On The Problem of Simultaneous Encoding of Magnitude and Location Information

Rui Castro Michael Wakin Michael Orchard

Department of Electrical and Computer Engineering, Rice University
6100 Main St., Houston, TX, 77005

Abstract

Modern image coders balance bitrate used for encoding the location of significant transform coefficients, and bitrate used for coding their values. The importance of balancing location and value information in practical coders raises fundamental open questions about how to code even simple processes with joint uncertainty in coefficient location and magnitude. This paper studies the most basic example of such a process: a 2-D process studied earlier by Weidmann and Vetterli that combines Gaussian magnitude information with Bernoulli location uncertainty. The paper offers insight into the coding of this process by investigating several new coding strategies based on more general approaches to lossy compression of location. Extending these ideas to practical coding, we develop a trellis-coded quantization algorithm with performance matching the published theoretical bounds. Finally, we evaluate the quality of our strategies by deriving a rate-distortion bound using Blahut's algorithm for discrete sources.

1. Introduction

Wavelets provide a sparse representation of information in natural images, with low-frequency energy captured in a few coefficients, and information about edges captured in the few coefficients near the edges. The information of the edge coefficients consists both of uncertainty in the location of those coefficients, and uncertainty in their values. A wide range of today's top image coding algorithms, using an approach that is mathematically known as "nonlinear approximation", devote one portion of the bitrate to specifying locations of coefficients to be coded, and a second portion of bitrate to code the values of those coefficients. Many intuitively reasonable approaches for balancing bitrates have been used to code location and magnitude of coefficients, but the fundamental principles governing optimal trade-offs between location and magnitude uncertainty have not been studied, even in the simplest cases. The most common approach used in practical image coders is to encode the location information without error, and to separately encode the magnitude. This naïve approach can be found among popular coders (e.g. [1]) and in theoretical analysis [2]. Recently, Weidmann and Vetterli [3, 4] added formal structure to the problem of joint encoding for magnitude and location information, proposing and analyzing a novel compression framework.

This paper explores the fundamental principles governing efficient coding of processes with joint location and magnitude uncertainty by presenting, analyzing, and comparing a variety of coding strategies that improve on previously published bounds on coding performance. Motivated by these coding strategies, we develop an efficient and practical coder for such processes, and investigate theoretical rate-distortion (R-D) behavior. Our work focuses on a very simple 2-D process with very structured joint magnitude-location uncertainty, one that combines Gaussian magnitude uncertainty with Bernoulli location uncertainty. Despite the simplicity of the formulation, a closed form solution for the optimal R-D curve remains unknown.

To investigate the coding of this process, we propose a sequence of frameworks for reaching a jointly optimal balance of location and magnitude information, and we develop coding strategies from these frameworks. For each, we derive the operational R-D curve, and we compare these curves to gain insights on the coding of this process. Our later coding strategies improve upon the bound introduced by Weidmann and Vetterli.

Conclusions drawn from our analysis suggest that good practical coders for our test process should reflect specific relationships between the encoded bitstreams representing location and magnitude. Motivated by these conclusions, we propose a practical compression scheme based on trellis-coded quantization (TCQ). Our novel approach works by alternately optimizing two stages of trellis coding, one for magnitude and one for location. The Viterbi optimization of the trellis paths improves upon earlier approaches for managing tradeoffs between magnitude and location errors, and this practical algorithm actually matches the performance of earlier proposed bounds for the coding of this process.

Email:{rcastro, wakin, orchard}@rice.edu. Web: dsp.rice.edu

Finally, we apply the Blahut algorithm [5] to derive a theoretical upper bound on the R-D curve of the process. We use this bound to assess the significance of performance improvements achieved in our study. We conclude that the coding strategies offered in this paper represent significant advances on earlier studied coding strategies, but that room for modest improvement remains at low bitrates.

This paper is organized as follows: Section 2 formulates the problem; Section 3 investigates a series of coding strategies for this problem, and derives upper bounds for the R-D performance of each; Section 4 develops a practical TCQ-based coder and compares its operational performance to the bounds of Section 3; Section 5 uses Blahut's algorithm to bound the optimal R/D curve; and Section 6 offers some concluding observations.

