# Entropy-Constrained Tree-Structured Vector Quantizer Design

Kenneth Rose, David Miller, and Allen Gersho

*Abstract*— **Current methods for the design of pruned or unbalanced tree-structured vector quantizers such as the Generalized Breiman–Friedman–Olshen–Stone (GBFOS) algorithm are effective, but suffer from several shortcomings. We identify and clarify issues of suboptimality including greedy growing, the suboptimal encoding rule, and the need for time sharing between quantizers to achieve arbitrary rates. We then present the leaf-optimal tree design (LOTD) method which, with a modest increase in design complexity, alters and reoptimizes tree structures obtained from conventional procedures. There are two main advantages over existing methods. First, the optimal entropy-constrained nearest-neighbor rule is used for encoding at the leaves; second, explicit quantizer solutions are obtained at all rates without recourse to time sharing. We show that performance improvement is theoretically guaranteed. Simulation results for image coding demonstrate that close to 1 dB reduction of distortion for a given rate can be achieved by this technique relative to the GBFOS method.**

## I. Introduction

Vector quantization (VQ) [1] has become a valuable and essential technique in speech coding, where it is part of most standards and is in widespread commercial use. However, the same is not true for image and video compression. While the increased complexity of VQ for image coding is often viewed as the key reason, another relevant issue is that most speech coding algorithms require a fixed rate and hence cannot easily benefit from variable rate coding methods. Most of the early research on VQ methods was geared to fixed-rate designs and, therefore, did not offer strong competition against the use of scalar quantization in combination with entropy coding schemes, as in the well-known JPEG and MPEG standards. Recently, however, variable-rate tree-structured VQ schemes with manageable complexity have emerged, offering great potential for practical image coding. In this paper we offer a simple method to significantly enhance the performance achievable with these quantizers.

### A. Entropy-Constrained Vector Quantization

One of the more fundamental extensions of the basic VQ concept was the proposal to incorporate an entropy constraint within the design so as to produce quantizers optimized for subsequent entropy coding. The pioneering work in this area was performed by Berger [2], Farvardin and Modestino [3], and Chou, Lookabaugh, and Gray [4]. Incorporation of an entropy constraint has since been considered in a variety of quantization schemes, notably the work of Pearlman and colleagues [5], [6] and others [7], [8]. In [4], a design algorithm was presented for generating locally optimal entropy-constrained vector quantizers to minimize distortion subject to a Lagrangian constraint on rate. This algorithm, called entropy-constrained vector quantization (ECVQ), is a simple and elegant extension of the classical generalized Lloyd algorithm (GLA) for VQ design [9], itself a generalization of Lloyd's scalar quantizer design method [10]. The design problem is posed as the minimization of the Lagrangian $L = D + \lambda R$, where $D$ is the expected distortion, $R$ is the rate (reproduction entropy), and $\lambda$ is the Lagrange multiplier whose value determines a particular rate-distortion tradeoff. It was shown that the resulting quantizers outperform those obtained from a number of other variable-rate quantization schemes. However, a practical limitation of this approach, and of exhaustive-search fixed-length VQ, is encoding search complexity, which grows exponentially with both vector dimension and rate. The problem is particularly acute in the case of ECVQ where optimal (or close to optimal) codebooks can be many times larger than the corresponding fixed-rate codebook. Thus, in practice, vector quantizers of reduced encoding search complexity may need to be considered.

Manuscript received July 30, 1994; revised July 30, 1995. This work was supported by the National Science Foundation under Grant No. NCR-9314335, and by the University of California Micro Program, DSP Group, Inc., Echo Speech Corporation, Moseley Associates, Qualcomm, Inc., Rockwell International Corporation, Speech Technology Labs, and Texas Instruments, Inc. Portions of this paper were presented at the IEEE Data Compression Conference, Snowbird, UT, USA, March 1994.

The authors are with the Center for Information Processing Research, Department of Electrical and Computer Engineering, University of California Santa Barbara, CA 93106, USA (e-mail: rose@ece.ucsb.edu).

Publisher Item Identifier S-1057-7149(96)01409-1.

### B. Tree-Structured Vector Quantization

The classical technique of tree-structured VQ (TSVQ) introduced by Buzo *et al.* [11] offers a major reduction in VQ complexity (see also [1] for a tutorial presentation of TSVQ.) Initially proposed for fixed-rate applications, unbalanced tree structures were later proposed that allow convenient variable-rate coding [12]–[14]. The structural constraint of TSVQ implies some inherent loss in performance, since the set of possible VQ solutions has been restricted to contain only those that satisfy this additional constraint. Another source of performance degradation is the design approach for TSVQ. The conventional, standard design methods for TSVQ are "greedy" methods, growing trees one node at a time to minimize a local, heuristic cost function [11], [12], [14]. In fact, even convergence to a *local* optimum is not ensured, as can be demonstrated with very simple examples such as the design of a three-layer binary tree. Moreover, for TSVQ there are no iterative methods akin to GLA for monotonically reducing distortion.

