

Vector quantization by hierarchical packing of embedded truncated lattices

Vincent Ricordel and Claude Labit

IRISA/INRIA Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France

fax: (+33) 99.84.71.71,

e-mail: ricordel@irisa.fr, labit@irisa.fr

ABSTRACT

The purpose of this paper is to introduce a new vector quantizer (VQ) which takes place in a temporal-adaptative coding scheme for the compression of digital image sequences.

Our approach, which has to perform a fast codebook construction, unifies both efficient coding methods : a fast lattice encoding and an unbalanced tree-structured codebook design according to a distortion vs. rate tradeoff. Moreover, this tree-structured lattice vector quantizer (TSLVQ) has a convenient property : because of its lattice structure, no reproduction vectors have to be transmitted.

Briefly this TSLVQ technique is based on the hierarchical packing of embedded truncated lattices. We investigate here its design : by, first, explaining how to determine the support lattice and secondly how to obtain the hierarchical set of truncated lattice structures which can be optimally embedded with respect to the hierarchical packing. We then use a simple quantization procedure and describe the corresponding tree-structured codebook. Finally we present two unbalanced tree-structured codebook design algorithms based on the BFOS distortion vs. rate criterion.

Keywords : vector quantization, image sequence compression, coding, lattice structures, pruned trees.

1 CONTEXT OF THE STUDY

This paper deals with the design of a new VQ which takes place in a coding scheme for the compression of digital image sequences.

This paper mainly describes the quantization of an hybrid vector source (i.e an image source of vectors between vectors of motion-compensated prediction errors and vectors of transformed coefficients). Usual approaches consider that such a source has an iid multivariate generalized gaussian distribution. However in practice this input is always a nonstationary signal.

Since the design of a global and optimal codebook (in the sense of yielding the smallest possible distortion for a given rate and for any source) is unfeasible, a codebook design algorithm using a training procedure is well fitted to the image source vector quantization. In this way, a temporal updating of the VQ codebook⁸⁹ can be performed from representative training sequences modelling the source spatiotemporal statistic of

an image sequence.

The figure 1 illustrates the temporal adaptative VQ coding concept, the paper only describes the VQ design of this coder. Therefore our method has to perform a fast codebook design due to a fast encoding algorithm.

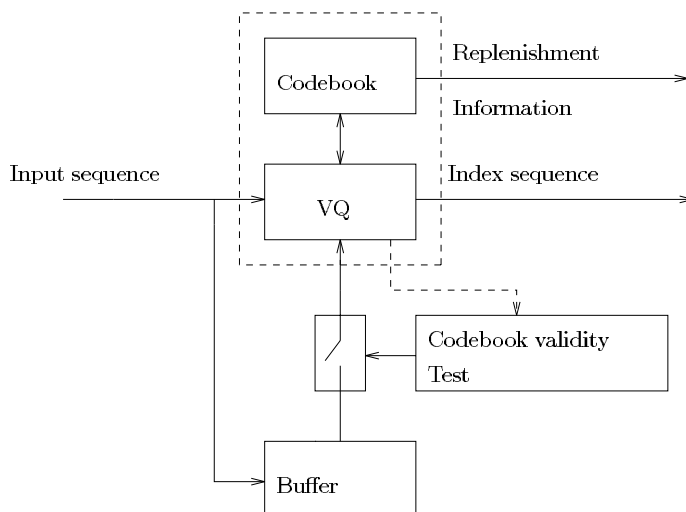


Figure 1: *Temporal adaptative coding scheme. The codebook validity test consists in a mean distorsion-based criterion.*

2 INTRODUCTION

Consider that we code a stationary memoryless vectorial source. According to the asymptotic equipartition property, as the space dimension increases, the vector probability function becomes essentially localized to a compact region of the vector space where the density is almost uniform.¹¹ The optimal condition for vector quantization occurs when all the codevectors are confined to a compact support region. So for a given bit rate (and consequently the codebook size), the average reconstruction error for the input vectors is decreasing. Because of the uniform density, a codebook designed by truncating the highly regular structure of a lattice is well fitted to vector quantization schemes without learning stage.⁶

However, in practice, the design of lattice VQ coder³ produces codevectors with small representativeness (the best quantizing lattices are known in dimensions lower than 25 and the codebook is a subset resulting from truncating the lattice according to a restricted source distribution modeling) and needs efficient entropy-based indexing module.

