Open-Loop Design of Predictive Vector Quantizers for Video Coding

Kenneth Rose, Hosam Khalil, and Shankar L. Regunathan
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106*

Abstract

The basic vector quantization (VQ) technique employed in video coding belongs to the category of predictive vector quantization (PVQ), as it involves quantization of the (motion compensated) frame prediction error. It is well known that the design of PVQ suffers from fundamental difficulties, due to the prediction loop, which have an impact on the convergence and the stability of the design procedure. In this paper we propose an approach to PVQ design that enjoys the stability of open-loop design while it ensures ultimate optimization of the closed-loop system. The method is derived for general predictive quantization, and we demonstrate it on video compression at low bit rates, where it provides substantial improvement over standard open and closed loop design techniques.

1 Introduction

Most video coding systems use predictive approaches and are, hence, composed of two main functional modules: the frame prediction module, and the prediction error (residual) compression module. The objective of the first module is to exploit the temporal redundancy in the correlation between consecutive frames. The current frame contents are predicted based on past or future observations. This module typically involves block-based motion compensation (whose parameters are transmitted as side-information) so as to achieve better approximation of the current frame. The second module is the lossy part of the codec where the prediction error, or residual, is compressed to the appropriate bit rate. The prediction residual is usually handled as a two-dimensional signal and, more specifically, as if it were a still image (intraframe coding). The predominant residual compression approach involves application of the Discrete Cosine Transform (DCT), and this is the method of choice in the major standards such as H.263 and MPEG.

An important justification for the use of DCT in still image compression hinges on the assumption that the signal can be well modeled as a Gauss-Markov process with a high autocorrelation coefficient. It has been shown that the performance of the optimal (Karhunen-Loeve) transform on such a signal is closely approximated by that of DCT. However, this argument does not hold for the prediction residual signal whose statistics are considerably different from those of a still image. It is, therefore, plausible that some other approach to residual compression, which takes into account the actual signal statistics, would provide substantial gains. An interesting example of a recent method, which departs from main-stream DCT techniques, is based on matching pursuit [1] where the residual is approximated using entries from a library of predefined two-dimensional functions.

We pursue a known alternative approach which is based on vector quantization. There are several arguments in support of VQ for video compression. Shannon's theory implies that vector quantizers are asymptotically optimal, where asymptotic here is in terms of vector length. (Note, in particular, that typical blocks in video coding correspond to long vectors.) Another important argument is that VQ is a very general framework and includes, for example, DCT compression as a special constrained case [2]. Thus, it may be argued that DCT can not outperform the best VQ.

On the other hand, there exist various serious objections to the use of VQ in video coding. The first difficulty is that of complexity. The VQ complexity grows exponentially with the product of vector dimension and rate. This led to numerous methods which constrain the VQ to reduce its search and/or memory complexity, but also inevitably compromise its performance. However, it is important to keep in mind that at very low bit rates even unconstrained VQ would be manageable. Another major objection is concerned with difficulties in the design of VQ for video coding applications. Predictive VQ (PVQ) design is problematic, and the design often fails to produce an optimal, or sometimes even a good, VQ. Another important difficulty is the variation in local statistics of the signal which need to be exploited by an adaptive system to achieve further gains.

*This work was supported in part by the National Science Foundation under grant no. NCR-9314335, the University of California MICRO program, ACT Networks, Inc, Advanced Computer Communications, Cisco Systems, Inc., DSP Group, Inc., DSP Software Engineering, Inc., Fujitsu Laboratories of America, Inc., General Electric Company, Hughes Electronics Corp., Intel Corp., Nokia Mobile Phones, Qualcomm, Inc., Rockwell International Corp., and Texas Instruments, Inc.

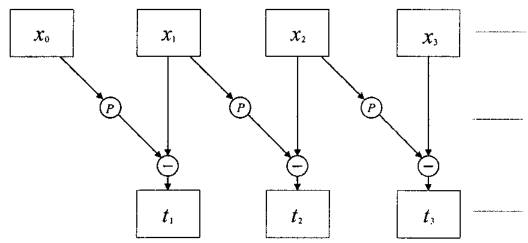

Figure 1: Open-loop procedure: x_j denotes original frame j, and t_j denotes motion compensation residual j. P represents the motion compensator.

It is our premise here that suboptimal PVQ design is a major stumbling block on the way to a truly competitive VQ approach for video coding. We hence propose to first attack this fundamental problem. In this paper, we cover traditional design methods and explain the difficulties in the training stage (Section 2). We develop a novel approach to solve the PVQ design problem (Section 3), and provide simulation results as experimental evidence that VQ is indeed an attractive approach for video coding (Section 4).