### C. Tree Pruning

An important advance was the introduction of *tree pruning* for TSVQ design by Chou, Lookabaugh, and Gray in 1989 [13]. This approach generalizes the known Breiman-Friedman-Olshen-Stone (BFOS) method for optimally pruning classification and decision trees [15] and is therefore called the generalized BFOS (GBFOS) algorithm. GBFOS is based on a simple observation. Starting with a large initial TSVQ having a high rate (entropy) and low distortion, one can trade an increase in distortion for a reduction in rate by pruning the tree. Consider the set of distortion-rate operating points of all the pruned trees obtainable from the initial tree. A pruning sequence produces a distortion-rate curve, where each pruning operation connects two operating points (before and after pruning) by a segment whose slope is $s = \Delta D/\Delta R$. Optimal pruning corresponds to finding the lower convex hull of the set of distortion-rate operating points. It is therefore achieved by always selecting the least steep slope, i.e., by successively pruning branches of maximum (least negative) $s$. While GBFOS can be efficiently used to design TSVQ's that satisfy an entropy constraint, there are inherent difficulties with such an approach that result in suboptimality of the solution. In this correspondence, we describe and address these problems.

An alternative to growing and pruning a tree is, of course, to grow an unbalanced tree directly. Methods for growing unbalanced trees have been suggested. Makhoul *et al.* [12] developed a method for

the design of unbalanced trees in the context of fixed-rate coding. More directly relevant to our discussion is the work of Riskin and Gray [14], who proposed a "greedy growing" method for the design of variable-rate TSVQ's. Their approach uses a heuristic decision rule to determine the order of node splits that does not ensure the optimality of the resulting trees. Consequently, the performance of TSVQ's produced by greedy growing is inferior to or, at best, the same as GBFOS. For this reason, we shall use the results of GBFOS in comparisons.

In Section II we enumerate and discuss problems associated with greedy growing and pruning methods that cause degradation in performance. In Section III we present the LOTD method. It eliminates some GBFOS shortcomings by readjusting the partition and the reproductions at the leaves to enforce the optimal entropy-constrained encoding rule. The method is shown to be theoretically superior to GBFOS in the quality of the solutions as well as in its ability to provide tree solutions at all rates. Experimental results are presented in Section IV, which show significant improvement on various sources. These gains are obtained with only a modest increase in design complexity.

## II. Shortcomings of Growing and Pruning Methods

The conventional approach for designing variable-rate TSVQ's is a two-phase procedure that involves growing an initial tree and then pruning the tree back to achieve the best possible distortion rate tradeoff. Although the growing phase of the design is greedy and suboptimal, tree pruning via the GBFOS algorithm allows one to determine efficiently a sequence of tree solutions that is optimal in a well-defined sense. More specifically, if we consider the set of distortion rate operating points corresponding to the pruned trees derivable from the given initial tree, then the solutions found by GBFOS are the extreme points on the set's convex hull. The GBFOS approach, therefore, has both practical and theoretical significance for TSVQ design. However, the GBFOS framework also imposes substantial constraints on the solution space over which the search for the optimal TSVQ is conducted. We discuss here three important shortcomings that result in suboptimal performance.

### A. Greedy Growing

First, we reemphasize that the solutions obtained by a tree-pruning method such as GBFOS depend on the initial tree, which is typically generated in a greedy fashion, either by the splitting algorithm [11] or possibly by the Riskin–Gray method [14]. Indeed, pruning consists of removing existing boundaries but it cannot adjust "poor" boundaries. For an extreme example, if the first split imposes suboptimal constraints on the partition at the leaves, pruning cannot undo this without removing the entire tree. For the fixed-rate case (as well as for data clustering in the context of statistical pattern recognition), it has been shown that splitting and greedy growing of unbalanced trees can be significantly suboptimal and that substantial performance gains can be achieved over these methods [16]–[18]. Thus, the performance achieved by any pruning method is limited by the quality of the initial tree.

### B. Sparse Set of Pruned Trees

A second problem relates to the fact that GBFOS only finds a finite set of TSVQ solutions and, hence, a convex hull uniquely specified by these extreme points. If we want to obtain a rate between two extreme points, then it is possible to "time share" between the two pruned-tree solutions so as to achieve a point on the convex hull with the given rate. However, concerns have been raised, particularly in image coding, regarding the undesired perceptual effects of time sharing due to nonstationary quantizer performance [19]. Moreover, since the theoretically optimal distortion-rate curve is almost always strictly convex, it is likely that there is a better tree-structured solution (albeit, not obtainable by pruning of the initial tree) below the convex hull point at the prescribed rate. In the sequel, we shall see that instead of time sharing between the solutions before and after a pruning operation, we can obtain a continuum of intermediate TSVQ solutions trading entropy for distortion through a gradual modification of reproduction values and of the encoding partition at the leaves. This "gradual" pruning can become a "complete" pruning operation at the limit when one leaf captures all the probability from its siblings or at the limit when the code vectors of the siblings merge to a single point. The corresponding convex distortion-rate curve may lie below the straight connecting line assumed by GBFOS.