The generalized Lloyd algorithm⁷ provides the locally optimal codebook for a given training set and a given bit rate, but the computation complexity of usual classification methods (with or without structural constraints) presents a limitation on their applicability for adaptive schemes.

To overcome these drawbacks, we propose a new lattice VQ scheme based on the hierarchical packing of embedded lattices.

The aim of the approach is to unify both efficient coding methods :

- a fast lattice encoding-decoding ;
- a (unbalanced) tree-structured codebook design which permits, in addition to its rapid search property, a space partition adapted to the vector source distribution and to a distortion vs. rate tradeoff.

The paper, which presents the TSLVQ design, is organized as follows :

- first, for a given vector space dimension, we present the TSLVQ support lattice ;
- secondly a lattice truncating method is applied in order to obtain the TSLVQ basic structure : a confined space region which can be, by contracting it, included in a support lattice voronoï cell ;
- then a hierarchical set of truncated lattices is organized (by suitably shifting the basic structure scale) involving the hierarchical packing ;
- we then present the quantization procedure and the corresponding tree-structured codebook design ;
- this tree structure can be pruned according to a distortion vs. rate criterion, consequently we investigate two unbalanced tree-structured codebook design schemes ;
- finally the last section presents simulation results.

3 TSLVQ DESIGN

3.1 Embedded truncated lattices

Some best quantizing lattices are known for space vector dimensions lower than 25 (these lattices are characterized by having the smallest geometric second moments which correspond to quantizing average squared errors per dimension G_n developed by Zador). Precisely we select for our application the support lattices for which Conway and Sloane have determined fast quantizing and decoding algorithms,² that is, unlike LBG-type algorithms, it is practically unnecessary to compute a norm to find the best reproduction vector. These selected lattices are $\mathbf{Z}^n/n \geq 1, D_n/n \geq 3, E_8, A_{16}$.

Briefly speaking, a lattice is a regular arrangement of points in a n -dimensional space centered in zero. We can describe it like a packing of identical spheres in \mathbb{R}^n where the lattice points are the spheres centers, the spheres radius ρ is called the packing radius. Moreover the covering radius R corresponds to the radius of the equal overlapping spheres (also centered at the lattice points) which cover optimally the space. Then, considering the L^2 euclidean metric, the regular structure is specifically truncated for the TSLVQ packing. Effectively we want that the confined space, after contraction, recovers maximally a voronoï cell of the support lattice. So the ideal basic structure is a dilated voronoï cell with a packing radius equal to :

$$r = (2 \times k + 1) \times \rho \text{ with } k \in \mathbb{N}^*$$

The lattice truncation energy is then given by⁵: $E_T = r^2$

Namely, the voronoi cells totally or partially within the multidimensional sphere of radius r constitute the basic confined space.

In high dimension and for large k , an upper bound for the number of lattice points that lie within this subset is calculated using the theta series of the lattice (which tells how many points there are at each distance from the origin) and considering the points into the sphere of radius θ , with :

$$\theta = (2 \times k + 1) \times \rho + R$$

The figure 2 illustrates the method using the simple \mathbf{Z}^2 lattice, without any loss of generality our approach can be generalized to higher dimensions by using high dimension lattices.

