PREDICTIVE VECTOR QUANTIZATION OF IMAGES USING A CONSTRAINED TWO-DIMENSIONAL AUTOREGRESSIVE PREDICTOR

V. John Mathews, Randall W. Waite and Thao D. Tran

Department of Electrical Engineering
University of Utah
Salt Lake City, Utah 84112

ABSTRACT

A novel approach to image compression using vector quantization of linear (one-step) prediction errors is presented in this paper. In order to minimize the image reconstruction error, we choose the optimum predictor coefficients (in a least-squares sense) that satisfy the additional constraint that the energy of the impulse response function of the inverse reconstruction filter is bounded by a small constant c. Further, the code vectors are selected such that the reconstruction error is minimized, rather than the quantization noise for the prediction error sequences. Examples demonstrating the excellent quality of the reconstructed images using our approach at bit rates below 0.65 bit/pixel are presented.

I. INTRODUCTION

Traditional methods of image compression have been built around scalar predictive and/or transform coding techniques [4, 7]. Vector quantization and related techniques are receiving increased attention because of their ability to achieve data rates that are fractions of a bit/sample without affecting the visual quality of the images drastically [1-3, 6, 9]. Our paper is concerned with a novel approach to image data compression using vector quantization of (one-step) linear prediction error sequences. As the results show, this method performs better than other algorithms of similar complexity available in literature.

The basic ideas involved in the Scalar Predictive Vector Quantization (SPVQ) algorithm may be briefly described as follows:

The image to be compressed is processed through a simple linear predictor and the resulting prediction error sequence is vector quantized. At the receiver, the vector quantized error sequence is passed through an appropriate reconstruction (inverse) filter to recreate a quantized version of the original image.

Comparing our scheme with traditional predictive quantization schemes (see Fig. 1), we find that the major difference is the fact that our scheme does not guarantee that the input signal to the reconstruction filter is the same as the output signal of the predictor. Because of this, the reconstruction errors will be larger than the quantization errors. Depending on the nature of the predictor (and the corresponding inverse filter), this increase in error can be very large. In order to minimize the effect of this additional noise, our algorithm does the following two things:

1. Instead of using a prediction filter designed to minimize the prediction error power, we will use a "constrained" predictor which is designed to minimize the prediction error power, subject to the constraint that the total energy of the inverse filter (i.e., the sum of squared values of the unit impulse response function of the inverse filter) is bounded by a small constant c. This constraint will ensure that the increase in the reconstruction noise power is not very large. For example, if we assume that the quantization noise sequence is white, it is easy to show that the reconstruction noise power is c times the quantization noise power. Thus the "constrained" predictor design helps reduce the variability of reconstruction error. Our experience is that a choice of c = 2.0 works well for images; i.e., the prediction error sequence is close to the unconstrained prediction error sequence, and at the same time, the reconstruction error is small.

2. The prediction error sequence is quantized using a distortion criterion for the reconstructed signals rather than the error sequence itself. That is, if $d(x,y)$ is our distortion measure and $e(n,m)$, $e_q(n,m)$, $x(n,m)$ and $x_q(n,m)$ denote the prediction error, quantized prediction error, image and reconstructed image sequences, respectively, instead of selecting code vectors that minimize $d\big(e(n,m), e_q(n,m)\big)$, we choose the code vectors that minimize $d\big(x,(n,m), x_q(n,m)\big)$. Thus, we strive not to select code vectors that produce minimum distortion encoding of the error sequence, but to select those that produce minimum reconstruction distortion of the images themselves.

The rest of the paper is organized as follows. In the next section, we describe some of the past work done in the area of image compression using predictive and/or vector quantization and then introduce the Scalar Predictive Vector Quantization algorithm. We also discuss our reasons for believing that the SPVQ method is superior to the other approaches described in this section. Section III contains the results of some experiments with the SPVQ algorithm. We summarize our results and also discuss further refinements of the SPVQ algorithm in Section IV.

II. PAST WORK AND THE SCALAR PREDICTIVE VECTOR QUANTIZATION ALGORITHM.

We will briefly review three different approaches to data compression and then describe the Scalar Predictive Vector Quantization algorithm. One of these methods was proposed for images and the other two for speech signals. Extension of the concepts involved in the last two approaches to data compression of images is straightforward.