2 Conventional Predictive Vector Quantizer Design

Predictive vector quantization was introduced in [3] and was successfully utilized in the field of speech compression. A major issue in PVQ design is that of obtaining the necessary training set for the quantizer design. To clarify this difficulty consider a regular VQ system, where the quantizer design is simply based on a set of source samples. It is thus possible to iteratively adjust the quantizer parameters while decreasing the distortion, computed over the training set, until convergence. In contrast with standard VQ, the PVQ quantizer operates on the prediction error. But since the prediction is based on the reconstruction of the previous frame, it depends on the quantizer itself. In other words, PVQ is essentially a closed loop system. Clearly, the "effective training set" which is the sequence of prediction errors, is not fixed but changes every time the quantizer parameters are modified. In [3], two techniques were introduced and have since been the most widely used training algorithms. In this section, we briefly sketch these approaches. The presentation is geared toward emphasizing the unresolved issues, and highlighting the distinctions with the approach proposed in this paper.

2.1 Open-loop approach

This simple approach is depicted in Figure 1. Here, we generate a training set of prediction error vectors for the original, *unquantized* input. It is called "open-loop" because it is not the reconstructed frame which is fed back through the predictor. The feedback is eliminated during design. Specifically, given a set of original samples (or frames) $S : \{x_0, x_1, x_2, \cdots, x_N\}$, we generate the required training set via

$$t_n = x_n - P(x_{n-1}), n = 1, 2, \ldots, N, \qquad (1)$$

where P is the predictor operator.

Once the training set $T : \{t_1, t_2, \cdots, t_N\}$ is fixed, we can apply a standard VQ design technique such as the generalized Lloyd algorithm ([4]).

The open-loop approach suffers from obvious shortcomings. The decoder does not have access to the previous original sample or frame. In order to avoid the notorious decoder "drift", we must predict from the reconstructed previous frame. Thus, the training set used for the design, is statistically different from the prediction error to be quantized in practice. This statistical mismatch results in accumulation of errors through the prediction feedback, and the performance is usually poor.

2.2 Closed-loop approach

To alleviate the statistical mismatch problem of the open-loop method, a closed-loop approach was presented in [3]. Figure 2 shows the main steps. An iterative algorithm is applied whereby a closed-loop (real) system is used to generate the prediction errors. Given a quantizer at iteration $i-1$, which we denote by $Q^{(i-1)}$, a training set of prediction errors $T^{(i)} : \{t_1^{(i)}, t_2^{(i)}, \cdots, t_N^{(i)}\}$, is generated for iteration i:

$$t_n^{(i)} = x_n - P(\hat{x}_{n-1}^{(i)}) \qquad (2)$$

where

$$\hat{x}_n^{(i)} = P(\hat{x}_{n-1}^{(i)}) + Q^{(i-1)}(x_n - P(\hat{x}_{n-1}^{(i)})) \qquad (3)$$

These errors are now fixed and a new quantizer, $Q^{(i)}$, is optimized for them (via the GLA technique). Next, a new sequence of prediction errors is generated for iteration $i+1$, and so on.

The result depends on the choice of initial quantizer $Q^{(0)}$. We have taken as initial quantizer the outcome of the open-loop method. Although results are generally superior to the open-loop approach, performance is still far from optimal in video coding applications.

2.3 Summary of shortcomings

The central design difficulty in predictive quantization is that quantization errors are fed back through the prediction loop, thus making the training of the quantizer a highly unstable procedure. More specifically, the actual effective training set (the sequence of prediction errors), in a straightforward design, changes in every iteration, and the effect of quantizer adjustment on the performance is unpredictable. In particular, there is a common effect of error build-up which causes large deviations in the statistics and tends to confuse the design procedure. Closed-loop training "ignores" the above difficulty and iterates as if an improvement of the quantizer for the current set of prediction errors ensures better performance once we close the loop to produce the prediction errors for the

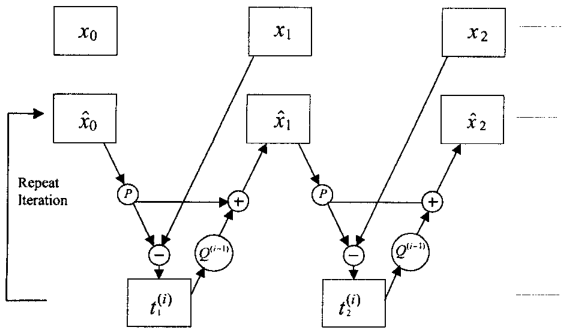

Figure 2: Closed-loop procedure: x_j denotes original frame j, $x_j^{(i)}$ denotes the j^{th} reconstructed frame at iteration i , and $t_j^{(i)}$ denotes the j^{th} motion compensation residual at iteration i. $Q^{(i-1)}$ is the vector quantizer trained on residuals from iteration $i-1$.