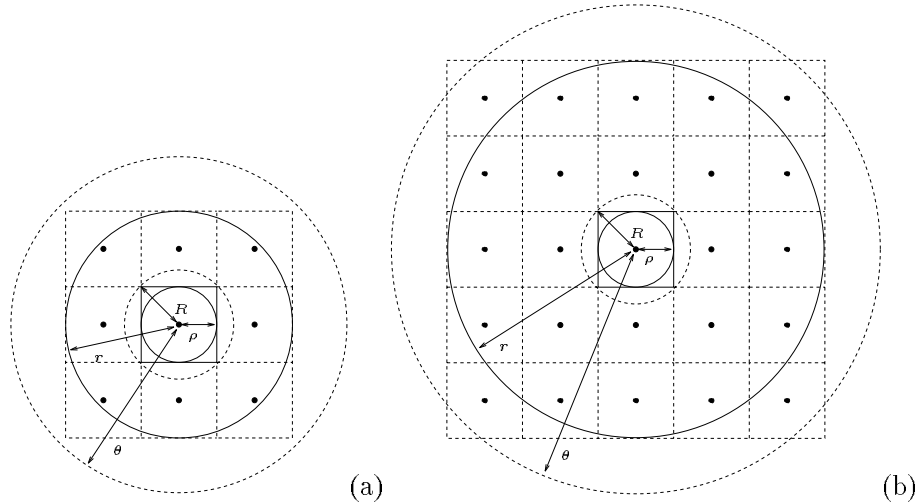


Figure 2: *Truncated \mathbf{Z}^2 lattice considering $k = 1$ (a), and $k = 2$ (b). The squares symbolise the voronoi cells and the dots symbolize the reconstruction vectors.*

3.2 Hierarchical set of truncated lattices

So, from the previous basic truncated space, by shifting its scale, we obtain a hierarchical set of multidimensional regular structures : it is possible, by a simple translation, to include a lower scale truncated lattice in any voronoi cell of the next higher scale structure.

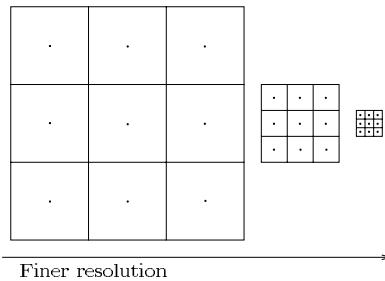


Figure 3: *Hierarchical set corresponding to the previous truncated \mathbf{Z}^2 lattice with $k = 1$.*

3.3 Quantization procedure

First of all, the source vectors are projected into the basic (coarser resolution) truncated lattice. Hence a successive quantization procedure follows : consider the input vectors localized within a voronoï cell of a given lattice structure, the vectors are (re)projected into the truncated lattice at the next finer resolution.

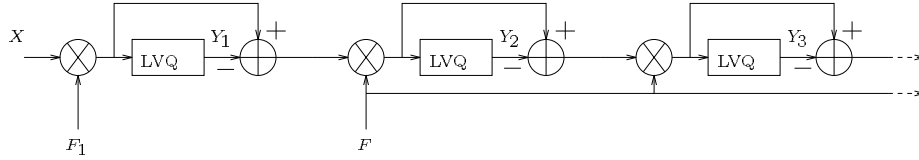


Figure 4: *Quantization scheme.*

The figure 4 illustrates the quantization procedure using successive scaling and translating operators. We have :

- X : source vectors ;
- Y_i : truncated lattice reproduction vectors (they are also the translating vectors) ;
- E_{\max} : maximum energy for the source to be encoded ;
- $F_1 = \sqrt{\frac{E_T}{E_{\max}}}$: scaling factor used in order to project X into the basic confined space ;
- $F = \frac{\sqrt{E_T}}{\rho}$: scaling factor used in order to (re)project the vectors into the next finer resolution lattice space ;

Only the parameters E_{\max} and k have to be fixed. At each stage, instead of scaling the lattice we scale the input vectors in order to use a fast encoding algorithm, and the quantization is performed with the same truncated lattice structure.

We remark an other aspect in figure 4 : contrary to usual multistage VQ, the quantization stages number can be variable for different input vectors (we are going to exploit this property) and only the index corresponding to the final stage reproduction vectors has to be transmitted.

3.4 A tree-structured codebook

So the codebook has a m -ary tree structure, with m corresponding to the basic confined space points number. A node is labeled by a lattice point, its children are labeled by the points of the lattice structure embedded into the node voronoï cell. A tree stage specifies the scale amplitude : the deeper is the tree, the finer is the resolution ; the reconstruction errors of the input vectors decrease while the terminal nodes number increases. The final codebook is the set of the terminal nodes or leaves.