### C. Encoding Rule Is Suboptimal

The most fundamental shortcoming is the encoding rule employed by GBFOS solutions. Recall the encoding rule for unconstrained VQ design [1]. It is the nearest neighbor rule that includes source vector $x$ in partition cell $S_j$ if $y_j$ is the nearest codevector, as follows:

$$x \in S_j \text{ only if } d(x, y_j) \leq d(x, y_k), \forall k. \tag{1}$$

In ECVQ [4], the rate constraint causes a modification of the nearest neighbor encoding rule to include an entropy cost based on cell probability $p_j$. The optimal entropy-constrained rule is

$$x \in S_j \text{ only if } d(x, y_j) - \lambda \log p_j \leq d(x, y_k) - \lambda \log p_k, \forall k. \tag{2}$$

In this expression $\lambda$ is the Lagrange multiplier, which is determined by the prespecified rate. Obviously, the modified encoding rule (2) *varies with rate*. Thus, the optimal partition boundaries and the probabilities associated with the regions also vary with rate. The modified encoding rule (2) must be satisfied by an optimal entropy-constrained solution. This reveals a fundamental shortcoming of pruning a fixed tree (with a fixed rate-independent partition): The encoding rule for the resulting pruned trees is not optimal. More specifically, pruning trades rate for distortion by merging partition regions; however, apart from removing some region boundaries, the remaining boundaries in the pruned tree are left unaltered as determined by the initial tree-growing algorithm. However, the optimal boundaries depend on the particular rate at which the resulting structure is to operate. The GBFOS method is constrained to use an initial tree grown while possibly incorporating the encoding rule for some fixed preassigned rate (or $\lambda$). In practice and in all published work, to the best of our knowledge, the encoding rule used in the growing phase simply ignores the entropy constraint (i.e., , $\lambda = 0$).

## III. Leaf-Optimal Tree Design

In this section, we present *leaf-optimal tree design* (LOTD), which eliminates two of the three major shortcomings of standard growing/pruning methods. The LOTD technique produces entropy-constrained TSVQ's for arbitrary rates while satisfying the necessary optimality conditions for encoding at the leaves. It thereby outperforms structures obtained from the GBFOS algorithm while providing quantizer solutions at all rates.

### A. Derivation

The previous section suggests that the distortion-rate performance of any tree structure obtained from a solution to the GBFOS algorithm can be improved by modifying the leaves. Here we show that this can be achieved by applying the ECVQ design method for unstructured VQ to redesigning the leaves. We associate a pair of variables with each leaf node of the tree, namely cell probability and reproduction

code vector. While keeping the nonleaf nodes of the tree fixed, adjusting the cell probabilities and code vectors for all the leaves leads to different tradeoffs between entropy and distortion. By optimizing over these variables, we can make leaves gradually "vanish" by decreasing their probability to zero or "merging" their code vectors.

Consider a TSVQ, denoted $Q_o$, that is not necessarily balanced. If $Q_o$ is obtained from the GBFOS algorithm by entropy-pruning, then the performance of this quantizer corresponds to an extreme point on the convex hull of the operational distortion-rate curve for some rate $R(Q_o)$. We now explicitly incorporate the entropy constraint in the following leaf optimization procedure to improve the GBFOS solution. Regarding $Q_o$ as an initialization, we fix the tree-structured partition from the root down to the parent layer of the leaves. We then pose the optimization problem

$$\min L = \min\{D + \lambda R\} \tag{3}$$

over the leaf probabilities, code vectors, and partition, subject to the structural constraint imposed on permissible partitions by the fixed nonleaf structure of $Q_o$. To express this constraint explicitly we introduce additional notation.

