Tarokh, "Design of successively refinable trellis-coded quantizers," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1490–1497, Jul. 1999.

[14] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[15] Gottfried Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.

[16] Angelos D. Liveris, Zixiang Xiong, and Costas N. Georghiades, "Compression of binary sources with side information at the decoder using ldpc codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, Oct. 2002.