Baker and Gray [1-3] proposed that before vector quantizing the images, the sample mean of the pixels belonging to each vector (block) ought to be removed. The mean residual vector quantization (MRVQ) and related methods proposed by them have product codebooks, one subset of the codebook for the sample mean and the other subset for the residual vector.

More recently, Cuperman and Gersho [5] proposed a vector predictive coding scheme for speech signals. The vector predictive scheme is exactly the same as in Fig. 1a, if we consider all the signals as vector quantities. Also, the predictor is a vector predictor (i.e., it predicts the next vector based on the present and past input vectors) and the quantizer is now a vector quantizer.

The third method we will discuss is that introduced by Schroeder and Atal [10] for speech signals and is known as the "Code Excited Linear Predictor" (CELP). In their approach, they use residual codebooks which consist of a fairly large number of code vectors that are very long (in [10] they used a vector length of 40 and a codebook size of 1024). Each residual code vector is passed through a synthesis filter H(z) and the output of the synthesis filter is compared with the input signal sequence. The residual code vector selected is the one that gives the minimum distortion. For each vector, both the index of the code vector and the parameters of the synthesis filter must be transmitted. Using the CELP scheme, Shroeder and Atal [10] were able to obtain "toll" quality speech at as low as 4.8 kbits/s. transmission rate.

The Scalar Predictive Vector Quantization (SPVQ) algorithm that we present next combines the good properties of all the above methods and also avoids many of the disadvantages associated with these methods. Conceptually, the SPVQ algorithm is closest to the CELP method. However, the SPVQ method can work with arbitrarily small vector sizes that also need smaller-size codebooks. This makes the approach computationally simpler than the CELP method.

Scalar Predictive Vector Quantization Algorithm.

Given an N x M image $x(n,m)$, the SPVQ algorithm consists of the following steps:

1. Partition $x(n,m)$ into smaller, nonoverlapping blocks of K x L pixels each. Compute the local mean μ_x associated with each block. Before the predictor is designed for each of these blocks, the local means must be removed from the image pixels. In our approach, instead of removing the mean values, we will remove a smoothed version of the means. For this, define a new sequence $z'(n,m)$ by replacing all $x(n,m)$ in each block by its local mean. Passing this sequence through a smoothing (low pass) filter will yield another sequence for which there is a smooth transition from one block to another. A simple smoothing filter that works well is

$$z(n,m) = (1-\gamma)^2 z'(n,m) + \gamma\big(z(n-1,m) + z(n,m-1)\big) - \gamma^2 z(n-1, m-1);\ 0 < \gamma < 1. \quad (1)$$

In Eq. 1 $z(n,m)$ is the output of the filter.

2. Obtain the mean residual sequence $y(n,m)$ by subtracting $z(n,m)$ from $x(n,m)$. Removing $z(n,m)$ instead of the actual local means from the image pixels will eliminate "blocky" reconstructed images. Let $\{a(k,\ell);(k,\ell)\epsilon\pi\}$ denote a set of predictor coefficients for the residual sequence in one block. Throughout this paper, we will work with causal predictors. The set of indices (k,ℓ), denoted by π is a finite set of non-negative integer pairs that does not include (0, 0). The transfer function of the predictor is then given by

$$H(z_1, z_2) = 1 + P(z_1, z_2), \quad (2)$$

where

$$P(z_1, z_2) = \sum_{(k,\ell)\epsilon\pi}\sum a(k,\ell)\, z_1^{-k}\, z_2^{-\ell}\ . \quad (3)$$

The transfer function of the inverse (reconstruction) filter corresponding to $H(z_1, z_2)$ in Eq. 2 is

$$H^{-1}(z_1, z_2) = \frac{1}{1 + P(z_1, z_2)}\ . \quad (4)$$

Let $r(n,m)$ denote the impulse response function of the reconstruction filter. $r(n,m)$ is a causal sequence. The energy ϵ_r of $r(n,m)$ is given by

$$\epsilon_r = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} r^2(n,m). \quad (5)$$

Select the optimum predictor coefficients $\{a^*(k,\ell)\}$ so that

$$J = \sum_{\forall(n,m) \text{ in a given block}} \sum e^2(n,m) \quad (6)$$

is minimized, where

$$e(n,m) = y(n,m) + \sum_{(k,\ell)\epsilon\pi} \sum a(k,\ell)\, y(n-k,\, m-\ell), \quad (7)$$

subject to the additional constraint that

$$\varepsilon_r < c \quad , \quad (8)$$

where c is a small positive constant.