Each node $j$ in the parent layer of the leaves has a fixed partition region $\{S_j\}$. Moreover, the descendent leaves of node $j$ partition the set $S_j$ into mutually exclusive sets $\{S_{jk}\}$. Thus, the contribution of $S_j$ to the total distortion is

$$D_j = \sum_k \sum_{x \in S_{jk}} d(x, y_{jk}) \tag{4}$$

and its contribution to the rate is

$$R_j = -\sum_k \sum_{x \in S_{jk}} \log p_{jk}. \tag{5}$$

The total distortion and total rate for the quantizer are $D = \sum_j D_j$ and $R = \sum_j R_j$, respectively. Hence, minimizing $L$ is equivalent to separately solving

$$\min L_j = \min\{D_j + \lambda R_j\} \tag{6}$$

over all possible partitions $\Pi_j = \{S_{jk}\}$, over all leaf codebooks $Y_j = \{y_{jk}\}$, and over all probability distributions $\{p_{jk}\}$ for the leaves, subject to the constraint $\sum_k p_{jk} = p_j$ where $p_j$ is the probability of parent node $j$. In other words, we must minimize $L_j$, the cost of $S_j$ over its partition, codebook, and probabilities. Note that in solving this problem we impose the structural constraint on the parent layer partition $S_j$. It is easy to see that the required optimization for the leaves of the $j$th parent node is almost exactly the same as the task of designing an optimal unstructured ECVQ. The only minor difference is that the sibling probabilities should sum up to the probability of their parent and not to one as would be the case for "standard" ECVQ. In fact, it is straightforward to show that we can break the problem into separate standard ECVQ design problems but with the resulting sibling probabilities renormalized to satisfy

$$\sum_k p_{jk} = p_j. \tag{7}$$

We can summarize our optimization problem as follows: Minimize the Lagrangian in (3) subject to the structural constraint given by the partition $\{S_j\}$. Clearly, we have applied ECVQ formalism to the data set subject to the partition imposed by the nonleaf nodes of the tree. By varying $\lambda$ over the positive real line and solving this "tree-constrained" ECVQ problem, we generate a sequence of solutions $\{Q(\lambda)\}$ that use the optimal entropy-constrained nearest neighbor rule (2) at the leaves. Note that the GBFOS solutions implicitly set $\lambda = 0$, since the tree is grown without explicitly incorporating entropy constraint into the process. Thus, our solution for $\lambda = 0$ is simply the initial GBFOS quantizer $Q_o = Q(0)$. Moreover, for any rate less than $R(Q_o)$ (i.e.,, for $\lambda$ strictly greater than zero), the encoding rule associated with the GBFOS solutions *cannot be optimal* except for the special case in which all code vectors are equiprobable and, thus, the ECVQ encoding rule is equivalent to the standard nearest neighbor rule. Note further that this statement of suboptimal encoding is not necessarily limited to the case of entropy cost, since other costs, when explicitly enforced, may also lead to solutions with a modified optimal encoding rule.

The above method for generating entropy-constrained solutions starting with a given TSVQ solution can be embedded within an iterative algorithm whose objective is to design a quantizer for a specified rate. The pseudocode for such an algorithm is given in Table I.

We emphasize again that the resulting TSVQ satisfies the entropy-constrained optimality conditions *for the leaves.* The solution is still constrained by the partition according to the parents. Once a tree has been designed with LOTD for a particular rate and corresponding value of $\lambda$, the encoding rule that is applied for actual use of the quantizer is the standard nearest neighbor encoding rule at all layers excepting the leaf layer, at which the modified nearest neighbor rule is applied for the given $\lambda$.

### B. Guaranteed Improvement Near the Initial Quantizer

We have seen that, starting with a GBFOS solution $Q(0)$, we can generate a set of solutions $\{Q(\lambda)\}$, i.e., new distortion-rate points. One cannot in general state that these new points will all lie below the operational distortion-rate curve of GBFOS. First, the cost function is nonconvex and so there may be local optima to trap the optimization. Second, by fixing the tree except for the leaves, we effectively limit the distortion-rate performance achievable starting from a particular initial tree $Q(0)$. It is possible that better performance for a particular rate will be achieved by "running" our method with a different convex hull extreme point as initialization. However, we can show that the resulting distortion-rate curve will always lie below that produced by GBFOS for sufficiently small $\lambda$, i.e., at least in the vicinity of $Q(0)$. Let us reconsider the basic optimization problem solved by LOTD, namely, minimization of the Lagrangian (3). A standard result in constrained optimization (see, for example, [20]) relates the Lagrange multiplier to the slope of the solution curve. For the LOTD problem it states that the slope of the distortion-rate curve at a solution point is determined by the value of the corresponding Lagrange multiplier via

$$\lambda = -\frac{dD}{dR}. \tag{8}$$

Our process starts at a GBFOS solution for $\lambda = 0$ and obtains points by increasing $\lambda$. It therefore generates a curve whose slope is zero when it meets the GBFOS curve. Our curve must, therefore, lie below GBFOS at least for some interval just below the initial rate, as is illustrated in Fig. 1. We conclude that GBFOS is suboptimal and that this simple improvement method will find solution points below GBFOS's convex hull for an interval in $\lambda$ for *each* extreme point on the convex hull.

