A NEW TRELLIS VECTOR RESIDUAL QUANTIZER: APPLICATIONS TO IMAGE CODING

Giovanni Motta

Volen Center for Complex Systems
Brandeis University
Waltham, MA 02254, U.S.A.
gim@cs.brandeis.edu

Bruno Carpentieri

Dipartimento di Informatica ed Applicazioni
Università di Salerno
84081 Baronissi (SA), Italy
bc@udsab.dia.unisa.it

ABSTRACT

We present a new Trellis Coded Vector Residual Quantizer (TCVRQ) that combines trellis coding and vector residual quantization. We propose new methods for computing quantization levels and experimentally analyze the performances of our TCVRQ in the case of still image coding. Experimental comparisons show that our quantizer performs better than the standard Tree and Exhaustive Search Quantizers based on the Generalized Lloyd Algorithm (GLA).

1. A NEW TRELLIS VECTOR RESIDUAL QUANTIZER

Quantization is the process of approximating a continuous-amplitude signal by a digital (discrete-amplitude) signal, minimizing a distortion measure (or error). Unfortunately, fixed the coding rate R, an optimal VQ requires computational and storage resources that grow exponentially with the vector dimension. Moreover, Lin in [1] showed that the design of an optimal Vector Quantizer is an NP-complete problem. The design of sub-optimal vector quantizers is an interesting alternative to scalar quantizers for applications that require good quality performances when a limited amount of computing resources is available.

In this paper we present a new VQ architecture that operates recursively on the quantization residuals. We will demonstrate the performances of this new VQ based on a trellis structure and compare it experimentally with the standard Exhaustive Search and Tree Vector Quantizers.

Quantizers based on Trellis Coding were first proposed by Fisher et al. in [2], and made use of the set partitioning ideas of Ungerboeck [3]. Fisher et al. in [2] presented results for Trellis Coded Vector Quantization (TCVQ) in up to four dimensions. Wang and Moayeri in [4] used the LBG algorithm for codebook design and report results for vector dimension up to 6. Laroia and Farvadin in [5] combined scalar-vector quantization (SVQ) with trellis coding. Belzer and Villasenor in [6] presented design techniques for vector quantizers with highly symmetric codebooks that facilitate low complexity quantization as well as partitioning into equiprobable sets for trellis coding.

Our quantizer is different from other structures proposed in literature: all the stages of our TCVRQ are used to encode the whole vector and our trellis works by removing the statistical dependence among vector components and not among distinct vectors.

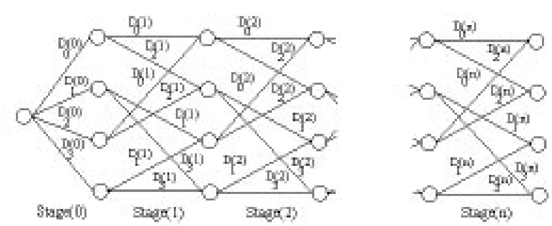

Figure 1: Trellis Coded Vector Residual Quantizer.

Figure 1 shows the VQ proposed; it is based on a multistage trellis that quantizes at each stage the quantization error (or *residual*) of the previous stage; we name it *Trellis Coded Vector Residual Quantizer* or for short TCVRQ. Our trellis quantizer is organized in a sequence of identical *stages*. Each one of these has *states* (labelled $S_0, \ldots, S_3$) and *transitions*. A small *Exhaustive Search Vector Quantizer* (ESVQ) is associated to each transition.

A P-stages TCVRQ, encodes a vector through P consecutive approximations. Initially the input vector $\mathbf{x} = \mathbf{x}_1 = [x_1, x_2, \ldots, x_n]$ is quantized as $\mathbf{y}_1 = Q_1(\mathbf{x}_1)$, the error $\mathbf{x}_2 = \mathbf{x}_1 - \mathbf{y}_1$ is generated and encoded by the following stage as $\mathbf{y}_2 = Q_2(\mathbf{x}_2)$, and then the process is iterated again. The $i-th$ stage of the quantizer works on the error $\mathbf{x}_i = \mathbf{x}_{i-1} - \mathbf{y}_{i-1}$ and the output vector is $\mathbf{y} = \sum_{p=1}^{P} Q_p(\mathbf{x}_p)$. The best sequence of ESVQs, associated to each branch of the graph, is selected minimizing (for example using Viterbi's algorithm) a given distortion measure. If, for each stage N_s is the number of the states, N_t is the number of transitions that leave a state and N_p is the codebook size for each ESVQ, a quantized vector is fully specified giving $P(\log_2(N_t)+\log_2(N_p))$ ($P\log_2(N_t)$ bits for the path and $P\log_2(N_p)$ for the ESVQ codeword).

Decoding is performed by following the path and summing all the codewords, i.e. if $\mathbf{x}^1$ is the original vector, $\mathbf{y}^p = Q^p(\mathbf{x}^p)$ is the p-th quantized residual, then: $\mathbf{y}^1 = Q(\mathbf{x}^1) = \sum_{p=1}^{P} Q^p(\mathbf{x}^p)$. Decoding can be ended before the P-th stage if we desire a variable-rate quantizer.

