ADAPTIVE VECTOR QUANTIZATION OF IMAGE SEQUENCES USING GENERALIZED THRESHOLD REPLENISHMENT

James E. Fowler and Stanley C. Ahalt

Department of Electrical Engineering
The Ohio State University, Columbus, OH 43210 USA

ABSTRACT

In this summary, we describe a new adaptive vector quantization (AVQ) algorithm designed for the coding of nonstationary sources. This new algorithm, generalized threshold replenishment (GTR), differs from prior AVQ algorithms in that it features an explicit, online consideration of both rate and distortion. Rate-distortion cost criteria are used in the determination of nearest-neighbor codewords and as well as in the decision to update the codebook. Results presented indicate that, for the coding of an image sequence, 1) most AVQ algorithms achieve distortion much lower than that of nonadaptive VQ for the same rate (about 1.5 bits/pixel), and 2) the GTR algorithm achieves rate-distortion performance substantially superior to that of other AVQ algorithms for low-rate coding, being the only algorithm to achieve a rate below 1.0 bits/pixel.

1. INTRODUCTION

Over the last 20 years, vector quantization (VQ) has received significant attention as a powerful technique for data compression. VQ is theoretically attractive due to results from rate-distortion theory that show that VQ is asymptotically optimal for the coding of a data source whose statistics are stationary in time. Although VQ has been successfully applied to the coding of many types of data, including images and video, these sources can rarely be assumed to be stationary in practice, leading to a gap between the performance predicted by theory and that actually obtained in real implementations. Indeed, the nonstationary nature of the sources common in practical applications has prompted a search for more general VQ algorithms that are capable of adapting to changing source statistics as coding progresses. Such algorithms use what we call adaptive vector quantization (AVQ).

In this summary, we first present a mathematical definition of AVQ which accurately describes the operation of an AVQ communication system while being sufficiently general to apply to all previously reported AVQ algorithms. We follow with the key contribution of this work, a new AVQ algorithm called generalized threshold replenishment (GTR). The GTR algorithm differs from prior algorithms in that, 1) it is an online algorithm that does not rely on substantial buffering or iterative processing, and 2) it employs an explicit consideration of both rate and distortion, the two quantities that measure the performance of a data-compression algorithm. We conclude this summary with a sample of the experimental results obtained for the GTR algorithm. In these results, we compare the rate-distortion performance of GTR to that of several prior AVQ algorithms for an image sequence. These results show that the GTR algorithm achieves rate-distortion performance superior to that of other reported AVQ algorithms, particularly for low-rate coding.

2. ADAPTIVE VECTOR QUANTIZATION

VQ is the generalization of scalar quantization to higher dimensions [1]. Briefly, nonadaptive VQ consists of a vector quantizer, Q, that maps vectors from N-dimensional space to a fixed finite set $\mathcal{C}$ of N-dimensional vectors; i.e., $Q : \Re^N \rightarrow \mathcal{C}$. Set $\mathcal{C}$ is called the *codebook* of the vector quantizer.

Rate-distortion theory [2] states that, for a stationary, ergodic random process, there exists a rate-distortion function, $R(D)$, such that, for a given distortion D, $R(D)$ is the lower bound on the minimum achievable average rate for any coding method. In essence, the theory shows the existence of a vector quantizer that achieves this bound as the dimension of the quantizer becomes infinitely large [2].

This theoretic asymptotic optimality of VQ has inspired its use in many applications. However, most sources of practical interest are, in reality, nonstationary. A number of algorithms, known collectively as *adaptive vector quantization* (AVQ), have been introduced (e.g., [3–8]) to provide more efficient coding for these sources. These algorithms compensate for the changing source statistics associated with nonstationary sources by periodically updating the VQ codebook.

We have developed the following mathematical definition to describe in general terms the operation of these AVQ algorithms. Assume that we have a N-dimensional random-vector process, $\mathbf{X}_t$. We define *adaptive vector quantizer*, Q_t, as follows. Let $\mathcal{C}^*$ denote a large *universal codebook*, $\mathcal{C}^* \subseteq \Re^N$, that is fixed for all time t. We define a sequence of *local codebooks*, $\mathcal{C}_t$, such that

$$\mathcal{C}_t \subset \mathcal{C}^* \tag{1}$$

at each time t. We restrict each set $\mathcal{C}_t$ to be finite. Adaptive vector quantizer Q_t is a time-variant mapping from N-dimensional Euclidean space to the local codebook for

J. E. Fowler was supported by an AT&T Ph.D. Scholarship. To appear in *Proc. ICASSP*, April 1997.

time t; i.e.,

$$Q_t : \Re^N \to \mathcal{C}_t. \tag{2}$$

The output of the adaptive vector quantizer is another random-vector process,

$$\hat{\mathbf{X}}_t = Q_t(\mathbf{X}_t). \tag{3}$$

Note that, in an AVQ communication system, the encoder must transmit to the decoder not only codeword indices from the quantizer but also information describing the contents of the local codebook; this latter quantity is commonly known as *side information.* There are numerous other details of both a theoretic and practical nature involved in the construction of an AVQ system; for a more thorough investigation, consult [9].