## IV. Simulation Results

Some simulation results are depicted in Figs. 2–4. We compare the performance of our method with GBFOS [13], which is ensured to bound the performance of greedy growing [14]. The GBFOS curves were obtained by pruning an initial balanced tree in order to trade entropy for distortion performance. The initial trees were obtained in the conventional way using the splitting algorithm [11]. Each point on the LOTD curve is obtainable by applying the algorithm described

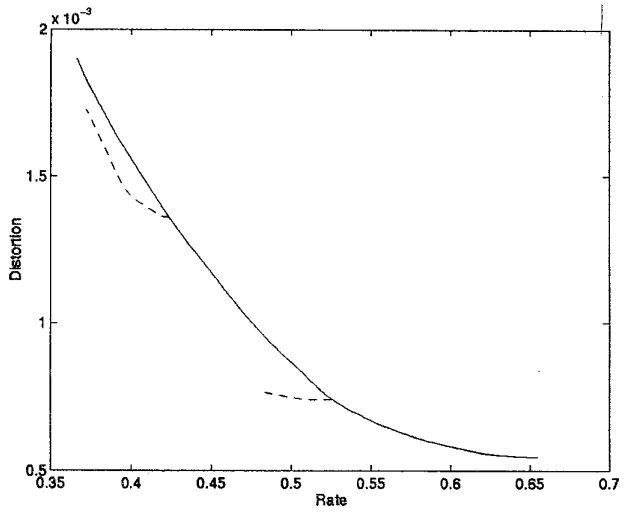


Fig. 1. Example showing our distortion-rate curves (dashed) departing from three particular GBFOS solutions as we increase $\lambda$ and optimize. Note the zero slope of our curve where it leaves the GBFOS curve (solid line). This guarantees superior distortion-rate performance at least for small $\lambda$. This curve was generated using the image data of Lena and Barbara as the training set.

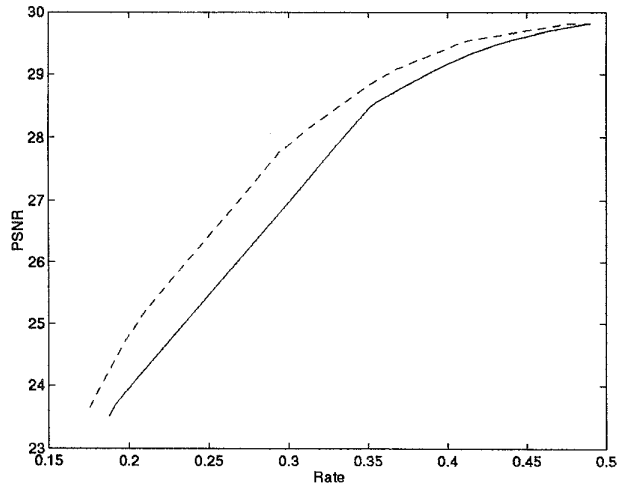


Fig. 2. Peak signal-to-noise ratio (PSNR) versus rate performance for GBFOS (solid curve) and LOTD (dashed curve) within the training set, which was randomly extracted from four images. Vector size is 16, the tree branching factor is eight and the initial tree depth is three.

in Table I at a prescribed rate. However, since for depicting the performance we are interested in finding the entire distortion-rate curve rather than a single point on it (which is the objective in Table I), we have eliminated repeated computation for consecutive points in a straightforward manner. The results indicate that our method offers significant gains over other methods for both Gaussian and image sources. It should be stressed again that unlike GBFOS and greedy growing, our approach provides real quantizers for each point on its curve without use of time sharing. Note also that Kiang *et al* suggested the ROPA method [19], whose objective is to eliminate time sharing between "distant" pruned TSVQ's by allowing the use of pruned trees that are not extreme points on the convex hull. However, ROPA simply increases the number of available distinct solutions. It does not provide quantizers at arbitrary rates. Moreover, ROPA compromises the GBFOS distortion-rate performance. LOTD, on the other hand, provides quantizers *at any rate* while obtaining *better distortion-rate performance than GBFOS.*

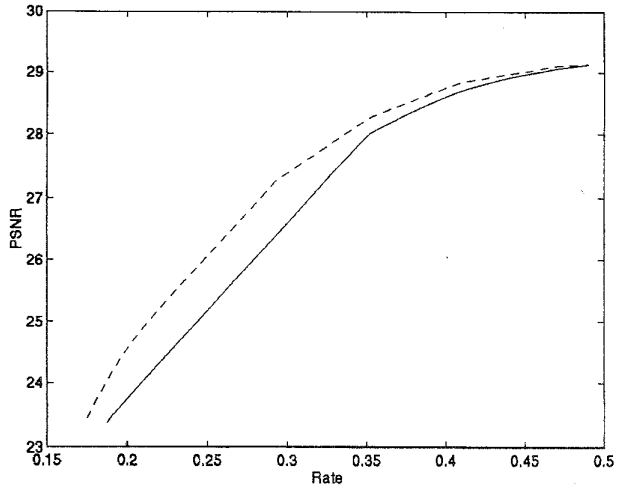


Fig. 3. PSNR versus rate performance for GBFOS (solid curve) and LOTD (dashed curve) outside the training set.