2. Problem Formulation

Consider a memoryless source where symbols are $\mathbb{R}^2$ vectors where one of the entries is zero and the other entry is drawn from a Gaussian distribution. Our goal is to encode such a source.

Let $\{X_i\},\ i \in \mathbb{N}$ be independent and identically distributed (i.i.d.) random variables, $X_i \sim \mathcal{N}(0, \sigma^2)$; let $\{B_i\}$ be i.i.d. Bernoulli random variables, independent of $\{X_i\}$, $B_i \sim Ber(1/2)$ (B_i takes values on $\{0, 1\}$).

Define the symbols $Y_i = X_i \cdot (B_i, 1 - B_i)$, that is, $Y_i \in \mathbb{R}^2$ and takes one of the two possible forms, either $(0, X_i)$ or $(X_i, 0)$. Each symbol is characterized by the magnitude X_i and the location information B_i, indicating what entry of the 2-tuple Y_i is valued X_i and consequently what entry of Y_i is valued 0. Another way of defining the symbols Y_i is to regard them as drawn from a probability measure μ in $\mathbb{R}^2$ such that the axes have probability one.

Our objective is to construct a sequence $\{\widehat{Y}_i\}$ encoding the sequence $\{Y_i\}$ using on average R bits per symbol, such that the mean square error (MSE) distortion measure

$$D(R) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[(Y_i - \widehat{Y}_i)^2] \tag{1}$$

is minimized. Denote by $D^*(R)$ the optimal solution of this problem. This corresponds to the theoretical R-D curve.

3. Coding Strategies

We consider a series of coding schemes and compare them using their operational R-D curves, which yield upper bounds for the curve $D^*(R)$. Most of these approaches involve explicitly encoding the magnitude sequence $\{X_i\}$ with R_X bits per symbol and the location sequence $\{B_i\}$ with R_B bits per symbol, giving rise to the sequences $\{\widehat{X}_i\}$ and $\{\widehat{B}_i\}$, respectively. These sequences are then translated into the final estimate $\{\widehat{Y}_i\}$. Recall for Gaussian random variables that the optimal MSE rate distortion curve is

$$D_X(R_X) = \sigma_e^2 = \sigma^2 2^{-2R_X}, \tag{2}$$

where R_X is the bit rate per symbol. For Bernoulli random variables, we consider the Hamming distortion, corresponding to probability of error $p_e = D_B(R_B)$ where $p_e \in [0, 1/2]$ and

$$R_B = 1 + p_e \log_2(p_e) + (1 - p_e)\log_2(1 - p_e), \tag{3}$$

where R_B is the bit rate per symbol.

3.1. Basic

The simplest coding scheme for this problem (and the one used on practical coders of sparse sources as well) involves spending one bit per symbol conveying the location information B_i and using the remaining bits to encode the magnitude information X_i (for $R > 1$). In this case we set $\widehat{Y}_i = \widehat{X}_i \cdot (\widehat{B}_i, 1 - \widehat{B}_i)$. Figure 1 shows the operational R-D bound for this coder obtained by taking the convex hull of admissible operating points. In practice, for rates less than a bit per symbol, achieving a point on the convex hull requires a time-sharing scheme to average the performance between two operating points.

3.2. Location/Magnitude Uncertainty

The above strategy relies on lossless encoding of the location information. To improve on that technique we consider another coding scheme, exploiting vector quantization results in both location and magnitude information.

First, we encode optimally the magnitude sequence $\{X_i\}$ and the location sequence $\{B_i\}$ using R_X and R_B bits per symbol (2,3). It can be shown that the optimal estimate $\widehat{Y}_i$ in terms of the MSE (1) is then given by

$$\widehat{Y}_i = \widehat{X}_i \cdot \Big[(1 - p_e, p_e)\widehat{B}_i + (p_e, 1 - p_e)(1 - \widehat{B}_i)\Big]. \tag{4}$$

Notice that, in general, both entries of this estimate are nonzero.

