In Proc. of the 29th Asilomar Conf. on Signals, Sys. and Computers, Oct.-Nov. 1995

Adaptive Quantization without Side Information Using Scalar-Vector Quantization and Trellis Coded Quantization*

Youngjun Yoo and Antonio Ortega
Signal and Image Processing Institute
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-2564

Abstract

We combine backward adaptive quantization with the scalar-vector quantizer (SVQ) and the trellis coded quantizer (TCQ) both of which have an underlying scalar quantizer (USQ) in their structure. The resulting adaptive scalar-vector quantizer (ASVQ) and adaptive trellis coded quantizer (ATCQ) redesign the USQ based on the past quantized outputs. The adaptive quantizers require no side information and also outperform the SVQ and the TCQ, respectively, when the input signal is non-stationary. For an input sequence from a bimodal source switching infrequently between two Gaussian distributions with the same mean and different variances, both adaptive quantizers achieve performance gains of more than 1.3dB over the non-adaptive quantizers designed on the training set from the same bimodal source. Also the adaptive quantizers demonstrate minimal performance degradation due to adaptation when stationary inputs are considered.

1 Introduction

Most currently used quantization schemes are designed based on training sets and/or models of the input. The performance of quantizers using such *a priori* knowledge of the input is largely affected by the choice of the training set or the input model. However, in practice, it may be hard to have a good training set or sufficient knowledge on the input model. Thus there exists a motivation for adaptive quantization schemes which do not require any (or as little as possible) *a priori* information on the signal of interest.

We can categorize adaptive quantization schemes into two broad classes [1]: forward adaptation and backward adaptation. In forward adaptive quantization, the encoder makes a decision on how to update the quantizer based on current and future inputs. Thus side information has to be sent to the decoder to specify the changes.

In backward adaptation, the quantizers are updated based only on the previously quantized data. While this approach has the drawback of requiring encoder and decoder to have similar complexities, it also has the advantage of avoiding the need for overhead information transmission to the decoding end. In the remainder of this paper we will concentrate on backward adaptive quantization which, for convenience, we will refer to as adaptive quantization.

Early examples of adaptive quantizers [2, 3] focus on the dynamic range adaptation of a uniform quantizer based on past quantized data. Recent work has extended the adaptation to include updating of both dynamic range and bin sizes for a scalar quantizer [4]. This quantizer can "learn" the distribution of a stationary memoryless source and, therefore, performs well in the presence of long range dependencies of data as, for example, in a bimodal distribution.

The goal of this paper is to use the adaptive quantization technique of [4] as a building block in quantization schemes, such as the scalar-vector quantizer (SVQ) [5] and the trellis coded quantizer (TCQ) [6], which are constructed based on an underlying scalar quantizer (USQ). Our motivation is to demonstrate how adaptivity can be added and provide good results for popular quantization techniques such as TCQ and SVQ both of which have useful properties. We will introduce the adaptive scalar-vector quantizer (ASVQ) and the adaptive trellis coded quantizer (ATCQ) where we will use the previously quantized data to update USQ's of the SVQ and the TCQ.

The SVQ introduced in [5] approximates the performance of the entropy constrained scalar quantizer (ECSQ) while quantizing the input vectors at a fixed rate and retaining structural and computational simplicity. The fixed rate approach is attractive to avoid the potential problems of transmitting variable rate quantizer data over channels with error. The popularity of the TCQ [6] stems from the fact that it can outperform scalar quantizers with an encoding complexity which is still far less than that of vector quantizers.

The paper is organized as follows: In Section 2, we introduce the adaptive scalar quantization scheme used. Then we explain how adaptivity can be combined with the SVQ and the TCQ in Section 3. Experimental results with the ASVQ and the ATCQ are given in Section 4.

2 Backward adaptive quantization

*This work was supported in part by the National Science Foundation under award MIP-9502227 (CAREER).