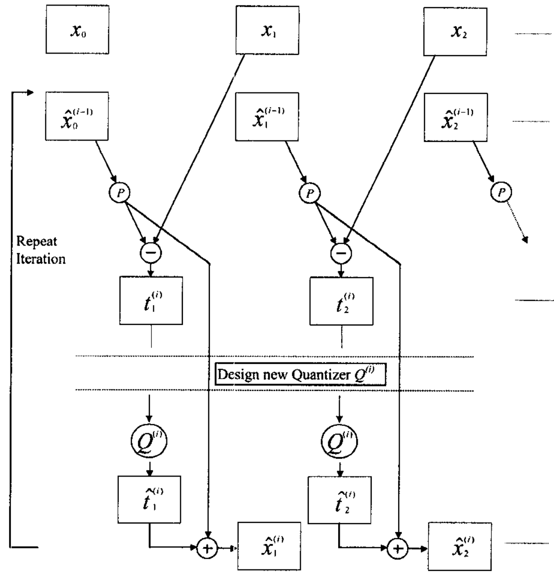

Figure 3: Proposed procedure: x_j denotes original frame j, $x_j^{(i)}$ denotes the j^{th} reconstructed frame at iteration i , and $t_j^{(i)}$ denotes the j^{th} motion compensation residual at iteration i. $Q^{(i)}$ is the vector quantizer trained on residuals from iteration i, and $\hat{t}_j^{(i)}$ is $t_j^{(i)}$ quantized by $Q^{(i)}$.

next iteration. Open-loop training, though it has a fixed training set, and hence is ensured to converge, is mismatched with the true mode of operation of the quantizer. A notable alternative closed-loop method is the stochastic approach of Chang and Gray [5]. It is, however, generally known that the problem has not been satisfactorily solved as yet [6], [7], [2].

3 Proposed Method

The objective of the proposed design approach is to enjoy the best of both worlds, namely, to enjoy the design stability of the open-loop mode while ultimately optimizing the system for closed loop operation. We achieve this with the following procedure (see also the block diagram of Figure 3):

Step 1. *Apply some initial PVQ to the training sequence of frames to obtain a reconstructed sequence, with the corresponding sequences of next-frame prediction, and prediction error.*

Step 2. *Design an optimal VQ for the given (fixed) sequence of prediction errors.*

Step 3. *Apply the optimized VQ to quantize the same prediction errors used in Step 2.*

Step 4. *Add the sequence of quantized prediction errors to the next-frame-prediction sequence to obtain a new reconstructed sequence.*

Step 5. *Use the reconstructed sequence to generate a new next-frame prediction sequence. (*No quantization*)*

Step 6. *Compute the sequence of prediction errors.*

Step 7. *Go to Step 2.*

Observations: i) The entire design is in open-loop mode. Note that we compute prediction errors for the entire sequence before quantization. ii) Steps 2, 3, and 4 ensure decrease in overall distortion as they simply apply the best quantizer to the prediction errors. iii) Step 5 does not strictly ensure decrease in distortion, but does so to the extent that smaller prediction error is expected to lead to smaller quantization error. (A typical assumption in general predictive video coding, see e.g. H.263).

As the distortion is generally decreasing, we expect the process to converge. The minor difficulty with step 5 causes a small limit cycle instead of perfect convergence but this appears to have no practical significance. Most important is to observe the implications of convergence: At convergence the reconstructed sequence is unchanged from one iteration to the next. This means that the next frame prediction would be the same even if it were based on the reconstruction of the current frame (instead of on the reconstructed frame from the previous iteration). In other words, *this is equivalent to closed-loop operation.* But the algorithm is running all the time in open loop! We thus have developed a procedure which is "open- loop" in nature, yet converges to optimization of the closed-loop performance.