The optimal values of σ_e^2 and p_e (which indicate the allocation of the bitrate R between R_X and R_B) can be found using numerical methods. Figure 1 shows the corresponding bound, an improvement over the previous bound. For rates $R < 1/2$ we send only magnitude information, while for $R > 3$ we essentially send one bit per symbol to encode the location information. The behavior of this coder at low bitrates suggests a different strategy, leading to the following approach.

3.3. Classification-Based Strategies

Type 1. Clearly, location errors are most costly when the magnitude of X_i is large. That suggests the following

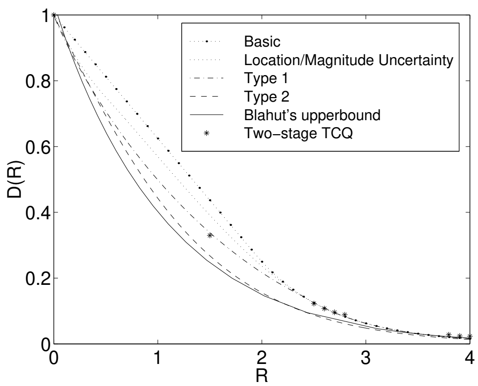

Fig. 1. *Theoretical and operational R-D bounds for $\sigma^2 = 1$. Lines describe theoretical upper bounds derived in Section 3. Asterisks mark operational points using the TCQ method of Section 4. This practical coder achieves performance very comparable to the two-class scheme [3].*

scheme, described in [3] and included here for the sake of completeness.

Consider the two sequences

$$\begin{aligned} \{a_i\} &= \{i : |X_i| \leq T\}, \\ \{b_i\} &= \{i : |X_i| > T\}, \text{where } T > 0. \end{aligned}$$

We first send the indexing information in $\{a_i\}$ (or equivalently $\{b_i\}$). For the elements of the sequence $\{Y_{b_i}\}$ we spend one bit per symbol encoding the location information B_{b_i}, and R' bits encoding the magnitude. We do not spend any bits encoding the sequence $\{Y_{a_i}\}$. For any symbol Y_{a_i} we use the representation $\widehat{Y}_{a_i} = (0, 0)$. Hence we are indeed coding only symbols with large magnitudes.

The optimal threshold is obtained by minimizing the distortion with respect to T, for a given rate R. To be able to do this, we need to bound the distortion incurred when coding the (non-Gaussian) sequence X_{b_i}. An upper bound is provided by a Gaussian having the same variance; notice that this does not give a tight bound.

The corresponding bound is depicted in Figure 1. As can be seen, this improves on our earlier bounds. For high rates ($R > 3$) the optimal threshold T is zero; that is, we obtain the bound of Section 3.1.

Type 2. One of the drawbacks of the previous approach is that we need to send the indexing information describing $\{a_i\}$. To avoid this we consider now a scheme using classification *after* encoding the magnitude values. Depending on the value of the encoded magnitude $\widehat{X}_i$ we encode only partial location information with probability of error $p(\widehat{X}_i) \in [0, 1/2]$. Using (4) to construct the encoded sequence $\widehat{Y}_i$, the distortion is a function of both σ_e^2 and $p(\cdot)$.

Finding the optimal configuration is difficult, but we simplify the problem by considering a parametric class for the function $p(\cdot)$, taking the form

$$p(x) = \sum_{k=0}^{n-1} p_k I_{[\alpha_k, \alpha_{k+1}]}(x), \tag{5}$$

where $\alpha_0 = -\infty, \alpha_n = \infty$, and $\alpha_k < \alpha_{k+1}\ \forall i$.