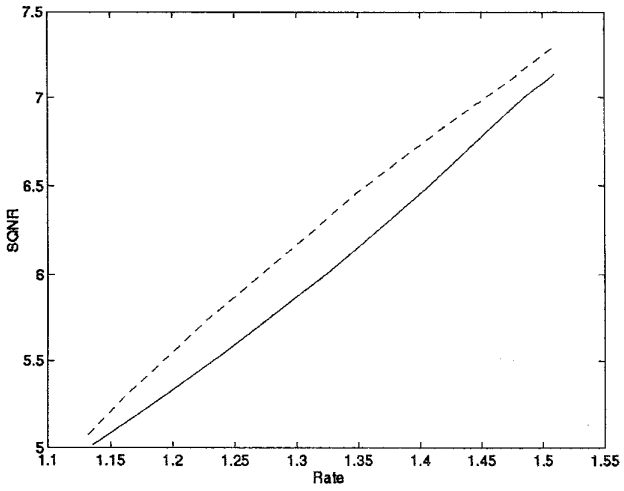


Fig. 4. Signal-to-quantization noise ratio (SQNR) versus rate for GBFOS (solid curve) and LOTD (dashed curve) for a 4-D, isotropic Gaussian source. The tree branching factor is four and the initial tree depth is five.

In Figs. 2 and 3 we show the gains for compression of image data consisting of four $512 \times 512$ pictures: Karen, Lynda, Sailboat, and Peppers. We segmented the images into $4 \times 4$ blocks and randomly extracted 33 000 blocks as training data, while using the remaining blocks as test data. We then applied GBFOS and LOTD to design TSVQ's with a branching factor of eight and a depth of three, i.e., 512 initial leaves. The quantizers' performance inside and outside the training set is shown in Figs. 2 and 3, respectively. Within the training set, we observe consistent significant gains of close to 1 dB for a sizeable range of rates. Outside the training set we see comparable degradation of both GBFOS and LOTD, and the relative gain of LOTD is only slightly diminished. Note that we tested the performance outside the training set but on a set of reasonably similar statistics. Of course, if the test set contains substantially different statistics from the training set, then the gains can be greatly compromised and little can be said about such randomized results since both methods simply attempt to optimize the quantizer for the training set.

TABLE I
A PSEUDOCODE SKETCH OF LOTD FOR GENERATING A TSVQ AT A GIVEN TARGET RATE

*Run* GBFOS to obtain a sequence of pruned TSVQs of *decreasing* rates: $(R_i, Q_i)$.

*Given* the target rate $\hat{R}$

*Set* the quantizer index to $i = \max\{j : R_j > \hat{R}\}$.

*Initialize* the convergence threshold $\epsilon$, and the increment $\Delta\lambda$

*Initialize* $D_{min} \leftarrow$ large value.

do {

*Set* $\lambda = \Delta\lambda$.

*Initialize* $\{S_{jk}\}$, $\{y_{jk}\}$ and $\{p_{jk}\}$ from $Q_i$.

do {

*Initialize* $L \leftarrow$ large value.

do {

Update leaf partition: $x \in S_{jk}$ if $x \in S_j$ and

$d(x, y_{jk}) - \lambda \log p_{jk} \leq d(x, y_{jl}) - \lambda \log p_{jl}$ for all $l$.

Update code vectors: $y_{jk} = \arg\min_y \sum_{x \in S_{jk}} d(x, y)$.

Update probabilities: $p_{jk} = \frac{1}{M} \sum_{x \in S_{jk}} 1$.

Compute $D$, $R$, and $L = D + \lambda R$.

} while not converged: $\Delta L / L > \epsilon$

$\lambda \leftarrow \lambda + \Delta\lambda$

} while $R > \hat{R}$

if $(D > D_{\min})$ or $(i = 0)$: output the tree $Q$, STOP.

$D_{\min} \leftarrow D$, $i \leftarrow i - 1$, store the current tree in $Q$.

}

Note that the phenomenon of diminishing gains as the rate approaches the maximum rate of the curve is a misleading boundary effect. Since our method uses GBFOS solutions as a starting point, its performance must approach GBFOS performance as we approach the full tree used to initialize the pruning phase. This is so because here the "right" TSVQ to initialize LOTD is of higher rate than what is available.

Fig. 4 demonstrates that gains are also obtained for standard data sources such as the Gaussian. For this example, we generated 60 000 4-D isotropic Gaussian vectors and used an initial tree with branching factor four and depth five. The gains here are roughly 0.3 dB for the rate interval shown.

