

DESIGN OF SUCCESSIVE APPROXIMATION LATTICE VECTOR QUANTIZERS

Stephan F. Simon

Lambert Bosse

Institut für Elektrische Nachrichtentechnik
Aachen University of Technology (RWTH)
52056 Aachen, Germany
simon@ient.rwth-aachen.de

ABSTRACT

Two methods to overcome the problems with large vector quantization (VQ) codebooks are lattice VQ (LVQ) and product codes. The approach described in this paper takes advantage of both methods by applying residual VQ with LVQ at all stages. Using LVQ in conjunction with entropy coding is strongly motivated by the fact that entropy constrained but structurally unconstrained VQ design leads to more equally sized VQ cells. The entropy code of the first LVQ stage should aim at exploiting the statistical properties of the source. The refinement LVQ stages quantize the residuals. Simulations show that there exist certain scales of the refinement lattices yielding extraordinary performance. We focus on the search of these scales.

1. INTRODUCTION

Successive approximation data descriptions are naturally obtained using residual vector quantization (RVQ) [1]. Residual vector quantizers use a sequence of encoder stages where each stage encodes the residual vector of the prior stage. RVQ is one of several methods of imposing a product code on a vector quantizer in order to reduce computation and memory requirements simultaneously. The codebook size of the product code is then equal to the product over all stage codebook sizes. Successive approximation of an input vector opens the possibility to precisely control the bitrate. This property is required in many source coding applications, e.g. hierarchical or scalable image or video coding.

A two-stage vector quantizer with an unstructured codebook at the first stage and lattice vector quantization (LVQ) for the quantization of the approximately Gaussian residual was presented in [2]. The effectiveness of that method is dependent on the feasibility of using a large enough first-stage codebook to exploit most of the source memory.

In this paper, multi-stage RVQ with lattice vector quantization at *all* stages is considered. LVQ offers extremely fast algorithms which find the nearest lattice point for a given input vector. Furthermore it requires no memory for codebook storage if the lattice points are used as decoder codebook and enumeration algorithms (see e.g. [3]) are applied. Entropy coded LVQ at the first stage makes large vector dimensions possible and facilitates the exploitation of the source memory. Using entropy coded LVQ is strongly motivated by the fact that entropy constrained but structurally unconstrained VQ design leads to more equally sized VQ cells [4].

Either concatenated codewords for (pyramid) radius coding and enumeration or usual entropy codewords can be used. The first method allows the use of a large first-stage codebook. The latter method requires more memory for the codeword storage but allows a better adaptation to the joint probability density function of the source.

With LVQ at all stages, the shape of the Voronoi cells

of the prior stage determine the boundary of the input distribution of the following stage. For the first refinement stage, the assumption of an equal distribution inside the cell boundary is reasonable if the source distribution is smooth and the first-stage codebook is not too small. If the equal distribution assumption is true for all stages, the determination of the rate-distortion performance of any refinement stage is source-independent and can be regarded as a geometrical problem.

The following assumptions are made:

- Given the dimension of the input vectors, the best known lattice vector quantizer for this dimension shall be used in all the stages. (Refer to [5] for the types of lattices and [6] for fast quantization algorithms). Then, the relation between the lattices of two consecutive stages can be characterized by a scale factor, a translation vector and a rotation matrix. These parameters describe the transformation from the lattice of stage p into the lattice of stage $(p + 1)$ (and vice versa). We denote them as *transformation parameters*.
- In the refinement stages (all stages except for the first), the residual error is assumed to be equally distributed inside the shape of a Voronoi region of the prior stage. This assumption needs to be discussed in section 2.3.
- Entropy coding is used for the indices of the codebook vectors.

Section 2. will introduce and illustrate these assumptions and their consequences in detail. The resulting residual LVQ model will allow us to compute operational rate-distortion functions (ORDFs) for the refinement stages. We will observe that – depending on the transformation parameters – outstanding points with exceptional performance appear in the ORDFs. Sections 3. and 4. address the questions how these points can be found and how RVQ design can benefit by this knowledge.