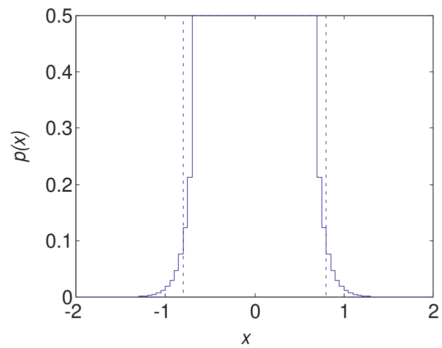

Fig. 2. *Function $p(\cdot)$ for R=1.35 bits. Solid: Multiple classes ($n = 80$ classes). Dashed: Two classes.*

Optimizing the corresponding MSE distortion over σ_e^2 and p_k yields the rate distortion bound shown in Figure 1. There is a significant improvement over the previous bounds, especially for rates $R > 3$. Unlike the previous coding schemes, this scheme only sends *partial* location information for those rates.

It is interesting to observe that the general shape of the function $p(\cdot)$, in Figure 2, essentially describes a classification of the source symbols into two classes: for small magnitudes no location information is encoded, and for large magnitudes location is encoded without error. This is similar in spirit to the *Type 1* scheme, although we encode magnitude information even for small magnitude values. Although this may explain some difference in performance between the two coders, we believe that the apparently large gap between the bounds may have another possible explanation: Computing the R-D bound for the *Type 1* scheme requires bounding the distortion of the sequence $\{X_{b_i}\}$. There is no closed form for the distortion incurred on coding such a sequence, forcing us to use an upper bound that is possibly not tight. This yields a R-D bound for the *Type 1* strategy that might not be representative of its actual performance.

4. Practical Implementation

In this section, we develop a practical and feasible compression scheme based on trellis-coded quantization (TCQ) [6]. TCQ allows efficient quantization of Gaussian random variables, and its optimization structure is easily adapted for lossy compression of Bernoulli random variables.

Although TCQ encodes a sequence of variables one symbol at a time, it achieves compression efficiency by

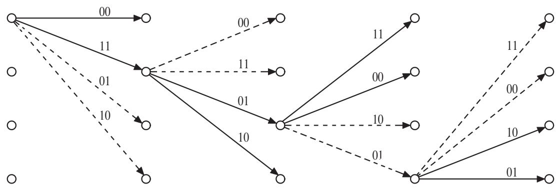

Fig. 3. *Location encoding with penalized transitions in a four-state Ungerboeck trellis. Each transition corresponds to a pair of encoded Bernoulli values. Arithmetic coding is used to specify the particular transition from a given state. Each state has four possible transitions under the assumed distribution Q: two with probability q (solid lines), and two with probability $0.5 - q$ (dashed lines).*

placing a dependence on the quantization errors. Errors are cleverly placed in locations where they minimize the impact on the coded sequence. For TCQ coding of a Gaussian, for example, variables are often quantized to the wrong quantization bin in order to save bits. The Viterbi algorithm, however, weighs the distortion costs of such errors, in order to minimize the overall impact on distortion.

We use two stages of TCQ to code our information. Magnitude information $\{X_i\}$ is coded in the first stage, and location information $\{B_i\}$ is coded in the second. The quantized symbols $\{\widehat{Y}_i\}$ are assembled at the decoder by combining the sequences $\{\widehat{X}_i\}$ and $\{\widehat{B}_i\}$. Each stage is required to meet a target rate (R_X or R_B) and optimizes its trellis path based on the path chosen by the other stage; the Viterbi algorithm helps to minimize the impact of errors. Before actually encoding any values, we iterate between optimizing the two stages in order to find the jointly optimal pair of trellis paths.

4.1. Stage I: Magnitude Optimization

Stage I assumes that $\{B_i\}$ has been coded lossily to $\{\widehat{B}_i\}$, and it attempts to find the optimal trellis-coding for $\{X_i\}$ with a desired rate of R_X bits per symbol. Our algorithm uses the Ungerboeck trellis and is identical to the Gaussian TCQ algorithm presented in [6]; the only difference in our implementation is the distortion function which is minimized. At each node, we set $\widehat{Y}_i = \widehat{X}_i \cdot (\widehat{B}_i, 1 - \widehat{B}_i)$. Instead of minimizing the distortion of the Gaussian itself, we choose a trellis path for $\{\widehat{X}_i\}$ to minimize the distortion of $\{\widehat{Y}_i\}$. As in [6], a TCQ scheme such as this restricts our operational values of R_X to be integers.