In all the examples and derivations presented in this paper, we worked with a simple, separable predictor with transfer function

$$H(z_1,\, z_2) = \left(1 + \alpha z_1^{-1}\right)\left(1 + \beta z_2^{-1}\right) \quad (9)$$

where $|\alpha|$, $|\beta| < 1$ to ensure stability of the inverse filter.

3. Given the coefficients of the predictor, the codebook and the image sequence in any given block, the sequence can be vector quantized. The vector sizes are usually much smaller than the block size K x L. In order to vector quantize the sequence, we will pass each code vector through the reconstruction filter and the code vector chosen is that which would produce the minimum distortion between the image sequence and the output of the reconstruction filter.

Since we are using autoregressive predictors for the SPVQ algorithm, the reconstruction filters will have an infinite impulse response (IIR) structure. As a result, the optimal encoding of the residual sequence is very complex. We will now propose a suboptimal encoding procedure, that is much simpler computationally. In this approach, image vectors are encoded sequentially so that when each vector is coded, we will assume that the optimal choice of code vectors for all the previous image vectors have been made. As a result, the encoding complexity will only be proportional to the size of the codebook.

4. The indices of the code vectors and the predictor parameters along with the block means must be transmitted to the receiver. At the receiver, this information is enough to reconstruct the image by passing the code vectors through the reconstruction filters and adding to the output the smoothed mean values $\left(z(n,m)\right)$.

Several remarks are in order here.

1. Even though there is no explicit generation of the prediction error sequences, one can consider the SPVQ algorithm as a scheme where the prediction errors are first computed and then vector quantized. Vector quantization of the prediction errors introduces quantization noise in these sequences which will in general be amplified during image reconstruction. This in turn implies that the variability of the quantization noise will be larger for the reconstructed images than for the prediction error sequences. Since we would like to use relatively small vector and codebook sizes, it is very important that the variability of the reconstruction noise is minimized as much as possible. The SPVQ algorithm achieves this by the following two means.

a. The code vectors are chosen in such a way that the reconstruction error rather than the quantization error for the prediction error sequence is minimized.

b. The design of the "constrained" predictor further guarantees that the variability of the reconstruction noise is small. If we assume that the quantization noise for the prediction error sequences is white, then the reconstruction noise power is at most c times that of the prediction error quantization noise power. Actually, the reconstruction noise will be smaller than this amount due to step a. Because of this reduced variability, a small-size codebook will be able to adequately represent the image pixels involved.

2. The SPVQ algorithm has several conceivable advantages when compared with the three schemes we discussed earlier. As long as the autoregressive modeling is reasonably accurate, the prediction error sequence will have a smaller dynamic range than the mean residuals. This indicates that when the same number of code vectors are used, vector quantization of the prediction error sequences will produce smaller quantization errors than vector quantization of mean residuals as done by Baker and Gray. Also, the MRVQ and related algorithms require up to a third of the total number of bits transmitted to convey information about the block means. The amount of side information that must be transmitted for the predictor coefficients and block means in the SPVQ approach is negligible.

The vector predictive quantization algorithm [5] of Cuperman and Gerhso has all the advantages of the scalar predictive coding algorithm and also tries to take advantage of the inherent superiority of vector quantization over scalar quantization. However, in this situation, one would be predicting the vector based on previous vector inputs (equivalently, the scalar entries of the vector are predicted by samples that are possibly as far from them as the size of a vector). In most practical situations involving images, correlation of samples (after the mean is extracted) is much smaller at large distances than when they are adjacent. Thus, one can expect to make a better prediction of the image sequence using scalar prediction than by vector prediction and as a result, the overall performance of the SPVQ algorithm should be better than that of the predictive vector quantization scheme.