3. THE GENERALIZED THRESHOLD REPLENISHMENT ALGORITHM

In this section, we describe our new AVQ algorithm called generalized threshold replenishment (GTR). GTR is an online algorithm that does not require large amounts of batch computation, and it employs cost criteria involving both rate and distortion measures. The GTR algorithm weighs the distortion performance against the cost in rate in both the coding of the current source vector and the updating of the local codebook.

The GTR algorithm first selects a codeword from the current local codebook as the potential coding of the current source vector by considering both the distortion between the two vectors and the rate needed to specify the codeword to the decoder. This rate is estimated from the current codeword probabilities, assuming that variable-length entropy coding of the VQ indices is used following the quantizer. Once this winning codeword is chosen, a decision rule is evaluated to see if a codebook update would result in a reduction in distortion outweighing its associated cost in rate. If so, the current source vector is added to the local codebook, replacing the winning codeword.

More specifically, the GTR algorithm operates as follows:

Given: initial local codebook, $\mathcal{C}_0$
initial codeword probabilities, $p_0(i)$, for each codeword $\mathbf{c}_i \in \mathcal{C}_0$
rate-distortion parameter, λ
windowing parameter, ω
initial time, $t = 1$

Step 1: Calculate the codeword lengths of the VQ-index entropy coder:

$$l(\mathbf{c}_i) = -\log_2 p_{t-1}(i).$$

Step 2: Find the distortions between each codeword $\mathbf{c}_i \in \mathcal{C}_{t-1}$ and current source vector $\mathbf{X}_t$:

$$\delta(\mathbf{c}_i) = d(\mathbf{c}_i, \mathbf{X}_t).$$

Step 3: Calculate the cost function for each codeword:

$$J(\mathbf{c}_i) = \delta(\mathbf{c}_i) + \lambda \cdot l(\gamma_t(\mathbf{c}_i)).$$

Step 4: Find the winning codeword:

$$\mathbf{c}^* = \arg \min_{\mathbf{c} \in \mathcal{C}_{t-1}} J(\mathbf{c}).$$

Let the index of $\mathbf{c}^*$ be denoted i^*.

Step 5: Calculate the distortion improvement and rate cost of a codebook update, as well as the update cost function:

$$\Delta d = -\delta(\mathbf{c}^*), \qquad \Delta r = l(\mathbf{X}_t),$$
$$\Delta J = \Delta d + \lambda \cdot \Delta r,$$

where $l(\mathbf{X}_t)$ is the number of bits needed to send $\mathbf{X}_t$ to the decoder.

Step 6: Set $\mathcal{C}_t = \mathcal{C}_{t-1}$. If $\Delta J < 0$, go to Step 6a. Else, go to Step 6b.

Step 6a: Set $\mathbf{c}^* = \mathbf{X}_t$ in $\mathcal{C}_t$. Send to the decoder $\mathbf{X}_t$, entropy-coded index i^*, and a flag indicating a codebook update. Go to Step 7.

Step 6b: Send the entropy-coded index i^* and a flag indicating no codebook update.

Step 7: Estimate the new codeword probabilities:

$$p_t(i) = \begin{cases} [\omega p_{t-1}(i)] / (\omega + 1), & i \neq i^*, \\ [\omega p_{t-1}(i) + 1] / (\omega + 1), & i = i^*. \end{cases}$$

Step 8: Set $t = t + 1$ and go to Step 1.

The parameter λ controls the tradeoff between rate and distortion within the algorithm, ultimately determining the rate-distortion performance of the algorithm. Larger values of λ focus the efforts of the algorithm on minimizing rate over distortion, whereas smaller values of λ result in performance with lower distortion at a higher rate. The *windowing parameter*, ω, controls the relative weighting of the past versus the present in the time-average estimates of the current codeword probabilities. The value of ω has been determined to be noncritical in the performance of the algorithm [9]; we will use $\omega = 100$ throughout the experimental results of the next section. Finally, $l(\mathbf{X}_t)$ is the length in bits of the representation of current source vector $\mathbf{X}_t$ sent to the decoder in the case of a codebook update. For the results of the next section which involve image data, we assume that a uniform scalar quantizer is used so that $l(\mathbf{X}_t) = 8$ bits per vector component. More complicated schemes are discussed in [9].