## V. CONCLUSIONS AND FURTHER DIRECTIONS

In this work we presented leaf-optimal tree design (LOTD), a practical method for improving the performance of pruned TSVQ's that are obtained from known methods of greedy growing or growing and pruning. The LOTD method not only provides improvement in distortion-rate performance, but also provides the ability to design real pruned-tree quantizers at arbitrary rates. This is in contrast to standard growing and pruning methods, for which time sharing is unavoidable. Experimental results show significant and consistent improvements over these methods on several sources.

While LOTD is a practical way to significantly improve upon growing and pruning methods such as GBFOS, the improvements attainable by this method are still limited by tree initialization, suboptimal encoding performed at nonleaf nodes, and local optimum traps. In order to address these remaining problems, we must derive an approach for jointly optimizing the entire unbalanced tree. Some preliminary work developing such a joint optimization method is described in [21]. It is inspired by the deterministic annealing algorithm [22], [23], which enables it to avoid many shallow local optima. It further employs the information-theoretic principle of minimum cross entropy (see Shore and Johnson [24]) to embed the interdependence between leaves and other layers of the tree within a useful probabilistic framework. It makes direct use of our recent results on TSVQ design [16]–[18]. Some experimentation with this approach has already shown that consistent significant gains are obtained with respect to the performance of LOTD and, of course, even greater gains with respect to standard methods. However, the complexity of this joint optimization method is substantially higher than that of LOTD and previous standard methods.

## REFERENCES


[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer, 1991.

[2] T. Berger, "Minimum entropy quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 149–157, 1982.

[3] N. Farvardin and J. Modestino, "Optimum quantizer performance for a class of nongaussian memoryless sources," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 485–497, 1984.

[4] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 31–42, 1989.

[5] D. P. de Garrido, W. A. Pearlman, and W. A. Finamore, "A clustering algorithm for entropy-constrained vector quantizer design with applications in coding image pyramids," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 83–95, April 1995.

[6] R. P. Rao and W. A. Pearlman, "Alphabet-constrained vector quantization," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1167–1179, 1993.

[7] F. Kossentini, M J. T. Smith, and C. F. Barnes, "Entropy constrained residual vector quantization," in *Proc. IEEE ICASSP*, April 1993, pp. 598–601.

[8] M. Lightstone and S. K. Mitra, "Entropy-constrained, mean-gain-shape vector quantization for image compression," in *Proc. SPIE Symp. Visual Commun. Image Processing*, Chicago, IL, Sept. 1994, vol. 2308, pp. 389–400.

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

[10] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, Mar. 1982 (reprint of the 1957 paper).

[11] A. Buzo, A. H. Gray Jr., R. M. Gray, and J. D. Markel, "Speech coding based on vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 562–574, 1980.

[12] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, pp. 1551–1588, 1995.

[13] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. Inform. Theory*, vol. 35, pp. 299–315, 1989.

[14] E. A. Riskin and R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers," *IEEE Trans. Signal Processing*, vol. 39, pp. 2500–2507, 1991.

[15] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Belmont,CA.: Wadsworth, 1980.

[16] D. Miller and K. Rose, "An improved tree-structured vector quantizer," in *Proc. IEEE Int. Symp. Inform. Theory ISIT*, San Antonio, TX, Jan. 1993.

[17] ——, "A nongreedy approach to tree-structured clustering," *Patt. Recog. Lett.*, vol. 15, pp. 683–690, July 1994.

[18] ——, "Hierarchical, unsupervised learning with growing via phase transitions," *Neural Computation*, vol. 8, no. 2, pp. 427–452, Feb. 1996.

[19] S.-Z. Kiang, R. L. Baker, G. J. Sullivan, and C.-Y. Chiu, "Recursive optimal pruning with applications to tree structured vector quantizers," *IEEE Trans. Image Proc.*, vol. 1, pp. 162–169, Apr. 1992.
[20] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering Optimization.* New York: Wiley, 1983.
[21] K. Rose, D. Miller, and A. Gersho, "Entropy-constrained tree-structured vector quantizer design by the minimum cross entropy principle," in *Proc. Data Compress. Conf.*, Snowbird, UT, Mar. 1994.
[22] K. Rose, E. Gurewitz, and G. C. Fox, "Statistical mechanics and phase transitions in clustering," *Physical Review Letters*, vol. 65, pp. 945–948, 1990.
[23] ——, "Vector quantization by deterministic annealing," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1249–1257, July 1992.
[24] J. E. Shore and R. W. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 26–37, 1980.