The code-excited linear predictor should perform very well with images. However, as pointed out earlier, the CELP is a very complex approach to data compression. To make this point more clear, let us consider a specific situation. To produce very good quality images, the CELP requires fairly large block sizes. Assuming that we use 32 x 32 blocks, the method will require a code book of approximately 2^{20} code vectors to encode the residuals using only 1/50 bit per pixel. One can see that the computational complexity involved here is extremely large. The SPVQ algorithm makes use of smaller-sized code vectors and codebooks and therefore is a much more simple approach to predictive vector quantization.

3. In conventional vector quantization schemes, the codebooks can be designed using the Linde-Buzo-Gray (LBG) algorithm [8] or one of its variations. Even though the LBG algorithm is conceptually and implementationally fairly simple, it cannot be used for designing optimal codebooks suitable for the SPVQ algorithm. This can be seen from the fact that the same image vector can be mapped into different code vectors depending on the nature of the adjoining image vectors. As a result, it is impossible to obtain a nonoverlapping partition of the training sequence so that each subset gets mapped into the same code vector. Since the LBG algorithm requires this type of a partition, it is obvious that the LBG algorithm or any of its variants cannot be used for designing optimal codebooks for the SPVQ method. Even though suboptimal, we have used codebooks designed using the LBG algorithm in this paper. They are useful mainly for two reasons:

a. If the codebook is dense enough, it is possible that the encoding based on the minimum distortion reconstruction criterion will be different from the minimum distortion encoding of the error sequence and the former encoding will fare much better than the latter approach. Our experience supports this conjecture.

b. One big advantage of the LBG algorithm is its conceptual simplicity. The fact that the training sequence can be partitioned into disjoint sets that map into a code vector is very useful. This fact enables the user to tailor the codebook to his needs. For example, it is possible to create codebooks with larger representation to edge pixels by merely having a training sequence with larger representation of edge pixels [9].

The remainder of this paper is devoted to discussing some experimental results that demonstrate clearly the ability of the SPVQ algorithm to produce high-quality images at low bit rates.

III. EXPERIMENTAL RESULTS

The image used for our experiments is termed "woman" and is shown in Fig. 2a. The image consists of 512 x 512 pixels with 8 bits/pixel resolution. The results of encoding the image using the SPVQ algorithm with 32 x 32 sub-blocks, 4 x 4 vectors, 1024 code vectors and $\gamma = 0.9$ is displayed in Fig. 2b. The bit rate for this example (including all the side information) is slightly less than 0.65 bits/pixel. We can see that the visual quality of the reconstructed image is good. Employing a widely-used definition of signal-to-quantization-noise ratio (SQR) [7] given by

$$SQR = \frac{\text{mean squared reconstruction error}}{(\text{peak-to-peak value of the image})^2}, \quad (10)$$

a quantitative measure for the SQR was obtained as 31.3 dB. Here the codebook was obtained using the LBG algorithm with a training sequence consisting of the residuals of seven images other than the "woman" image.

IV. SUMMARY AND CONCLUSIONS

In this paper we presented a novel approach to image compression using vector quantization of linear (one-step) prediction errors. Results presented in the paper demonstrate the ability of the SPVQ algorithm to produce good quality images at low bit rates. We are at present working on refining our method so as to yield even better results. Some of the areas that are being studied are the design of optimal codebooks for the SPVQ algorithm, the "constrained" predictor design for more complex structures, improved coding of edge pixels, incorporation of visual models into the data compression algorithm and further simplifications and improvements in the SPVQ encoder structure.

REFERENCES

1. R. L. Baker. Vector quantization of digital images, Ph.D. Thesis, Stanford University, June 1984.

2. R. L. Baker and R. M. Gray, "Image compression using non-adaptive spatial vector quantization," Proceedings of the 16th Asilomar Conference on Circuits, Systems, and Computers, 1982.

3. R. L. Baker and R. M. Gray, "Differential vector quantization of achromatic imagery," Proceedings of the International Picture Coding Symposium, pp. 105-106, March 1983.