We have investigated two versions of our GTR algorithm: the basic algorithm, described here, and a slightly more complicated move-to-front variant. The move-to-front variant features the shuffling of the codebook after each source vector is coded so that the winning codeword, or, in the case of a codebook update, the new codeword, is in the front of the codebook. The brevity of this summary requires the omission of further details on the move-to-front variant of the algorithm, which are given elsewhere [9]. However, it has been shown that the move-to-front GTR algorithm achieves a slight improvement in rate-distortion performance over the basic algorithm [9]. Consequently, the move-to-front variant is used in experimental evaluation of the GTR algorithm, the topic of the next section.

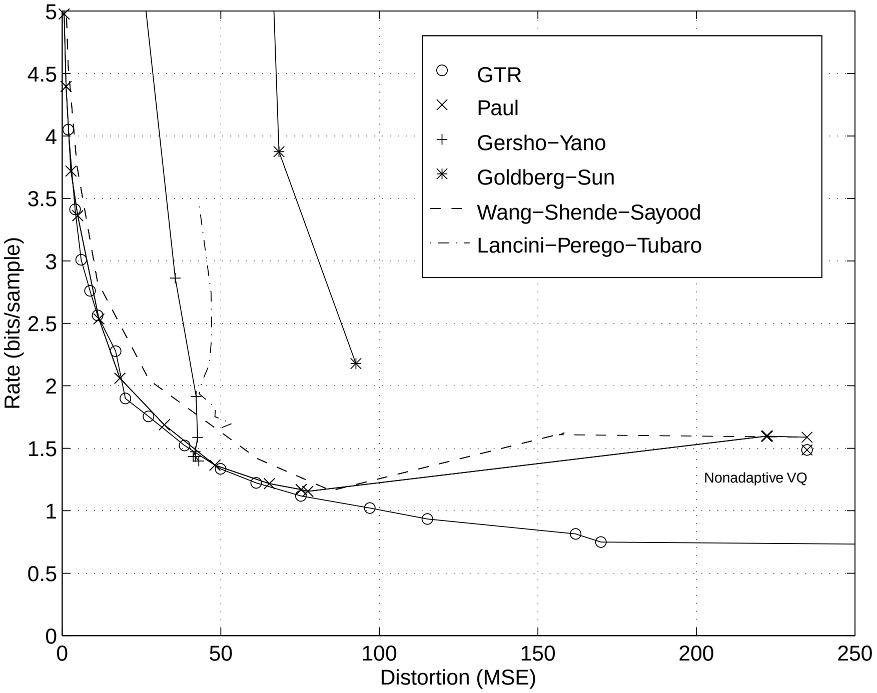


Figure 1: The rate-distortion performance of various AVQ algorithms [3–7] for the image sequence using 4-dimensional vectors and a local codebook of 256 codewords. The circled "×" represents the operation point of the nonadaptive vector quantizer for the same data.

4. EXPERIMENTAL RESULTS

In this section, we present the results obtained for the coding of an image sequence consisting of 8 image frames, 4 frames from the image sequence "Miss America" followed by 4 frames from the "Garden" sequence. Each image of the sequence is grayscale with 256 levels and has a resolution of 352 × 240 pixels. To produce an initial local codebook, we use an additional frame from the "Miss America" sequence as a training data set to the generalized Lloyd algorithm.

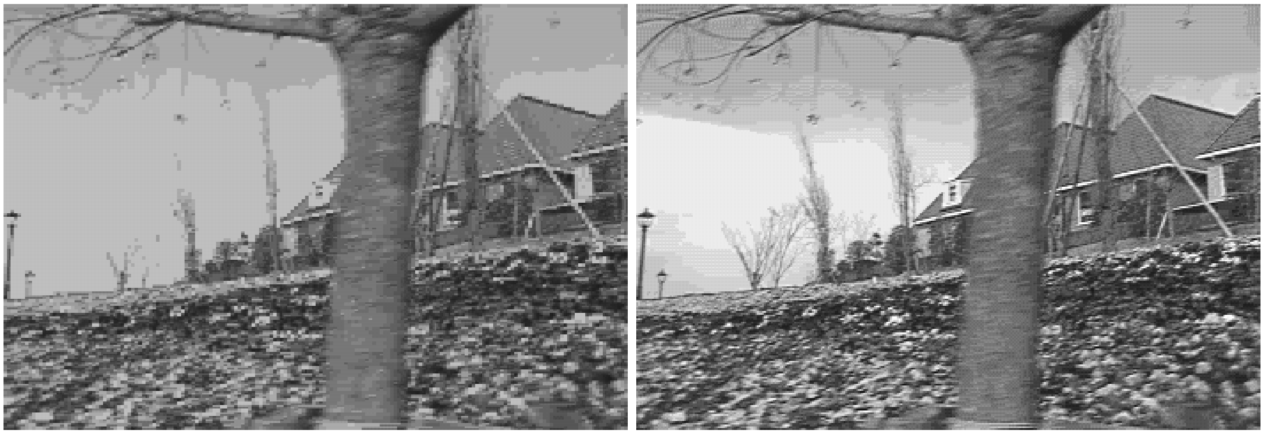
In Fig. 1, we plot the rate-distortion performance of various AVQ algorithms. We show also the performance of nonadaptive VQ. The nonadaptive vector quantizer uses the initial codebook for the entire image sequence, and its rate is the first-order entropy of the VQ indices. We present the final image of the quantized sequence for both the nonadaptive vector quantizer and the GTR algorithm in Figs. 2(a) and 2(b), respectively. For Fig. 2(b), we choose λ so that the GTR algorithm operates at approximately the same rate as the nonadaptive vector quantizer.