2. A MULTIPLE STAGE LATTICE VECTOR QUANTIZER MODEL

We consider two consecutive stages of a residual lattice vector quantizer. Let the “fine” lattice¹ Λ_f of stage $(p + 1)$ be defined as the set

$$\Lambda_f = \left\{ \mathbf{v}_f \in \mathbb{R}^n : \mathbf{v}_f = \mathbf{s} + \sum_{i=1}^n k_i \mathbf{a}_i, k_i \in \mathbb{Z} \right\} \quad (1)$$

of all integral combinations of n linearly independent basis vectors $\mathbf{a}_i \in \mathbb{R}^n, (i = 1, \dots, n)$, shifted by \mathbf{s} . The “coarse”

¹The indices c and f are used for “coarse” and “fine” lattice, respectively.

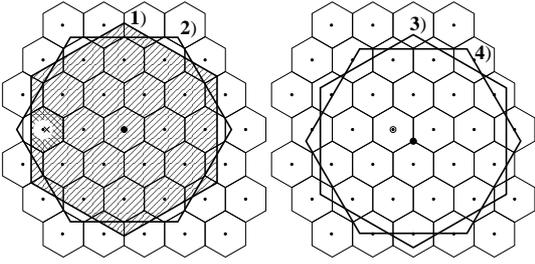


Figure 1. 1) no translation, no rotation, 2) no translation, rotation, 3) translation, no rotation, 4) translation and rotation.

lattice of stage p

$$\Lambda_c = \left\{ \mathbf{v}_c \in \mathbb{R}^n : \mathbf{v}_c = \alpha \mathbf{T} \sum_{i=1}^n k_i \mathbf{a}_i, k_i \in \mathbb{Z} \right\} \quad (2)$$

can be obtained from the fine lattice by applying the inverse shift, a rotation with rotation matrix \mathbf{T} , $\det(\mathbf{T}) = 1$ and a scaling with $\alpha \geq 1$. The nearest-neighbor region (so-called Dirichlet- or Voronoi region) of a lattice point $\mathbf{v} \in \Lambda$ can be defined as

$$\mathcal{C}(\mathbf{v}) = \{ \mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{v}\| \leq \|\mathbf{x} - \mathbf{z}\|, \forall \mathbf{z} \in \Lambda \setminus \{\mathbf{v}\} \}. \quad (3)$$

We also introduce the notation

$$\mathcal{C}(\mathbf{v}) + \mathbf{y} = \{ \mathbf{x} + \mathbf{y} : \mathbf{x} \in \mathcal{C}(\mathbf{v}) \}, \quad \mathbf{y} \in \mathbb{R}^n \quad (4)$$

which describes a translated Voronoi region.

\mathcal{C}_f and \mathcal{C}_c shall denote Voronoi regions of points from Λ_f and Λ_c , respectively. If \mathbf{v}_c is used as reconstruction vector of the region $\mathcal{C}_c(\mathbf{v}_c)$, the residual vector will always lie in $\mathcal{C}_c(\mathbf{v}_c) - \mathbf{v}_c = \mathcal{C}_c(\mathbf{0})$.

Fig. 1 shows an example with A_2 -lattices: the hatched area is $\mathcal{C}_c(\mathbf{0})$. The fine lattice Λ_f is used for the quantization of the residual error. For a Voronoi cell $\mathcal{C}_f(\mathbf{v}_f)$ of a point from the fine lattice, three cases can be distinguished: either

1. it lies completely inside $\mathcal{C}_c(\mathbf{0})$ such that $\mathcal{C}_f(\mathbf{v}_f) \cap \mathcal{C}_c(\mathbf{0}) = \mathcal{C}_f(\mathbf{v}_f)$, or
2. it is intersected by the border of $\mathcal{C}_c(\mathbf{0})$, or
3. it lies completely outside $\mathcal{C}_c(\mathbf{0})$, such that $\mathcal{C}_f(\mathbf{v}_f) \cap \mathcal{C}_c(\mathbf{0}) = \emptyset$.