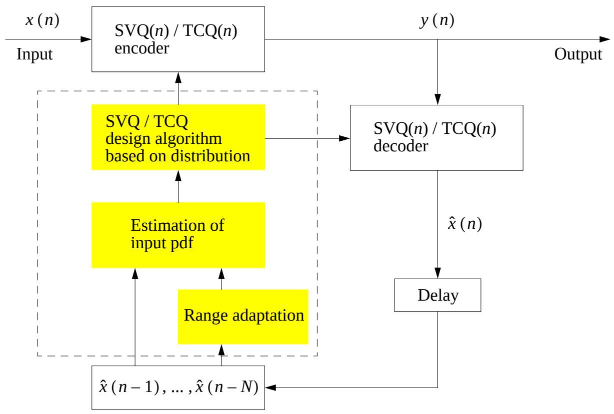


Figure 1: Block diagram of backward adaptation. In the diagram, we assume SVQ or TCQ to quantize the input though, in general, we can employ any quantizers for the purpose of actual quantization of input as long as the quantized data can be used to estimate the input distribution.

Early works on adaptive quantization mainly concentrated on the problem of adapting the dynamic range of a uniform quantizer according to changes in the input [2, 3]. Clearly, there should be a room for improvement in performance if the quantization levels and thresholds can be also updated on the fly. This was the motivation of the algorithm of [4] which will be used in this work.

Figure 1 shows the block diagram of the proposed adaptive quantizer based on the SVQ or the TCQ. The adaptation algorithm is composed of two basic building blocks, namely, *model estimation* and *quantizer design* as in [4]. The basic idea is to use past quantized data to estimate the input probability density function (pdf). Each time a new estimate of the pdf is obtained the parameters of the quantizer are updated. For that purpose, simplified versions of the quantizer design techniques for the SVQ and the TCQ are used at the encoder. The next input is then quantized using the updated quantizer. Depending on the required adaptation speed, we keep N past quantized output in the memory where N is called the adaptation window size. Note that we can either use a fixed value for N or determine N on the fly by monitoring the change in the estimated input distribution. An extensive treatment on the pdf estimation based on the quantized past can be found in [4]. Here we briefly sketch the main ideas.

Given M quantization levels r_i, $i = 0, \cdots, M-1$, with $M-1$ decision thresholds b_i, $i = 1, \cdots, M-1$, and the N most recent quantized sample occurrences, let n_i be the number of samples which fell into the i-th bin $[b_i, b_{i+1})$ for $i = 0, \cdots, M-1$, where $b_0 = -\infty$ and $b_M = \infty$. Then we use the normalized histogram

$$\left\{\frac{n_0}{N}, \ldots, \frac{n_{M-1}}{N}\right\}$$

to deduce the probability mass function P_i of the i-th bin such that

$$P_i = \int_{b_i}^{b_{i+1}} f(x)\,dx = \frac{n_i}{N}, \quad i = 0, \cdots, M-1\ , \quad (1)$$

where $f(x)$ is assumed to be the pdf of the input source. P_i's are used to determine $\hat{f}(x)$, the estimate of $f(x)$.

To simplify the problem, we assume a piecewise linear function for $\hat{f}(x)$. Hence we can determine $\hat{f}(x)$ which also needs to satisfy

$$\int_{b_i}^{b_{i+1}} \hat{f}(x)\,dx = P_i, \quad i = 0, \cdots, M-1\ , \quad (2)$$

by evaluating $\hat{f}(x)$ at $x_0, \cdots, x_{M-1}$, where we choose

$$x_i = \frac{b_i + b_{i+1}}{2}, \quad i = 0, \cdots, M-1\ . \quad (3)$$

Here we need to further assume that $\hat{f}(\hat{b}_0) = \hat{f}(\hat{b}_M) = 0$ to completely specify $\hat{f}(x)$ in $[\hat{b}_0, \hat{b}_M]$, the support of $\hat{f}(x)$, which can be found from the range adaptation algorithm. Then $\hat{f}(x)$ at $x \neq x_i, i = 0, \cdots, M-1$, is determined as the linear interpolation of the data points

$$\{\hat{f}(\hat{x}_0) = 0, \hat{f}(x_1), \cdots, \hat{f}(x_{M-1}), \hat{f}(\hat{x}_M) = 0\}.$$

For the estimation of dynamic range of the input distribution, we use a method different from that of [4]. We first detect a new statistical trend in the most recent quantized data by observing the empirical entropy (or sample entropy) determined from the histogram of the latest data; then, if the change in the empirical entropy is significant, i.e., if there is more change than a prespecified threshold, we turn on the range adaptation algorithm of [2].

