TREE-STRUCTURED VECTOR RESTORATION CODING OF VIDEO

J. S. McVeigh[*], *S.-W. Wu*[**], *M. W. Siegel*[*] *and A. G. Jordan*[*]

[*]Carnegie Mellon University, Pittsburgh, PA 15213

[**]AT&T Bell Laboratories, Murray Hill, NJ 07974

1. INTRODUCTION

The predictive coding of video typically entails a motion estimation and compensation step followed by lossy encoding of the residual between the original and predicted images. While this strategy has been shown to be effective for moderate to high rate video compression applications (e.g., the MPEG-1 video coding standard [1]), it neglects the fact that significant correlation may exist between the predicted and residual images. To achieve superior rate-distortion performance, this correlation should be exploited in applications that require high compression ratios.

Recently, we have presented preliminary results on a new predictive coding technique that we call vector restoration, that adaptively exploits the non-trivial correlation between the predicted and residual signals [2]. Vector restoration (VR) is closely related to the technique of residual vector quantization (VQ) [3]. We previously have shown that VQ is a special and suboptimal case of VR [2]. Experiments showed performance gains over VQ and a transform-based technique for the low bit rate compression of video, at the expense of increased encoder and decoder complexity.

While our initial study on full-search vector restoration with unstructured codebooks yielded encouraging results, the complexity of VR will have to be reduced for it to become a viable compression technique. Towards this end, in this article we extend the extensively researched concepts of tree-structured codebooks and optimal codebook pruning [4, 5] to vector restoration. We provide a brief discussion on the relationship between pruned tree-structured VR and VQ, and we include experimental comparisons between these techniques for the compression of several standard video-conferencing sequences.

2. REDUCED COMPLEXITY VECTOR RESTORATION

The core portion of the vector restoration coder is depicted in Fig. 1. A finite set restoration functions are designed off-line and are stored at the encoder and decoder. For each original-predicted vector pair, the function that yields minimum distortion is selected and is applied to the predicted vector at the decoder to *restore* the original vector.

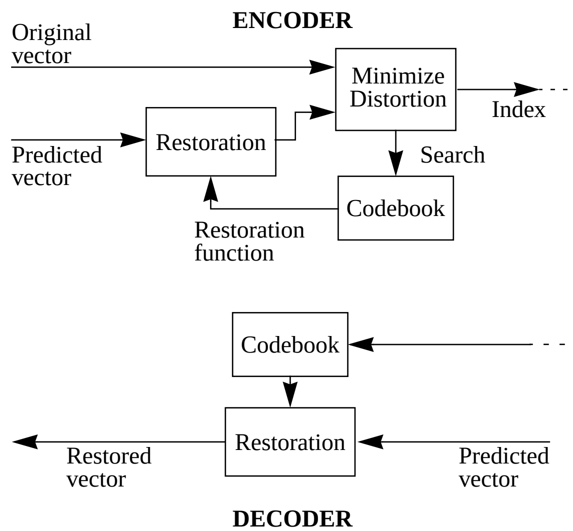

Figure 1: Vector restoration based coding

For mathematical tractability, we restrict the restoration functions in the VR codebook to affine transformations, consisting of a linear estimation and a translation portion. In keeping with our desire of reduced coder complexity, we apply the constraint of separability on the linear estimation portion of the function. This constraint reduces both the storage requirement of the VR codebook and the number of multiplications and additions that must be performed by the restoration operation[1]. While in theory separability reduces the performance of the VR coder, in practice we have observed that separable transforms are more robust and often outperform unconstrained restoration functions

1. To realize the multiplication and addition reductions, the restoration function must be applied in block-matrix form. For simplicity of notation, we present the equivalent vector form obtained by row-ordering the corresponding blocks.

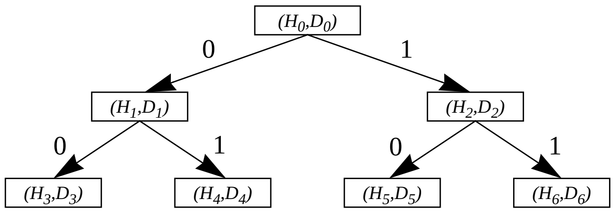

Figure 2: Balanced binary tree structure.

on vectors outside the training set. For an *n*-dimensional original vector X, and an *m*-dimensional predicted vector $\tilde{X}$, the restored vector $\hat{X}$ obtained by using the i^{th}-indexed restoration function, is given by,

$$\hat{X} = (A_i \otimes B_i)\tilde{X} + D_i \tag{1}$$

where the separable transform and translation vector are given by $H_i = A_i \otimes B_i$ and D_i, respectively.