5. CONCLUSIONS

For low compression ratios (a rate of 1.5 bits/pixel or greater), most of the AVQ algorithms have distortion performance significantly better than that of nonadaptive VQ. For example, at a rate equal to that of the nonadaptive vector quantizer (about 1.5 bits/pixel), most of the AVQ algorithms achieve an MSE of around 50, which results in very little visual distinction between their quantized images. However, the AVQ algorithms achieve substantially less distortion than nonadaptive VQ at this rate. This improvement in image quality due to AVQ is illustrated in Figs. 2(a) and 2(b), where the quantized image of the GTR algorithm has much less visual distortion, particularly for edges and other areas of high detail, than the corresponding image for the nonadaptive vector quantizer.

More distinction between the AVQ algorithms is observed as the compression ratio increases. Particularly, several algorithms were unable to achieve rates below about 1.5 bits/pixel. Of those algorithms that were able to produce a coding at a rate below 1.25 bits/pixel, only the GTR algorithm was able to maintain a monotonic decrease in rate for increasing distortion. As a consequence, GTR was the only algorithm to achieve a coding at a rate less than 1.0 bits/pixel.

Our experimental results show that the GTR algorithm is well suited to low-rate coding applications. In particular, we have shown here that our GTR algorithm features rate-distortion performance superior to not only that of nonadaptive VQ but also that of other AVQ algorithms for the low-rate coding of an image sequence. Although beyond the scope of this summary, similar results have been obtained for other data sources [9].



(a) Nonadaptive VQ (4-dimensional vectors, 256 codewords, MSE = 234.9, 1.487 bits/pixel).

(b) GTR algorithm (4-dimensional vectors, 256 codewords, $\lambda = 16$, MSE = 38.6, 1.522 bits/pixel).

Figure 2: The final frame of the image sequence

6. REFERENCES

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Kluwer international series in engineering and computer science, Norwell, MA: Kluwer Academic Publishers, 1992.

[2] T. Berger, *Rate Distortion Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1971.

[3] D. B. Paul, "A 500-800 bps Adaptive Vector Quantization Vocoder Using a Perceptually Motivated Distance Measure," in *Conference Record, IEEE Globecom*, pp. 1079–1082, 1982.

[4] A. Gersho and M. Yano, "Adaptive Vector Quantization by Progressive Codevector Replacement," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 133–136, May 1985.

[5] M. Goldberg and H. Sun, "Image Sequence Coding Using Vector Quantization," *IEEE Transactions on Communications*, vol. COM-34, pp. 703–710, July 1986.

[6] R. Lancini, F. Perego, and S. Tubaro, "Neural Network Approach for Adaptive Vector Quantization of Images," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, (San Francisco, CA), pp. 389–392, March 1992.

[7] X. Wang, S. Shende, and K. Sayood, "Online Compression of Video Sequences Using Adaptive VQ Codebooks," in *Proceedings of the IEEE Data Compression Conference* (J. A. Storer and M. Cohn, eds.), (Snowbird, UT), pp. 185–194, IEEE Computer Society Press, 1994.

[8] M. Lightstone and S. K. Mitra, "Adaptive Vector Quantization for Image Coding in an Entropy-Constrained Framework," in *Proceedings of the International Conference on Image Processing*, vol. 1, (Austin, TX), pp. 618–622, November 1994.

[9] J. E. Fowler, *Adaptive Vector Quantization for the Coding of Nonstationary Sources.* Ph.D. dissertation, The Ohio State University, 1996.