The range adaptation algorithm in [2] uses a simple decision rule to adjust the dynamic range of the uniform scalar quantizer: With appropriately defined notions of the *inner* and *outer* bins, we expand the dynamic range if the last quantizer output belongs to one of the outer bins; otherwise, we reduce the range. The quantization levels are also adjusted proportionally to the change in the dynamic range.

3 Adaptive scalar-vector quantizer and adaptive trellis coded quantizer

The adaptive quantizer of [4] achieved better performance than both the fixed Lloyd-Max quantizer and the fixed ECSQ for non-stationary input sources while showing minimal performance degradation for stationary inputs. Our motivation is to apply the adaptive quantization technique to a vector quantizer (or a block-based quantizer) designed on an underlying scalar quantizer (USQ) so that we redesign the quantizer using the estimation of the marginal input pdf. Hence we consider two quantizers whose structures are based on a USQ: the scalar-vector quantizer and the trellis coded quantizer.

3.1 Scalar-vector quantizer (SVQ)

The SVQ was first proposed in [5]. The motivation of the SVQ is to design a fixed-rate vector quantizer which can be robust against transmission error in noisy environments. While reducing the error propagation problem, the SVQ requires less search complexity than conventional vector quantizers (VQ), such as the Lind-Buzo-Gray VQ [1], due to its special codebook structure.

Figure 2 illustrates a 2-dimensional SVQ codebook constructed on the USQ for a memoryless Gaussian source. Each quantization level q_i has a length ℓ_i which is the rounded self-information, $[-\log p_i]$, of the i-th bin of the USQ, where p_i is the probability of an input component being quantized to q_i. The SVQ codebook consists of the grid points in the shaded area of Figure 2 – the darker the region, the more likely the contained codevectors are used in quantization.

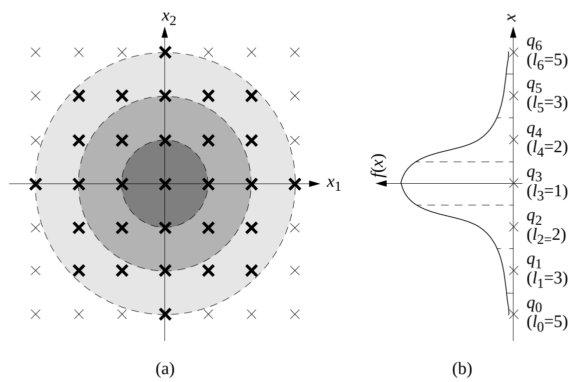


Figure 2: 2-dimensional SVQ codebook for i.i.d. Gaussian source. (a) SVQ codevectors are chosen from the 2-dimensional grid such that the codeword length is no greater than a predetermined threshold L. (b) Quantization levels and the associated lengths of the underlying scalar quantizer which determines each component of the vector grid.

In the m-dimensional case, we have n^m vector points in an m-dimensional grid which is made of n-bin USQ. If we restrict the quantization budget to r bits per sample, we have to choose 2^{rm} codevectors among n^m grid points as the reproduction levels in the SVQ codebook which can best represent the input source. Let the codevector length be defined as the sum of lengths of component quantization levels. Then the codevectors with smaller lengths are highly likely to be used in quantization. Hence we arrange n^m grid points in increasing order of codevector length and form the SVQ codebook with the first 2^{rm} points. This procedure gives a threshold L on the codevector length such that a grid point z is a SVQ codevector if and only if its length is no greater than L.

Therefore an SVQ is completely defined in terms of a triple $(\mathcal{Q}, \mathcal{L}, L)$ where $\mathcal{Q} = \{q_1, \cdots, q_n\}$ is the set of quantization levels of the USQ, $\mathcal{L} = \{\ell_1, \cdots, \ell_n\}$ is the corresponding set of lengths, and L is the threshold on the codevector length for the SVQ codebook. For the detailed description and the design algorithms of SVQ, we refer to [5]. In our adaptation scheme, we update and estimate $\mathcal{Q}$ and $\mathcal{L}$ using our estimated pdf.