A tree-structured vector restoration (TSVR) coder is based on a tree of restoration functions as shown in Fig. 2. Encoding is performed through a binary decision at each level of the tree for the restoration node that yields minimum distortion between the original and restored vectors. The binary search continues until a terminal, or leaf, node is reached. The path of '0's and '1's describing the sequence of binary searches is transmitted to the decoder and is used to retrieve the specified restoration function.

The codebook generation for tree-structured vector restoration (TSVR) and vector quantization (TSVQ) differ only slightly in the specifics of the calculation of the codebook entries and the splitting of these entries to generate higher-rate codebooks. In our initial study [2], we provided a detailed treatment of the calculation of the optimum restoration function from a partitioned set of training vectors using the generalized Lloyd algorithm [6].

Higher rate TSVR codebooks are obtained by splitting the terminal nodes' restoration functions of an initial codebook into left and right children (*lchild* and *rchild*) restoration function nodes. We split the parent restoration function by perturbing the translation portion of the restoration function. The linear estimation portion is merely replicated from the parent nodes to its children nodes. This effectively divides the partition and maintains the relationship used to exploit the predicted-residual vector correlation for the parent node. The complete restoration function splitting is given by (2) and (3), where ε is the perturbation vector.

$$H_{lchild} = H_{rchild} = H_{parent} \tag{2}$$

$$D_{lchild} = D_{parent} + \varepsilon \quad \text{and} \quad D_{rchild} = D_{parent} - \varepsilon \tag{3}$$

We set the j^{th} element of the perturbation vector equal to one-half the standard deviation of the difference between the j^{th} elements of the original and restored vectors. The standard deviation quantities are obtained from the diagonal of the original-restored vector conditional covariance matrix $C_{X\hat{X}}$, where

$$C_{X\hat{X}} = C_{XX} - C_{X\tilde{X}}H_i^T - H_i C^T_{X\tilde{X}} + H_i C_{\tilde{X}\tilde{X}} H_i^T \tag{4}$$

Equation (4) allows for the calculation of the perturbation vector without requiring the encoding of the training data for a new restoration function.

Once the tree-structured VR codebook has been generated for the desired depth, optimal codebook pruning using the generalized BFOS algorithm [4] can be applied directly to obtain variable rate TSVR codebooks. The two variants of this technique trade either average rate (length-pruned tree structure) or average entropy (entropy-pruned tree structure) for average distortion by discarding nodes from the tree, resulting in an unbalanced tree structure.

3. EXPRIMENTAL RESULTS

All image frames were predicted using block-based motion compensation with half-pixel accuracy and 16×16 blocks. Predicted and corresponding original image blocks were subdivided into four 8×8 blocks for use in pruned tree-structured vector restoration (PTSVR) and residual vector quantization (PTSVQ) simulations.

The original and predicted training vectors were obtained from the luminance component of 149 frames of each the *Mom* and *Mom and Daughter* video-conferencing sequences using an MPEG-1 simulation. The vectors were included in the training set only if the prediction distortion was greater than a fixed threshold. This resulted in a training set of 91219 vectors.

Tree-structured VR and VQ codebooks with eleven levels were designed from the training set. Figure 3 shows the distortion performance for the training set obtained by length-pruning the balanced binary tree codebooks to various average rates. Similar results were obtained for entropy-pruned TSVR and TSVQ.

For test purposes, 149 frames of the *Grandmom* video-conferencing sequence were coded via PTSVR and PTSVQ. The first frame in the sequence was losslessly encoded and the remainder of the frames were predicted from the previously reconstructed frame. Both codebooks were pruned to approximately 0.14 bpp. A predicted block was coded only

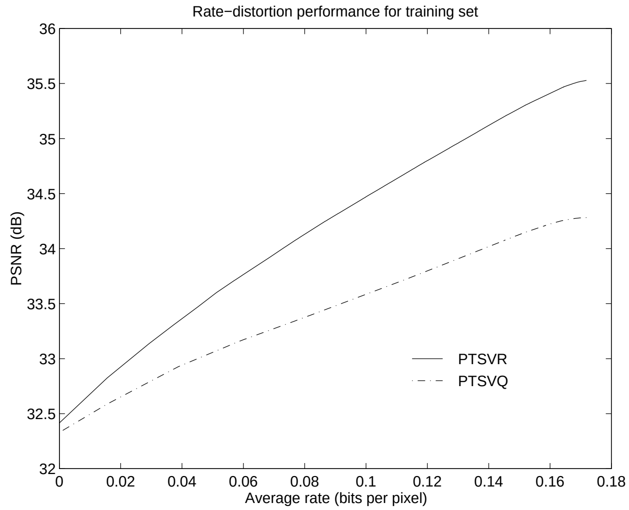

Figure 3: PSNR versus average rate for training set.