2. CODING COMPLEXITY

The coding complexity of our TCVRQ needs to be measured in terms of computational complexity (C) and storage requirements (S). Using a trellis with P stages and a codebook of L levels for stage, the quantization of n-dimensional vectors with Mean Square Error (MSE) requires a number of comparisons for each vector given by:

$$C_{TCVRQ} = (L * (1 + 2 * (P - 1))) * n \qquad (1)$$

and the storage requirements are proportional to:

$$S_{TCVRQ} = P * L * n. \qquad (2)$$

An optimal quantizer (ESVQ) with the same coding rate ($P\log(L/2) + 1$ bit/vector) uses for each vector $C_{ESVQ} = n * 2^{P\log(L/2)+1}$ and requires $S_{ESVQ} = n * 2^{P\log(L/2)+1}$ storage locations: the TCVRQ approach is computational and memory effective with respect to optimal ESVQ.

3. CODEBOOK DESIGN

We design the quantization levels for each stage using a modified LBG algorithm. The training sets for each trellis branch are composed only by residuals originated from the entering branches. Random Partitioning is used to partition the levels for the ESVQs on the branches. The stages are sequentially optimized, i.e. a local optimum for the stage i is found by using only information from the $0\ldots(i-1)$ stages. RQ joint optimization techniques described in [7] have not been useful to these experiments due to the excessive computational burden of dealing with 10-stages VRQ. In spite of their low computational complexity, sequential optimization and random partitioning perform well. We have not observed substantial performance increase using more sophisticated partitioning algorithms.

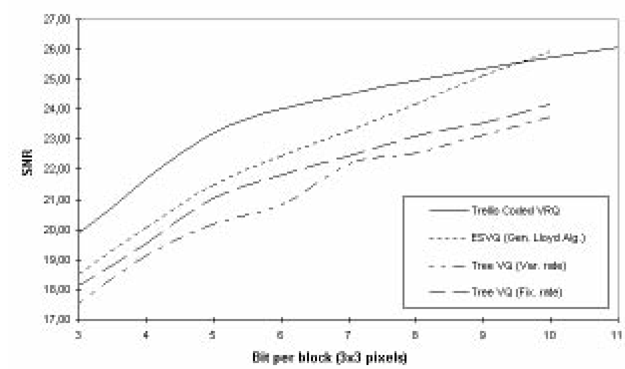

Figure 2: Performance comparison.

4. EXPERIMENTAL RESULTS

Our TCVRQ is a general-purpose VQ, with low computational costs and small memory requirements. This makes it very appealing for low bit-rate coding applications. In order to evaluate TCVRQ performance in a real application and with a natural source, we have experimented our TCVRQ with a training set composed by 12 images, all of dimensions 512 by 512 pixels, with 256 gray levels, including standard test images like "Ape", "Blonde", "Hotel Lotus", etc.. We divide the images into vectors of 3 by 3 pixels and quantize the vectors directly by minimizing their Mean Square Error (MSE). Minimization is performed on the trellis with the Viterbi Algorithm. We have compared the performances of our TCVRQ with three classical Vector Quantizers:

- an Exaustive Search Vector Quantizer designed using Generalized Lloyd Algorithm;
- a Tree Vector Quantizer with an unbalanced tree and a variable rate of coding;
- a Tree Vector Quantizer with a balanced tree and a fixed rate of coding.

We have tested these quantizers on a set of 18 images out of the training set (images like "Peppers", "Woman with a Hat" etc.).

Figure 2 plots the average SNR obtained as a function of the bits used to encode each vector (block). The compression rate ranges between 24:1 and 7:1. The figure shows that our TCVRQ exhibits a slope similar to the two Tree quantizers and, in the interval between 3 and 10 bits/block, outperforms the standard ESVQ.

In terms of memory and computational complexity it is clear that our TCVRQ outperfoms the standard algorithms. The memory (M) and the number of comparisons (C) needed to quantize a n-dimensional vectors with k bits per vector can be summarized as follows for each of the three different approaches:

- **TCVRQ:** $M = O(P*L*n)$, $C = O(2*P*L*n)$. (L levels, P stages, k = P log(L/2) bit/vector)
- **ESVQ:** $M = O(2^k * n)$, $C = O(2^k * n)$.
- **TreeVQ:** $M = O(2^{(k+1)} * n$, $C = O(k * n)$.

5. REFERENCES

[1] J. Lin, "Vector Quantization for Image Compression: Algorithms and Performance", Ph.D. Dissertation, Brandeis University, Computer Science Dept., 1992.

[2] T. R. Fischer, M. W. Marcellin, and M. Wang, "Trellis Coded Vector Quantization", IEEE Trans. Inform. Theory, vol.37, No. 3, pp.1551-1566 , 1991.

[3] G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals", IEEE Trans. on Information Theory, Vol IT-28, N.1, January 1982.

[4] H. S. Wang and N. Moayeri, "Trellis Coded Vector Quantization", IEEE Trans. on Comm., Vol 28, N. 8, 1992.

[5] R. Laroia and N. Farvadin, "Trellis-based scalar-vector quantizers for memoryless sources", IEEE Trans. Inform. Theory, vol.40, No.3, 1994.

[6] B. Belzer and J. D. Villasenor, "Symmetric Trellis Coded Vector Quantization", Proceedings of IEEE Data Compression Conference (DCC96), 1996.

[7] C. F. Barnes, "Residual Vector Quantizers", PhD Thesis, Brigham Young University, 1989.