3.2 Trellis coded quantizer (TCQ)

The TCQ proposed in [6] is also derived from a scalar quantizer. Motivated by the set partitioning ideas from trellis coded modulation [7], the TCQ is designed on a USQ having twice as many levels as the quantizer rate, r, allows. The levels are then partitioned into 2^{m+1} subsets where $m \leq r$. We consider here a particular case of rate-2 TCQ with 4 subsets (i.e., $m = 1$).

We include the 8-level USQ and its partition into 4 subsets in Figure 3 (b) to depict this particular TCQ. Note that each subset is assigned two quantization levels such that the average distance between the levels within a subset is maximized.

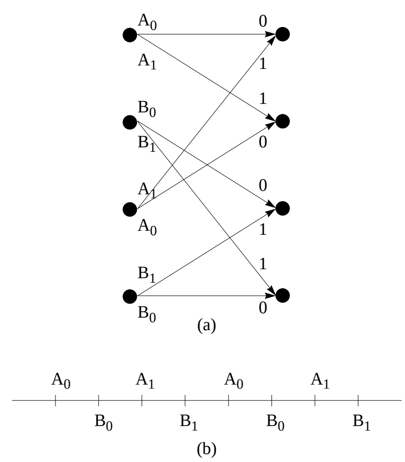


Figure 3: 4-state 8-level uniform trellis code. (a) The state transition diagram restricts the quantization of the current input depending on the quantization of the previous input. (b) Quantization levels of the underlying scalar quantizer are partitioned into two sets of $A = \{A_0, A_1\}$ and $B = \{B_0, B_1\}$ where each of the subsets A_i's and B_i's has two corresponding levels.

Since the quantizer rate is $r = 2$, we are in general allowed to use at most $2^r = 4$ distinct levels to quantize an input sample. For the TCQ, a finite state machine (or a state transition rule of 4-state trellis in Figure 3 (a)) enables the quantizer to use one of two sets $\{A_0, A_1\}$ and $\{B_0, B_1\}$ each of which contains two subsets containing a pair of levels. Since the available set of levels depends on the state of the finite state machine, the best encoding for a given source corresponds to the optimal path through the trellis. As a result, TCQ can have more degrees of freedom in choosing levels to use due to various combination of quantization level subsets along the trellis path.

The best choice of subsets for quantization of each input sample is made such that it globally minimizes the overall distortion of a given block of input samples using only the allowable transitions from the state transition diagram. The Viterbi algorithm is used to find the best trellis path which is a concatenation of state transitions through the block of samples.

Note that, since the SVQ and the TCQ have additional constraints on top of the USQ, inputs may not be always quantized to their nearest neighbor in the USQ. This will affect the our pdf slightly (the bin counts will not give an exact estimate of the probability of the source) but we have observed that the effect is negligible.

3.3 Complexity considerations

For both the ASVQ and the ATCQ, we use the empirical entropy of the output to detect the change in source statistics. We can calculate the empirical entropy at little extra cost since we keep generating the output data histogram.

The input pdf estimation and the range adaptation take only a small fraction of the overall complexity because we have an estimation strategy which uses a linear interpolation for the pdf estimate and a simple decision rule for the range adaptation. Hence, from Figure 1, most of the additional complexity of the ASVQ and the ATCQ compared to their respective non-adaptive quantizers arises from the quantizer redesign block.

In general, the SVQ and the TCQ are designed by iterative algorithms to find a set of quantization levels of the USQ based on a given training sequence. Note however that, for the ASVQ and the ATCQ, the USQ levels are found based on the piecewise linear pdf estimate. By taking advantage of this, it is possible to implement the quantizer design algorithms with much reduced complexity as compared to their TCQ and SVQ counterparts. As an example in the ATCQ case, all that is needed in each iteration is to adapt the USQ.