We next introduce some mathematical notation and further explain the algorithm. The main objective is to avoid accumulation of errors through the prediction loop. We therefore base our prediction on the reconstructed samples of the previous iteration. The training set is , in effect, generated by

$$t_n^{(i)} = x_n - P(\hat{x}_{n-1}^{(i-1)}), n = 1, 2, \cdots, N. \quad (4)$$

Compare this equation with (1) for standard open-loop, and (2) for the closed loop design. Note further that execution of equation (4) is done for the whole sequence, without the effect of quantization error accumulation. Having collected the set of training samples, we optimize a new quantizer $Q^{(i)}$. The new quantizer is used to generate the new set of reconstruction frames based on

$$\hat{x}_n^{(i)} = P(\hat{x}_{n-1}^{(i-1)}) + Q^{(i)}(x_n - P(\hat{x}_{n-1}^{(i-1)})), n = 1, 2, \cdots, N \quad (5)$$

In effect, we are quantizing here the exact same training vectors that went into the training procedure. Neglecting the possible suboptimality of GLA itself, this is the best quantizer for these vectors. We are thus assured that we are quantizing them to the best of our resources and that the resulting reconstructed frames are improved. Under the reasonable (and common) assumption that better reconstruction provides better next-frame prediction, we get monotone improvement throughout the process.

The rationale underlying the use of reconstruction from the previous iteration for prediction becomes apparent when we consider the implications of convergence. The distortion is monotone decreasing, but the rate of improvement is diminishing as the system approaches convergence. Thus, further iterations do not modify the quantizer

$$Q^{(i+1)} = Q^{(i)} \quad (6)$$

which immediately ensures that the reconstruction sequence is fixed:

$$\hat{x}_n^{(i+1)} = \hat{x}_n^{(i)}, \quad (7)$$

the training set is fixed:

$$t_n^{(i+1)} = t_n^{(i)}, \quad (8)$$

and that the next-frame prediction sequence is fixed:

$$P(\hat{x}_{n-1}^{(i)}) = P(\hat{x}_{n-1}^{(i-1)}). \quad (9)$$

The significance of (9) is obvious: *prediction from the reconstruction of the previous iteration is equivalent to prediction from reconstructed samples in the same iteration.* In other words, although we always work in an open-loop mode, at convergence we have reached a point where the prediction is effectively the same as that of the normal closed-loop system. Thus, upon convergence we have optimized the closed-loop system.

Sequence	Coder	PSNR	Rate	$D + \lambda R$
Salesman	H.263	30.13	10.36	71.32
	PVQ	30.19	10.27	70.32
Claire	H.263	34.33	6.62	29.23
	PVQ	34.76	6.65	26.98
Akiyo	H.263	32.58	6.02	40.67
	PVQ	33.32	5.99	35.03

Table 1: Performance comparison of H.263 and PVQ for the test image sequences "Salesman", "Claire", and "Akiyo". The comparison is in terms of PSNR in dB, rate in Kb/s, and the rate-distortion Lagrangian.

4 Simulation Results

The PVQ design method has been adopted to and tested on video sequences. We implemented a video codec where 8 by 8 blocks of residuals are used as vectors. A total of 30 frames of the sequence *Carphone* were used as the training sequence. The design optimized an entropy-constrained [8] PVQ while maintaining an open loop which gradually converges to optimization of the closed-loop system, as described in the previous section. The system uses half-pixel motion compensation, and is basically a "bare-bones" H.263 scheme where the DCT/quantization module was replaced with the entropy-constrained VQ.

An entropy-constrained VQ design produces a variable codeword-length codebook. Highly probable vectors are assigned to shorter codewords than less probable vectors, and this provides for a more efficient source coder (see [2]). The quantizer design incorporates probability as follows: The codeword length for codeword i is approximated by the first-order entropy estimate $l_i = -log p_i$ where p_i is the probability of choosing codeword i (estimated from the training set). A Lagrangian $L = D_i + \lambda * l_i$ is then used to specify the total *cost* of encoding an input vector using codeword i. Here, D_i is the squared error distortion encurred when codeword i is chosen and λ is a constant multiplier that controls the Distortion/Rate tradeoff. Low bit rates command large λ. See [8],[2] for detailed algorithm.

Figure 4 depicts the comparison in performance of PVQ designed by the proposed design method with that of the standard closed-loop technique. The PSNR shown is that of the actual closed-loop performance of the coder using a PVQ obtained at each iteration and is equal to the average PSNR over the test sequence (which is the same as the training sequence, in this case). Note that both systems start their iterations by designing an open-loop-designed codebook, and thus have the same performance at the first iteration. Both systems perform better initially by reiterating but note the gradual accumulation of error in the closed-loop technique and the subsequent drop in overall PSNR of the system. On the other hand, the proposed approach shows gradual improvements, and eventually provides performance that is superior by several dB.