Figures 5 and 6 show tree-structured codebooks examples. The special interest of the hierarchical packing of embedded lattices can be seen in these figures where the space partition is well fitted to the input statistics.

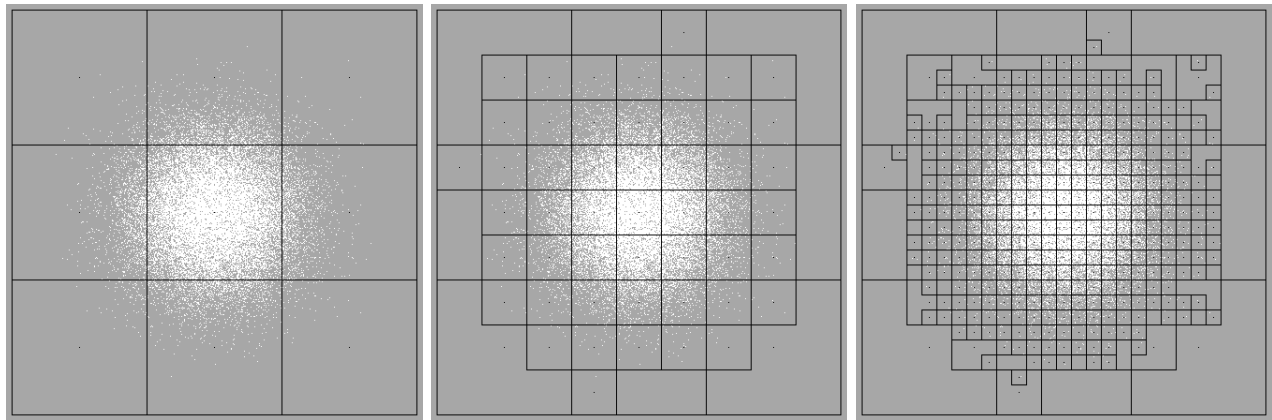


Figure 5: (a, b, c) *Tree-structured codebooks example corresponding to the previous truncated \mathbf{Z}^2 with $k = 1$. We consider 1 quantization stage (a), 2 quantization stages (b) and 3 quantization stages (c). The white dots symbolize the input vectors with iid multivariate gaussian statistic. The black dots symbolise the reproduction vectors.*

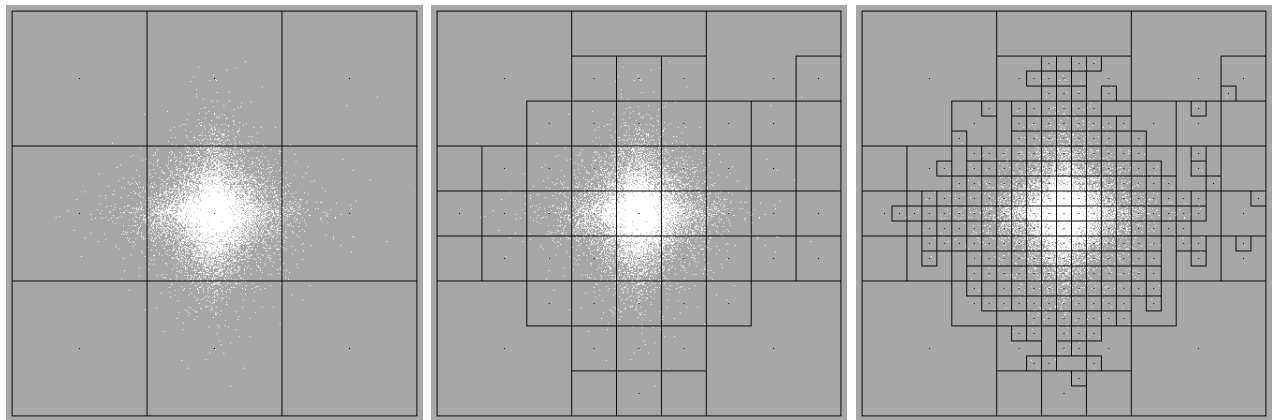


Figure 6: (a, b, c) *Tree-structured codebooks example corresponding to the previous truncated \mathbf{Z}^2 with $k = 1$. We consider 1 quantization stage (a), 2 quantization stages (b) and 3 quantization stages (c). The white dots symbolize the input vectors with iid multivariate laplacian statistic. The black dots symbolise the reproduction vectors.*