4 Experiments and results

We compare our adaptive quantizers to SVQ's and TCQ's designed by training. We consider two types of input sources: (i) A bimodal source obtained by switching between two i.i.d. Gaussian sources with the same mean but different variances. The transition probability between modes is 0.001; and (ii) an i.i.d. Gaussian source with parameters N(0,1).

For each source, a sequence of 40,000 samples is used as an input to both the ASVQ and the SVQ. The SVQ training sequence to obtain $\mathcal{Q}$, $\mathcal{L}$ and L consists of 100,000 samples and has the same characteristics as the simulation input. We use an adaptation window of 50 vectors, as an adaptation parameter for the ASVQ. For both quantizers, the rate is 2.0 bits per sample and the vector dimension is 8.

Figure 4 contains the plots of the SNR changes by the ASVQ and the SVQ through the whole simulation sequences and serves as the performance comparison. Figure 4(a) shows that the proposed ASVQ performs better than the trained SVQ when a non-stationary input source is applied. The average SNRs included in the figure are obtained by averaging the SNRs evaluated for sets of 50 vectors. The overall SNRs of the ASVQ and the SVQ for the whole input sequence are 8.89dB and 8.00dB, respectively. The performance loss of the ASVQ compared to the SVQ is minimal when the input is stationary, as we can see in Figure 4(b). And, in this case, the overall SNRs of the ASVQ and of the SVQ are 9.73dB and 9.96dB, respectively.

We also experimented with an ATCQ and a TCQ under the same setting as for the ASVQ/SVQ experiment. Here we use a block length of 100 samples to generate a trellis and the adaptation window of 5 blocks. Figure 5 includes the resulting plots of the SNR changes for the bimodal and the stationary Gaussian input sequences. The overall SNRs for the ATCQ and the TCQ are 9.32dB and 8.77dB for the bimodal source, and 10.03dB and 10.13dB for the stationary Gaussian source, respectively.

Finally, we summarize the experimental data in Table 1. We include, in the parentheses, the data from the ASVQ and the ATCQ which use only the range adaptation algorithm to update quantizers provided each of the quantizers has a uniform USQ. Hence, we conclude that the ASVQ and the ATCQ with the input pdf estimation can achieve performance gain over those with the range adaptation only.

References

[1] A. Gersho and R. Gray, *Vector Quantization and Signal Compression.* Boston, MA: Kluwer Academic Press, 1992.

[2] A. Gersho and D. Goodman, "A training mode adaptive quantizer," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 746–749, Nov. 1974.

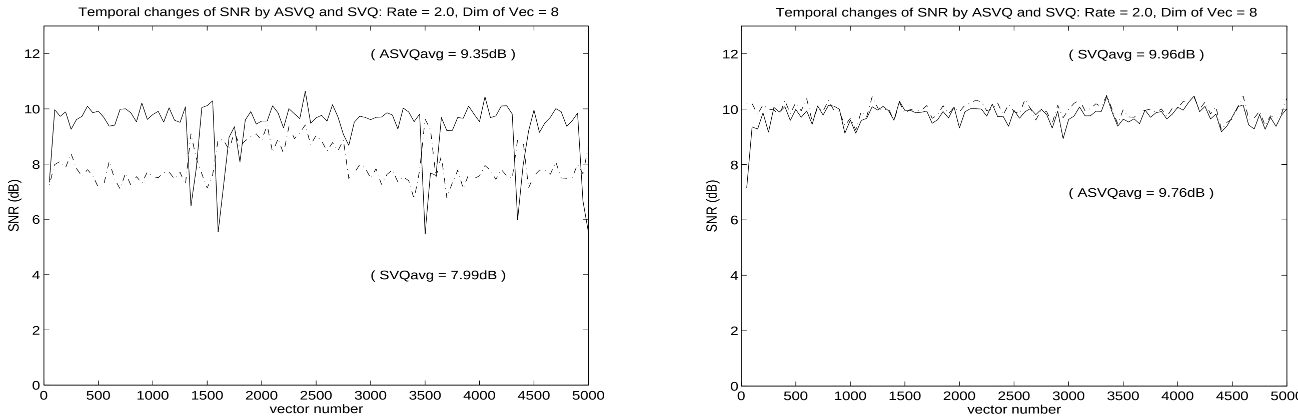
[3] N. Jayant, "Adaptive quantization with a one-word memeory," *Bell Sys. Tech. J.*, vol. 52(7), pp. 1119–1144, Sep. 1973.