Using the mean squared error criterion and assuming an equal distribution of the residual error over $\mathcal{C}_c(\mathbf{0})$, the optimal (decoder) reconstruction vectors are $\mathbf{y} = \mathbf{v}_f$ in case 1 and

$$\mathbf{y} = V(\mathcal{C}_f(\mathbf{v}_f) \cap \mathcal{C}_c(\mathbf{0}))^{-1} \int_{\mathcal{C}_f(\mathbf{v}_f) \cap \mathcal{C}_c(\mathbf{0})} \mathbf{x} d\mathbf{x} \quad (5)$$

in case 2, with the volume $V(\mathcal{R}) = \int_{\mathcal{R}} d\mathbf{x}$. If not stated otherwise, optimal reconstruction vectors will be used. One such vector is marked by an “x” in Fig. 1. The probability for the cell of a point \mathbf{v}_i from the fine lattice is given by

$$P_i = \frac{1}{V_c} \int_{\mathcal{C}_f(\mathbf{v}_i) \cap \mathcal{C}_c(\mathbf{0})} d\mathbf{x} \quad (6)$$

with the shortcut $V_c = V(\mathcal{C}_c(\mathbf{0}))$. Further, Fig. 1 shows four different methods (from an infinite number of possible choices) for the transformation parameters. For methods 1) and 2) both lattices have the same origin, whereas for

methods 3) and 4) the fine lattice is shifted such that a *deep hole*² of Λ_f falls onto $\mathbf{0}$. For methods 1) and 3) no rotation is used, for methods 2) and 4) the rotation angle is $\frac{\pi}{6}$.

Using the mean squared error, we have

$$D_c = \frac{1}{nV_c} \int_{\mathcal{C}_c(\mathbf{0})} \mathbf{x}^T \mathbf{x} d\mathbf{x}, \quad (7)$$

$$D_f = \frac{1}{nV_c} \sum_{\mathbf{v}_i \in \Lambda_f} \int_{\mathcal{C}_f(\mathbf{v}_i) \cap \mathcal{C}_c(\mathbf{0})} (\mathbf{x} - \mathbf{v}_i)^T (\mathbf{x} - \mathbf{v}_i) d\mathbf{x}, \quad (8)$$

for signal and noise per dimension, respectively. The factor V_c^{-1} represents the constant probability density function. Assuming the self-information $l_i = -\log_2 P_i$ as approximation for the number of bits needed to represent the index i , we are now equipped to compute operational rate-distortion functions (ORDFs) for the refinement stage. Section 2.2. will provide an example.

2.1. Implementation Issues

The ORDF computations require multidimensional integrations of (5), (6) and (8). Describing the boundaries of the region $\mathcal{C}_f(\mathbf{v}_f) \cap \mathcal{C}_c(\mathbf{0})$ analytically is a difficult problem if the dimension n is large. We solved the problem numerically using Sobol’s quasirandom sequence generator which is suitable for multidimensional integration [7].

Attention must be paid with the numerical evaluation of (8): let us assume that the “Sobol-set” $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of size N covers the region $\mathcal{R} = \mathcal{C}_f(\mathbf{v}_f) \cap \mathcal{C}_c(\mathbf{0})$ with a “quasi-equal” distribution. The integral for the distortion contributed by \mathcal{R} with reconstruction vector \mathbf{v}

$$d_f = \frac{1}{nV(\mathcal{R})} \int_{\mathcal{R}} (\mathbf{x} - \mathbf{v})^T (\mathbf{x} - \mathbf{v}) d\mathbf{x} \quad (9)$$

will then be approximated by

$$\hat{d}_f = e + \frac{1}{nN} \sum_{s=1}^N (\mathbf{x}_s - \mathbf{v})^T (\mathbf{x}_s - \mathbf{v}). \quad (10)$$