if its distortion was greater than a fixed threshold. Since the reconstructed vector distortion may be greater than that of the predicted vector distortion for both VR and VQ, a block was scalar quantized in both simulations when this situation occurred and the prediction distortion was greater than the specified threshold. The per-frame peak signal-to-noise ratio (PSNR) and bit count results are shown in Fig. 4 (average values in parentheses). These results are for coded blocks and do not include bit counts for frame or macroblock header information. The gains experienced for the test sequence are consistent with those of the training set.

4. CONCLUSIONS

We have extended the concepts of tree-structured codebooks and optimal codebook pruning from vector quantization to vector restoration based coding; the only difference is in the splitting of the codebook. However, significant performance gains result. For the *Grandmom* sequence PTSVR yielded an average gain of approximately 1.23 dB over PTSVQ, with only a 3% increase in bit rate.

Future work will include comparisons of optimally pruned TSVR with rate-controlled DCT-based coders. While this study focused on reduced complexity vector restoration through tree-structured codebooks, we also plan to examine the performance of an entropy-constrained vector restoration implementation [7].

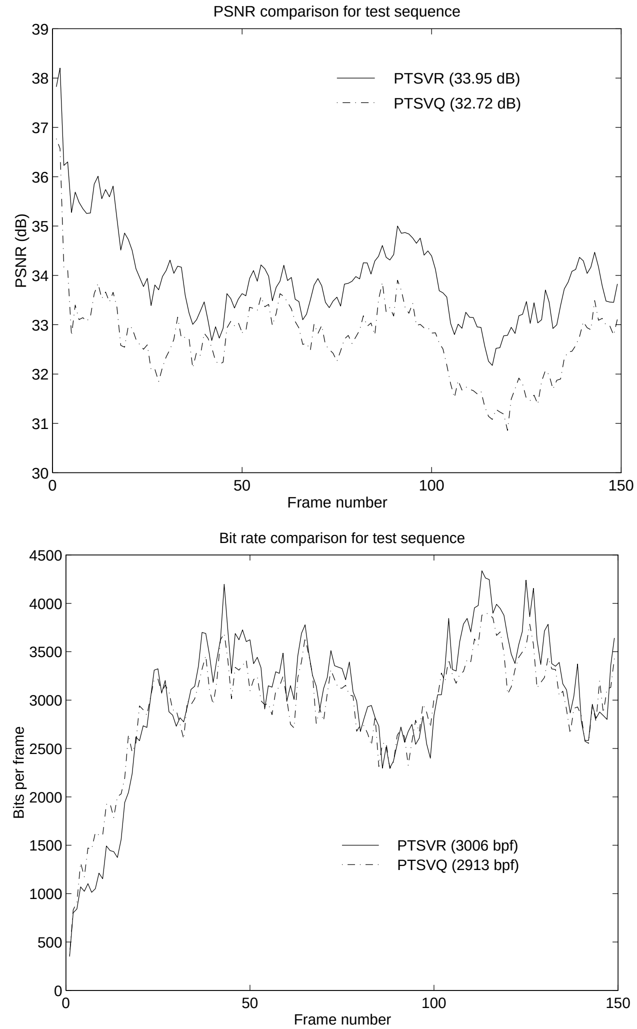

Figure 4: Equivalent rate performance comparison between PTSVR and PTSVQ for *Grandmom* sequence.

5. REFERENCES

[1] ISO/IEC/JTC1/SG29/WG11, ISO/IEC 11172-2, "Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s - Part 2: Video," May 1993.

[2] J. S. McVeigh, S.-W. Wu, M. W. Siegel and A. G. Jordan, "Vector restoration for video coding," *Proc. IEEE Internat. Conf. on Image Processing*, October 1995, pp. TBD.

[3] A. Gersho and R. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1991.

[4] P. A. Chou, T. Lookabaugh and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. on Info. Theory*, vol. 35, no. 2, 1989.

[5] T. Lookabaugh, E. A. Riskin, P. A. Chou and R. M. Gray, "Variable rate vector quantization for speech, image, and video compression," *IEEE Trans. on Communications*, vol. 41, no. 1, 1993, pp. 186-199.

[6] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. 28, no. 1, 1980.

[7] P. A. Chou, T. Lookabaugh and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 37, no. 1, 1989, pp. 31-42.