4.2. Stage II: Location Optimization

Stage II assumes that $\{X_i\}$ has been coded lossily to $\{\widehat{X}_i\}$, and it attempts to find the optimal trellis-coding for $\{B_i\}$ with a desired rate of R_B bits per symbol. To allow lossy encoding of location information (rates $R_B < 1$), we use a special adaptation of the Ungerboeck trellis, where each transition encodes a pair of locations. The pair of locations has four equally likely outcomes; typically two bits would be required to encode each pair. By adjusting the likelihood of the encoded transitions, however, we may encode pairs using fewer than two bits.

In particular, we allow four transitions from each state in the trellis. These correspond to the four possible outcomes of a *pair* of Bernoulli variables. We encode the trellis path using arithmetic coding with transition probability distribution $Q = \{q, q, 0.5 - q, 0.5 - q\}$, with $q \in [0, 1/4]$. As shown in Figure 3, each transition is labeled with the corresponding pair of encoded Bernoulli values. For any trellis path that actually obeys the distribution Q, then, each transition takes on average $H(Q)$ bits to encode; notice that $H(Q) \in [1, 2]$. We adjust q to in order to ensure that $H(Q) = 2R_B$ as desired. Thus, we may meet any desired $R_B \in [0.5, 1]$ by adjusting the distribution Q, and by finding a path through the trellis that obeys that distribution.

For a fixed distribution Q, we use Lagrangian optimization to find the optimal trellis path that obeys the distribution. In particular, we define D_{path} to be the distortion of a given path through the trellis, where we measure the distortion of the sequence $\widehat{Y}_i = \widehat{X}_i \cdot (\widehat{B}_i, 1 - \widehat{B}_i)$. We define R_{path} to be the rate required to encode a particular trellis path according to the model Q. For any value of λ, then, the Viterbi algorithm finds the trellis path which minimizes the quantity $D_{path} + \lambda R_{path}$; individual transitions are simply penalized by their Lagrangian costs instead of their distortions. We search for the proper value of λ such that the trellis path minimizing $D_{path} + \lambda R_{path}$ also matches the distribution Q. As a result, we have found the path with the lowest possible distortion that meets $R_{path} = R_B$.

4.3. Results

In practice, we use an eight-state Ungerboeck trellis for each stage. (The number of allowable transitions does not depend on the number of states). To initialize the two-stage coder, we assume that $\{\widehat{B}_i\} = \{B_i\}$. We iterate three times between the two stages, as the chosen trellis paths seem to converge rather quickly.

We find optimized trellis paths for several combinations of $R_X \in \{1, 2, 3\}$ and $R_B \in [0.5, 1]$; points on the convex hull of the resulting curve are shown in Figure 1. We observe that this practical coder achieves performance comparable to upper bound from the *Type 1* scheme, but the *Type 2* classification bound exceeds both by a considerable margin.

The optimization of Stage II places the location errors on occasions when they have the lowest impact on distortion. For the case of $R_X = 1$ and $R_B = 0.5$, Figure 4 shows the distribution of the values $\{X_i\}$ when $\widehat{B}_i = B_i$ and when $\widehat{B}_i \neq B_i$. As expected, most location errors occur for small values of X_i.

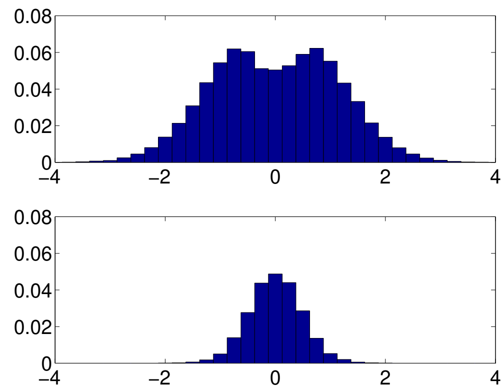

Fig. 4. *Distribution of* $\{X_i\}$ *when location is coded correctly (top) and incorrectly (bottom).*

5. Theoretical bounds

The above coding schemes provide us a way of obtaining upper bounds for the optimal rate distortion curve for the process $\{Y_i\}$. These bounds allow us to compare the relative performance of the coding strategies. No information is given, however, regarding their performance when compared to the best coder possible. In this section we attempt to answer this question.