The expectation for the error e is a positive value which approximates 0 as $N \rightarrow \infty$. A thorough analysis of this error yields the simple result

$$e \approx \frac{1}{N} \sum_{s=1}^N G_s V_s^{\frac{2}{n}} \quad \text{with} \quad (11)$$

$$G_s = V_s^{-1 - \frac{2}{n}} \frac{1}{n} \int_{\mathcal{S}_s - \mathbf{g}_s} \mathbf{x}^T \mathbf{x} d\mathbf{x}, \quad (12)$$

where G_s is the *normalized second moment* of the small Voronoi cell \mathcal{S}_s assigned to the point \mathbf{x}_s from the Sobol-set with volume $V_s = \int_{\mathcal{S}_s} d\mathbf{x}$ and center of gravity $\mathbf{g}_s = \int_{\mathcal{S}_s} \mathbf{x} d\mathbf{x}$. The only required simplification in (11) is the assumption $\mathbf{g}_s \approx \mathbf{x}_s$. Further assuming that an n -dimensional Sobol-set produces Voronoi cells with a “typical” normalized second moment, say $G^{(n)}$, a simplified version can be obtained

$$e \approx \frac{G^{(n)}}{N} \sum_{s=1}^N V_s^{\frac{2}{n}}. \quad (13)$$

²A *deep hole* is a point with maximum distance from the lattice points. Its distance from the closest lattice points is called covering radius R .

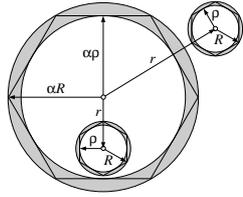


Figure 2. Covering radius and inradius of Λ_f and Λ_c .

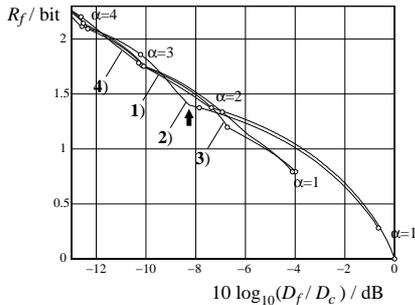


Figure 3. Rate-distortion functions for 2-dimensional residual lattice VQ. The small circles mark points where the scale-factor α is integer-valued.

Because of the “quasi-equal” distribution of the Sobol-set, this can be further simplified to obtain

$$e \approx \frac{G^{(n)}}{N} V^{\frac{2}{n}}, \quad (14)$$

where V is the average volume of the Voronoi cells of the points from the Sobol-set. We now have a correction term which improves the exactness of the numerical evaluation of (8) or allows to reduce the size of the Sobol-set.

There is even more potential for the increase of exactness and execution speed: numerical integrations are only needed for points of the refinement lattice having distances r from the origin satisfying $\alpha\rho - R \leq r \leq \alpha R + R$ (see Fig. 2), where ρ is the inradius and R the covering radius of Λ_f . For the other case $r < \alpha\rho - R$, the distortion introduced by a random variable with equal distribution over a lattice Voronoi region is tabulated e.g. in [5].

2.2. Example

Fig. 3 shows the results of the four methods introduced in Fig. 1 in an average rate per dimension (R_f) versus SNR plot. Two important details can be read from Fig. 3:

- If an individual ORDF is regarded, we find a few outstanding points for which the rate-distortion performance is remarkably good. (E.g. the arrow marks the point -8.252 dB, 1.399 bit of curve 2) with $\alpha = 2.32$).
- If the ORDFs are compared to each other, we find that none of the four methods is always superior. The best choice depends on the desired rate or SNR.