# Fast Tree-Structured Nearest Neighbor Encoding for Vector Quantization

Ioannis Katsavounidis, C.-C. Jay Kuo, and Zhen Zhang

***Abstract*—This work examines the nearest neighbor encoding problem with an unstructured codebook of arbitrary size and vector dimension. We propose a new tree-structured nearest neighbor encoding method that significantly reduces the complexity of the full-search method without any performance degradation in terms of distortion. Our method consists of efficient algorithms for constructing a binary tree for the codebook and nearest neighbor encoding by using this tree. Numerical experiments are given to demonstrate the performance of the proposed method.**

## I. Introduction

Nearest neighbor (NN) encoding [1] is finding the nearest point for an unknown input point from a set of fixed point—called the codebook in vector quantization (VQ)—in a $k$-dimensional vector space. Its fundamental role in the VQ area is indicated by the fact that NN encoding is a synonym for vector quantizing. Moreover, it constitutes the major computational task of the generalized Lloyd algorithm (GLA), which is the most commonly used algorithm for VQ codebook design. It is also used extensively in pattern classification and decision-making problems. A straightforward solution to this problem is the full-search method, which involves an exhaustive search of the distances for all available points; in this way, the complexity of encoding each component of a vector point with full search is proportional exponentially to the dimension $k$ and the bit rate $r$.

Because of its importance, many researchers have looked into this problem in an effort to find ways to accelerate the vector quantization process. We can classify previous work into two groups. The first group consists of methods that do not solve the nearest neighbor problem itself but instead seek a suboptimal solution that is almost as good in the sense of mean squared error (MSE). One such method is to use a tree-structured codebook search. In tree-structured VQ (TSVQ) [1], the search is performed in stages. In each stage, a substantial subset of candidate vectors is eliminated from consideration by a relatively small number of operations. In a binary tree search, the input code vector is compared with two predesigned test vectors at each stage or node of the tree. The nearest test vector determines which of two paths through the tree to select in order to reach the next stage of testing. At each stage, the number of candidate code vectors is reduced to roughly half the previous set of candidates. Efficient TSVQ design often requires simultaneous design of the codebook and tree structure. Various methods for TSVQ codebook design have been proposed, such as splitting [2] and single-node-splitting [3]. Another recent approach is fine-coarse VQ [4] that operates on arbitrary unstructured codebooks, claiming only a slight increase in distortion over full search but with a substantial improvement in speed.

The second group addresses an exact solution of the nearest neighbor encoding problem with less computation than that of exhaustive search. A very simple yet effective method is the *partial distortion calculation* reported by Bei and Gray in [5]. It is important to note that this method requires no memory overhead, but only provides moderate acceleration—about four times over full exhaustive search. Other methods include the *projection method* [6] and its variants [7], [8]. Binary hyperplane testing [9] and its more general form, i.e., $K-d$ trees [10], [11] have also been widely used for fast search. The design of optimal search trees suffers the "curse of dimensionality," which in this problem is expressed in the form of extremely high computational complexity. Thus, their applicability has been limited to small vector dimension and high-resolution cases. A recent work by Ramasubramanian and Paliwal [12] on the optimization of $K-d$ trees provides a family of search algorithms that are quite promising. Unfortunately, they do not guarantee the nearest neighbor classification of an arbitrary vector. Other work on this problem is centered around the use of the triangle inequality property of metric spaces. For details about fast NN-encoding algorithms built on this idea, see [13]–[15].

In this research, we focus on a fast tree-structured nearest neighbor encoder that significantly reduces the complexity of the full-search method without any performance degradation in terms of distortion. Both the construction of a binary tree-structured codebook and nearest neighbor encoding with such a tree-structured codebook are examined. The execution time of these two tasks is so small that it allows the application of the proposed method to slowly adaptive VQ schemes, where adaptation takes place after a number of vectors have been coded. One such case is the GLA, where a very large number of training vectors must be quantized for every iteration, while the update of the codebook only takes place once after every iteration.

This work is organized as follows. After introducing some basic geometrical properties in Section II, we present a new method where a binary search tree with respect to an arbitrary codebook can be obtained, and the nearest neighbor of an input vector can be located effectively in Section III. The performance of the proposed method is reported in Section IV.

## II. Basic Geometrical Properties

Let $C$ be a subset of the $k$-dimensional Euclidean space $R^k$ with cardinality $|C| = N$. We denote the elements of $C$ by $c_i$,

Manuscript received June 30, 1994; revised June 19, 1995. This work was supported by the National Science Foundation Presidential Faculty Fellow (PFF) Award ASC-9350309.

I. Katsavounidis and C.-C. J. Kuo are with the Signal and Image Processing Institute and the Department of Electrical Engineering Systems, University of Southern California, Los Angeles, CA 90089-2564 USA.

Z. Zhang is with the Communication Science Institute and the Department of Electrical Engineering Systems, University of Southern California, Los Angeles, CA 90089-2565 USA.

Publisher Item Identifier S 1057-7149(96)01308-5.