For reference, it should be mentioned that the corresponding "bare-bones" H.263 (with the standard

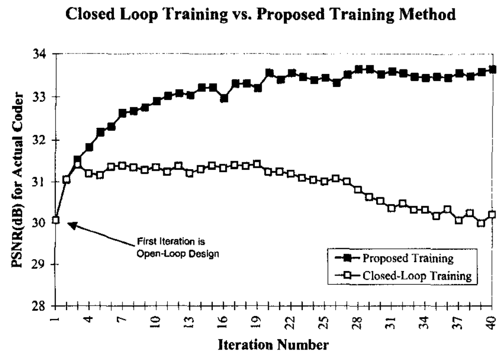

Figure 4: Performance comparison of standard closed-loop PVQ design with the proposed open-loop approach. The performance is given for real closed-loop mode operation, and for the PVQ available at the end of each iteration.

DCT module) achieved PSNR of about 31 dB which is significantly below the performance of our PVQ. The bit-rate was fixed at about 12 kb/s for coding the residual of this QCIF sequence; all other side information being the same for both coders.

So far we have considered the performance on the training set so as to emphasize the power of the proposed optimization technique as both PVQ design methods attempt to directly minimize the distortion-rate Lagrangian over the training set. Figure 4 shows clearly the enhanced power of the PVQ design technique, and the evolution through the iterations.

Tests on other sequences outside the training set also indicate improvements over H.263. Here, for the PVQ to be general enough, we used a total of 13 video sequences in the training phase. The test set is composed of the three independent video sequences *Salesman*, *Claire*, and *Akiyo*. Table 1 shows the comparison between the H.263 performance with that of the proposed PVQ. The PVQ design was stopped after about 25 iterations of the proposed algorithm. The H.263 bit rate was controlled so as to match that of the PVQ system. It can be seen that gains of 0.06 dB, 0.43 dB, and 0.74 dB can be achieved for the three sequences, respectively. Table 1 also provides the avarage lagrangian $D + \lambda * R$ which is a combined cost that takes into account both rate and distortion. Considerable improvements was obtained in all test sequences.

The design, in this case, involved two codebooks: one codebook optimized for blocks whose motion vector was zero, and another codebook optimized for blocks with nonzero motion. Note that the switching information need not be conveyed to the receiver as it is determined by the motion. (One can design more codebooks conditioned on the motion vector, but it seems that two codebooks is a reasonable compromise between compression performance and complexity, memory, and training requirements.) Codebook sizes used are about 10,000 and 2,500, respectively. It should also be noted that a fast search algorithm was developed which exploits the fact that, at low bit rates, a large number of residual blocks get quantized to zero. The exact details are omitted here.

5 Conclusions

In this paper, we have described a new approach to training predictive vector quantizers, which does not suffer from the statistical mismatch typical of open-loop training algorithms, nor from the instability experienced by closed-loop approaches. The proposed iterative algorithm is open-loop in nature but asymptotically optimizes the closed-loop system. Experimental results were first given for a simple PVQ design for video coding, and showed the superiority of the proposed method. Further experimental results were given to compare the system with standard DCT-based video coding. The preliminary simulation results provide evidence for the promise of the method.

References

[1] R. Neff and A. Zakhor, "Very low bit rate video coding based on matching pursuits," *IEEE Trans. Circuits and Systems for Video Technology*, vol 7, pp. 158-171, Feb. 1997.

[2] A. Gersho and R, M. Gray, *Vector Quantization and Signal Compression.* Boston, MA: Kluwer, 1992.

[3] V. Cuperman and A. Gersho, "Vector predictive coding of speech at 16 kbits/s," *IEEE Trans. Communications*, vol. COM-33, no. 7, pp. 685-696, July. 1985.

[4] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Communications*, vol. COM-28, pp. 84-95, Jan. 1980.

[5] P. C. Chang and R. M. Gray, "Gradient algorithms for designing predictive vector quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 679-690, Aug. 1986.

[6] H. Kwon, M. Venkatramam, and N. M. Nasrabadi, "Very low bit-rate video coding using variable block-size entropy-constrained residual vector quantizers," *IEEE Journal Selected Areas in Communications*, vol. 15, no. 9, Dec. 1997.

[7] R. Baker and J. Salinas, "A motion compensated vector quantizer with filtered prediction," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 1324-1327, 1988.

[8] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy constrained vector quantization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-37, pp. 31-42, Jan. 1989.