2.3. The Equal Distribution Assumption

The assumption of an equally distributed residual error inside a Voronoi cell boundary of the prior stage enabled source-independent determination of the rate-distortion performance of the refinement stage. The real distribution may differ from this assumption. We have to consider under which conditions the assumption is appropriate. More precisely, we are not interested in the exact distribution itself, but in the question if the coding results for real distribution

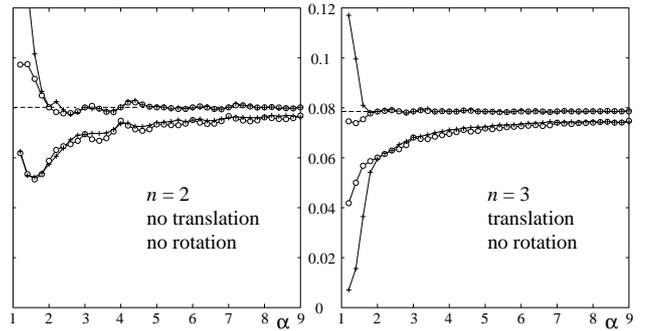


Figure 4. Comparison of stage $p = 1$ (o) and stage $p = 10$ (+) in terms of normalized D_f . Dashed: normalized second moment G . Left: method 1) for $n = 2$. Right: method 3) for $n = 3$.

and equal distribution differ significantly. If not, together with our knowledge about the emergence of the distribution, it is reasonable to claim that the equal distribution assumption is justified.

The idea is the following: starting with an equal distribution as input to the first stage, repeated quantization of the residual error is performed. All transitions from stage p to stage $(p + 1)$, $p \in \mathbb{N}$, shall use the same transition parameters. It can be observed that D_f/D_c converges after a few stages, long before floating point accuracy is reached. Now, let us compare stage one with a converged stage, say e.g. stage 10, in terms of D_f/D_c where D_f is a measured value and D_c is the distortion for an equally distributed input (that is what we actually use in the simulations). Thus, considering our simulations, stage one is the model we use for *all* following stages. Now, we have to check how good its results correspond to those of the “real” stage 10. In both cases, we normalize D_f such that a comparison with the known normalized second moment

$$G = V_c^{-1 - \frac{2}{n}} \int_{c_0(0)} \mathbf{x}^T \mathbf{x} dx \quad (15)$$

of a Voronoi cell of the considered lattice is also possible (dashed lines in Fig. 4). To give some examples, Figure 4 shows the results after stage $p = 1$ (circles) and stage $p = 10$ (plus-signs) for $n = 2$ with method 1) (left plot) and $n = 3$ with method 3). The upper pairs of curves were obtained using the lattice points (encoder codebook) as reconstruction vectors. The lower two pairs of curves were obtained with optimal reconstruction vectors (decoder codebook). We observe that for $\alpha > 2$ always a good correspondence between the results of stages 1 and 10 can be observed. This justifies the equal distribution assumption.

Additionally, for this investigation we found that the difference between (14) and the more general version (13) is negligible if $n \geq 2$.

3. FINDING GOOD LATTICE SCALES

Using a Lagrangian formulation like

$$J_\lambda = \frac{D_f}{D_c} + \lambda R \quad (16)$$

we usually solve the problem of minimizing distortion subject to a constraint on the entropy by minimizing J_λ . In a practical coder, the parameter λ may be driven by a buffer control unit which manages the tradeoff between rate and distortion. Minimizing J_λ for a given ORDF and a given factor λ yields an optimum operation point which will for our parametric representation be expressed in terms of the

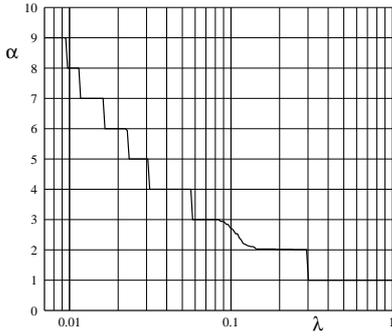


Figure 5. Optimal scale factors $\alpha(\lambda)$ for method 1).