4. R. J. Clarke, Transform Coding of Images, Academic Press, London, 1985.

5. V. Cuperman and A. Gersho, "Vector predictive coding of speech at 16 kbits/s," IEEE Transactions on Communications, Vol. COM-33, No. 7, pp. 685-696, July 1985.

6. R. M. Gray, "Vector quantization," IEEE ASSP Magazine, Vol. 1, No. 2, pp. 4-29, April 1984.

7. A. K. Jain, "Image data compression: A review," Proceedings of the IEEE, Vol. 679, No. 3, pp. 349-389, March 1981.

8. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Transactions on Communications, Vol. COM-28, No. 1, pp. 84-95, January 1980.

9. B. Ramamurthi and A. Gersho, "Image coding using segmented codebooks," Proceedings International Picture Coding Symposium pp. 44-45, March 1983.

10. M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): High quality speech at very low bit rates," Proceedings ICASSP 1985, pp. 937-940, Tampa, Florida, March 1985.

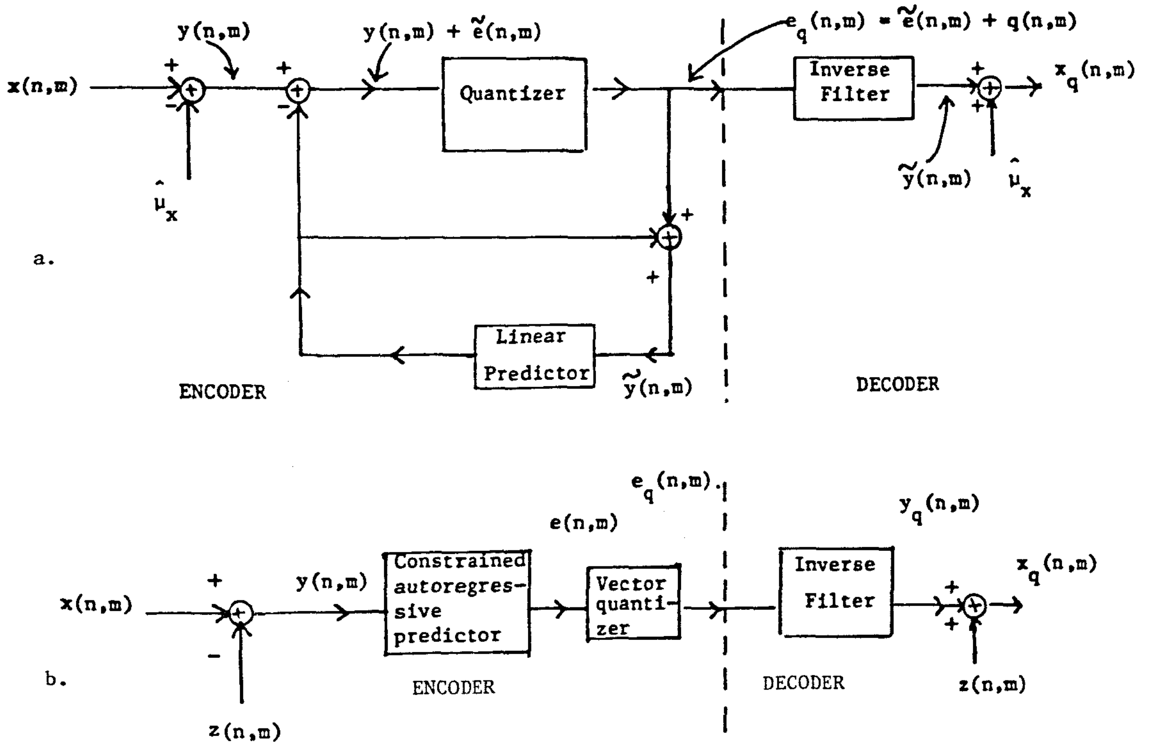

Fig. 1. a. Traditional predictive quantization. b. Predictive vector quantization algorithm presented in this paper. In actual implementation of the encoder, the prediction error sequences are not computed. The inputs to the vector quantizer are y(n,m) and the code vectors.

Fig. 2. a. Original "woman" image. b. Quantized "woman" (0.6484 bits/pixel)