A training procedure is performed to design the tree-structured codebook T . Let Y_{t_i} the reproduction vector and C_{t_i} the voronoi cell associated with a node t_i , L is the training sequence size (i.e the number of vectors X). Then, each tree node t_i is characterized by :

- $P(t_i) = \frac{\text{card}(C_{t_i})}{L}$: the probability of reaching t_i when encoding the given source ($\text{card}(C_{t_i})$ is the number of input vectors which lie within C_{t_i}) ;
- $d(t_i) = \frac{1}{\text{card}(C_{t_i})} \cdot \sum_{X \in C_{t_i}} \|X - Y_{t_i}\|^2$: the average distortion resulting from encoding the input vectors that lie within C_{t_i} with Y_{t_i} (it is an average squared error) ;
- $l(t_i) = -\log_2 P(t_i)$: the entropy code length associated with Y_{t_i} .

Since an entropy encoding is used for indexing the tree leaves. An efficient encoding and indexing is achieved if the basic confined points number is highly restricted⁴ ; so k is fixed to 1 for the lattice truncation energy calculation E_T .

3.5 Unbalanced tree-structured codebook design

Two classical strategies have been explored in order to obtain an unbalanced tree according to a distortion vs. rate tradeoff (exactly it is a distortion vs. entropy tradeoff) : a tree pruning and a greedy tree growing approach.

3.5.1 Tree pruning algorithm

We use the BFOS algorithm.¹

First a (balanced) complete tree is designed (because of the storage complexity, this approach is practically suitable only for a lower space dimension when the confined space points number is lowest). After that, this tree is pruned according to a distortion vs. rate criterion.

Namely, if S_{t_i} is a branch of T (i.e a subtree that is rooted at some node $t_i \in T$, and such as \tilde{S}_{t_i} , the branch leaves, are also leaves of T), we have :

- $\delta(S_{t_i}) = \sum_{t_j \in \tilde{S}_{t_i}} P(t_j) \cdot d(t_j)$: the branch average distortion ;
- $l(S_{t_i}) = \sum_{t_j \in \tilde{S}_{t_i}} P(t_j) \cdot l(t_j)$: the branch average entropy code length ;
- $\Delta\delta(S_{t_i}) = P(t_i) \cdot d(t_i) - \delta(S_{t_i})$: the increase in distortion if S_{t_i} is removed ;
- $\Delta l(S_{t_i}) = l(S_{t_i}) - P(t_i) \cdot l(t_i)$: the decrease in rate if S_{t_i} is removed ;
- $\lambda(t_i) = \frac{\Delta\delta(S_{t_i})}{\Delta l(S_{t_i})}$: the BFOS criterion.

$\lambda(t_i)$ can be interpreted as a possible piece-wise of the slope of the experimental distortion vs. rate curve. Thus we successively prune the branches for which $\lambda(t_i)$ is minimal such as the total average distortion is increased as little as possible for a decrease in total average length. Then the resulting sequence of

distortion vs. rate pairs corresponding to the pruned subtrees S (i.e S is a subtree with the same root node as the full tree T) lie on the convex hull of the operational distortion vs. rate function.

The BFOS algorithm could be considered as using a lagrangian optimization technique. It minimizes the functional $J(S) = \delta(S) + \lambda I(S)$, where λ is interpreted as a lagrangian multiplier (the slope of the distortion vs. rate curve) : varying λ , all of the distortion vs. rate pairs on the convex hull of the operational distortion vs. rate function can be found.

The average distortion or the average entropy code length associated to the pruned subtree enables to interrupt this codebook design procedure.

3.5.2 Greedy tree algorithm

In this part we don't use a complete tree as the starting point for the pruning process : any tree structure can be used, an individual leaf splitting (i.e we include a basic lattice structure into its associated voronoi cell) is applied to grow the tree which can then be pruned afterwards.