[4] A. Ortega and M. Vetterli, "Adaptive quantization without side information," in *Int'l Conf. on Image Proc., ICIP'94*, vol. 3, (Austin, Texas), pp. 856–860, Nov. 1994.

[5] R. Laroia and N. Farvardin, "A structured fixed rate vector quantizer derived from a variable-length scalar quantizers: Part I - Memoryless sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 851–867, May 1993.

[6] M. Marcellin and T. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. Commun*, vol. 38, pp. 82–93, Jan. 1990.

[7] G. Ungerboeck, "Channel coding with multilevel/ phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

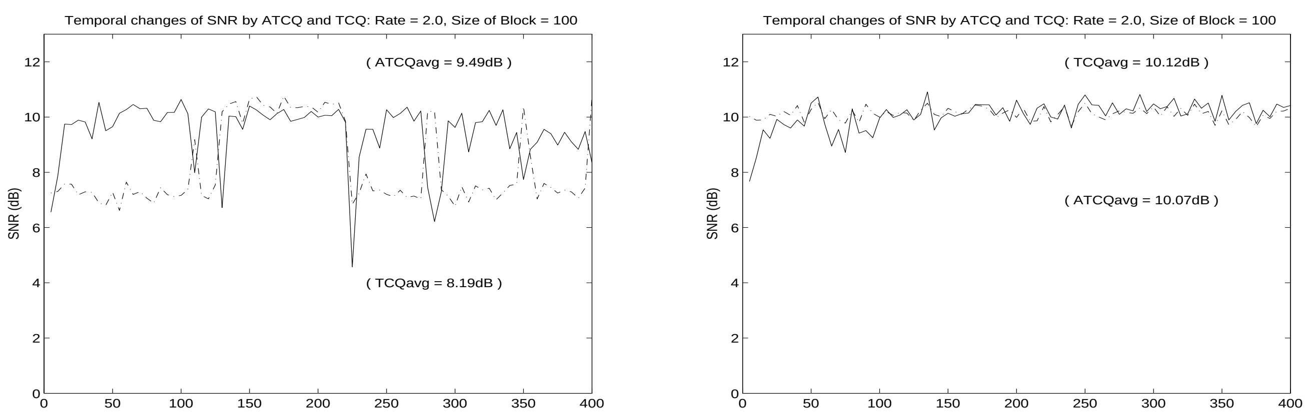


(a) (b)

Figure 4: ASVQ vs. SVQ (a) When the input is bimodal with each of the modes being Gaussian of the same mean and different variances, the performance of the ASVQ remains nearly constant throughout the sequence except at the point of mode change while SVQ performance is better for one of the two modes. (b) When the input is stationary memoryless Gaussian, there is a slight degradation in performance due to the adaptation.

Quantizer	**Nonstationary bimodal**		**Stationary Gaussian**	
	SNR_{avg}	SNR	SNR_{avg}	SNR
ASVQ	9.35 (9.24)	8.89 (8.75)	9.76	9.73
SVQ	7.99	8.00	9.96	9.96
ATCQ	9.49 (8.87)	9.32 (7.61)	10.07	10.03
TCQ	8.19	8.77	10.12	10.13

Table 1: Performance comparison of rate-2.0 quantizers for non-stationary and stationary sources. SNR_{avg} denotes the average of SNRs evaluated for blocks of input samples while SNR is the overall SNR for the whole sequence. In the parentheses are the SNRs from the adaptive quantizers using only the range adaptation on *uniform* USQ's. All results are in dB.



(a) (b)

Figure 5: ATCQ vs. TCQ (a)When the input is bimodal, the performance of the ATCQ remains nearly constant throughout the sequence except at the point of mode change and outperforms the TCQ while the TCQ performance is better for one of the two modes. (b) For a stationary memoryless Gaussian input source, the ATCQ experiences a slight degradation in performance due to the adaptation.