For discrete source and decoder alphabets and an arbitrary distortion measure, the Blahut algorithm [5] allows us to find the R-D curve with arbitrary precision using a finite algorithm. For a continuous source, an analogue of the algorithm also exists, however it is not well suited to implementation on a digital computer. One other possibility is to use a discrete approximation of the source alphabet and obtain an upper bound for the R-D curve.

To obtain an upper bound for the R-D curve we consider a finite partition $\{A_i\}$ of the source alphabet (in our case the axes of $\mathbb{R}^2$), and a finite decoder alphabet $\{d_i\} \in \mathbb{R}^2$. The points in the decoder alphabet should be representative of $\mathbb{R}^2$, but in practice it suffices to consider points close to the axis. We define the distortion measure carefully, in order to get an upper bound: for a symbol Y in the source alphabet belonging to the partition set A_i and a symbol d_j in the decoder alphabet we define the distortion as the maximum distortion between a symbol in A_i and d_j, that is $d(Y, d_j) = \max_{y \in A_i}(y - d_j)^2$. This definition is reasonable as long as the sets A_i are bounded. We do not encode symbols in the unbounded sets, taking into account the extra distortion introduced by such a practice.

Using this technique we obtain a tight upper bound for the R-D curve of the process, depicted in Figure 1. We observe that although the *Type 2* bound is very close to this upper bound there is still some room for improvement (approximately 0.4dB at the rate $R = 1$).

6. Conclusion

This paper considers one of the most basic processes having joint location and magnitude uncertainty. Our R-D curves demonstrate that the naïve approach of sending location losslessly is very close to optimal when coding at high bitrates, but is far from optimal for low bitrate coding. Intuitively, the low bitrate regime presents a conflict between two obvious observations - a) magnitude is not very useful without knowing where to put it, and b) location is not very useful without knowing what value to put there. Our experiments show that intelligent strategies to manage this conflict can yield significant gains. Each strategy is viewed as an approach to balancing the coding of two random processes - one representing location and one representing magnitude. We first consider coding them independently, while optimizing the allocation of bitrate to each. While this improves on a non-optimal allocation, improved strategies are based on the observation that the reliability of coding any given location should depend on the magnitude at that location. Thus, the two processes should be coded dependently. Our analysis of strategies for dependent coding verifies this claim, and we also demonstrate a practical TCQ-based coding scheme for dependent coding of the two processes.

Ultimately, we are not certain about how close our strategies come to the optimal curve $D^*(R)$. Ongoing work focuses on deriving a tight lower bound for $D^*(R)$ using Blahut's algorithm. In addition, we believe that some insight about more effective coding strategies may be gained by examining the distribution of the output alphabet obtained from Blahut's algorithm. Finally, we plan to generalize our results to larger classes of signals, including more pertinent examples from image compression.

7. References

[1] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Proc.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[2] A. Cohen, I. Daubechies, O. G. Guleryuz, and M. T. Orchard, "On the importance of combining wavelet-based nonlinear approximation with coding strategies," *IEEE Trans. on Information Theory*, vol. 48, no. 7, July 2002.

[3] Claudio Weidmann and Martin Vetterli, "Rate-distortion analisys of spike processes," in *Proc., IEEE Data Compression Conference – DCC '99*, Snowbird, Utah, 1999.

[4] Claudio Weidmann and Martin Vetterli, "Rate distortion behavior of sparse sources," submitted to IEEE Trans. on Information Theory, October 2001.

[5] Richard E. Blahut, "Computation of channel capacity and Rate-Distortion functions," *IEEE Trans. on Information Theory*, vol. 18, no. 4, July 1972.

[6] Michael W. Marcellin and Thomas R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. on Communication*, vol. 38, no. 1, January 1990.