Namely, we split the leaf that provides the best tradeoff between total average distortion over all terminal nodes and total average entropy. Considering the previous notations, $\Delta\delta$ specifies the decrease in average distortion and Δl specifies the increase in average length resulting from a specific node split. A splitting procedure on the leaves provides the largest possible slope $|\Delta\delta/\Delta l|$, namely we split one leaf so that the average distortion is decreased as much as possible for an increase in average length and hence the magnitude slope is maximized.

This algorithm¹⁰ is greedy in that it only considers the short term effect of extending the tree ; it considers what happens with the addition of a single add of leaves grow from the current leaf.

The average distortion or the average entropy code length associated to the greedy growing tree enables to interrupt this codebook design procedure.

Finally the overall information that we have to transmit to characterized the TSLVQ codebook is constituted by :

- E_T ,
- the unbalanced tree,
- the entropy index corresponding to the tree leaves.

Because of the lattice predefined structure no reproduction vectors have to be transmitted.

4 EXPERIMENTAL RESULTS

Comparative experiments applied to iid multivariate generalized gaussian synthetic sources have been performed. The training ratio (i.e the ratio of L on the leaves number) is upper than 150 in order to limit the reproduction vectors number.

The figures 7 and 8 illustrate the tree pruning approach with \mathbf{Z}^2 lattice. The tree growing approach is illustrated by the figure 9 with D_4 lattice.

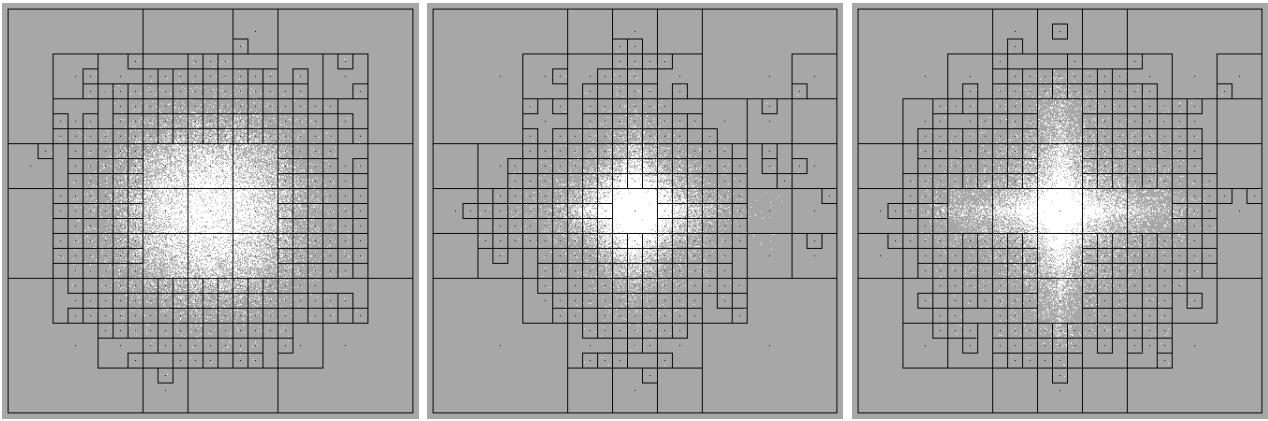


Figure 7: (a, b, c) *Tree pruning approach illustration considering respectively a gaussian distribution (a), a laplacian distribution (b) and generalized gaussian distribution (c).*

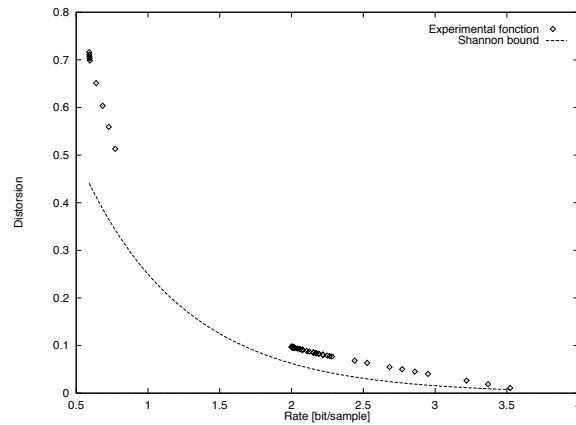


Figure 8: *Tree pruning approach with \mathbf{Z}^2 lattice : distortion rate function for memoryless Gaussian distribution source ($\sigma^2 = 1$).*

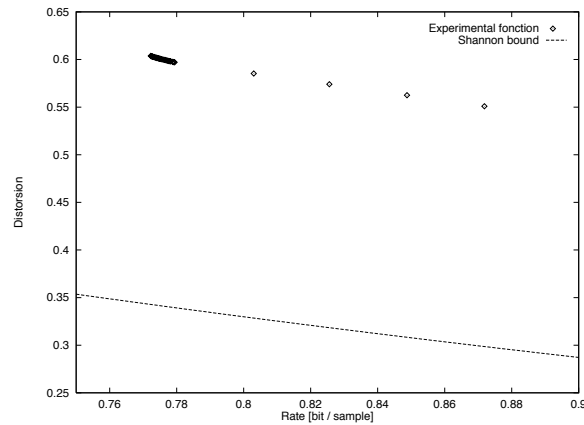


Figure 9: *Tree growing approach with D_4 lattice : distortion rate function extract for memoryless Gaussian distribution source ($\sigma^2 = 1$).*

The figure 7 shows how an unbalanced tree-structured codebook design using our method is adapted to the differential or hybrid image source coding : for a given rate, the high-density space region (where are located the lowest error magnitudes) is coarsely quantized in order to permit a finer coding of the low-density space region (where lie relevant vectors).

The figures 8 and 9 show examples of experimental distortion vs. rate curves, the figures include for comparison Shannon's distortion vs. rate function for the considered source. The interest of using higher dimensions is demonstrated since TSLVQ with D_4 lattice performs better than TSLVQ with \mathbf{Z}^2 lattice (a lower distortion is achieved for a lower rate). Very low bit rate will be achievable only with higher space dimensions ; this point will be tested in our further investigations and applied to real spatiotemporal sources.

5 REFERENCES

- [1] L. Breiman, J.H. Friedman, R.A Olshen, and C.J. Stone. – *Classification and regression Trees.* – The Wadsworth Statistics/Probability Series. Wadsworth, Belmont, California, 1984.
- [2] J.H. Conway and Sloane N.J.A. – Fast quantizing and decoding algorithms for lattice quantizers and codes. – *IEEE Trans. Inform. Theory*, IT-28(2):526–533, 1982.
- [3] J.H. Conway and Sloane N.J.A. – *Sphere Packings, Lattices and Groups.* – Springer-Verlag, 2 edition, 1993.
- [4] T. Eriksson. – Multistage vector quantization with dynamic bit allocation. – In *EUSIPCO*. Edinburgh, 1994.
- [5] T.R. Fisher. – A pyramid vector quantizer. – *IEEE Trans. Inform. Theory*, IT-32(4):568–583, 1986.
- [6] A. Gersho. – Asymptotically optimal block quantization. – *IEEE Trans. Inform. Theory*, IT-25(4):373–380, 1979.
- [7] A. Gersho and R.M. Gray. – *Vector Quantization and Signal Compression.* – Kluwer Academic Publishers, 1992.
- [8] M. Golberg, P.R. Boucher, and S. Schlien. – Image compression using adaptative vector quantization. – *IEEE Trans. Comm.*, 34(2):180–187, 1986.
- [9] P. Monet and C. Labit. – Codebook replenishment in classified pruned tree-structured vector quantization of image sequences. – *ICASSP*, pages 2285–2288, 1990.
- [10] E.A. Riskin. – A greedy tree growing algorithm for the design of variable rate vector quantizers. – *IEEE Trans. Signal Process.*, 39(11):2500–2507, 1991.
- [11] P. Zador. – Asymptotic quantization error of continuous signals and their quantization dimension. – *IEEE Trans. Inform. Theory*, IT-28, 1982.