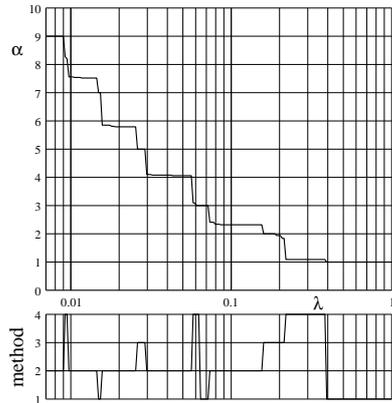


Figure 6. Optimal scale factors $\alpha(\lambda)$ for methods 1) - 4) using A_2 -lattice.

scale factor α . Fig. 5 shows the result of the minimization for method 1) of the two-dimensional example (without translation, without rotation)³. One would expect $\alpha(\lambda)$ to be a strictly decreasing function (because ORDFs are usually convex), but this is not true: $\alpha(\lambda)$ is more or less a staircase function, preferring the above-mentioned points with exceptional performance. Practically, this implies that switching e.g. between $\alpha = 3$ and $\alpha = 4$ yields better performance than using $\alpha = 3.5$. Fig. 6 shows the result when additionally the best of all permitted methods is searched. The lower curve shows the selected method.

4. RESULTS AND DISCUSSION

Further simulations were performed for dimensions $n = 3$ and 4 using the lattices D_3^* and D_4 , respectively. In order to keep the number of different choices of the transformation parameters handy, for D_3^* only a translation onto a deep hole and a rotation maximizing the number of parallel facets of \mathcal{C}_c and \mathcal{C}_f were used. For D_4 , only the translation onto a deep hole was used as alternative. So, we again have four different choices for D_3^* and two different choices for D_4 . The resulting optimal $\alpha(\lambda)$ functions are shown in Figures 7 and 8, where the numbers 1) - 4) have the same meaning as in the previous examples.

Regard again Fig. 5: except for a small region around $\lambda = 0.1$, the optimal adaptation of the considered quantizer to an arbitrary factor λ requires only a small number of scale factors. Therefore, transmission of the selected scale factor only needs a very small amount of overhead information. If the RVQ is allowed to switch between several methods, rate-distortion performance is generally improved. But

³In the example of Fig. 5, α happens to assume integer values.

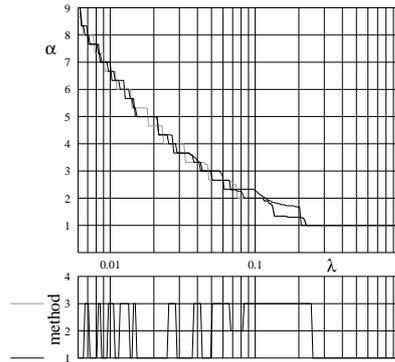


Figure 7. Optimal scale factors $\alpha(\lambda)$ for methods 1) - 4) using D_3^* -lattice. Thick line: best method, thin line: method 1), gray line: method 3).

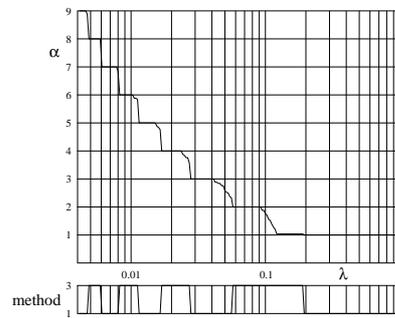


Figure 8. Optimal scale factors $\alpha(\lambda)$ for methods 1) and 3) using D_4 -lattice.

the number of permitted transformation parameter sets – and hereby the amount of overhead information – is increased, too (compare Figures 5 and 6). This may compensate the rate-distortion improvement if the factor λ changes frequently.

REFERENCES

- [1] C. F. Barnes, S. A. Rizvi, and N. M. Nasrabadi, “Advances in residual vector quantization: a review,” *IEEE Trans. Image Proc.*, vol. 5, pp. 226–262, Feb. 1996.
- [2] J. Pan and T. R. Fischer, “Two-stage vector quantization – lattice vector quantization,” *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 155–163, Jan. 1995.
- [3] Z. Mohd-Yusof and T. R. Fischer, “An entropy-coded lattice vector quantizer for transform and subband image coding,” *IEEE Trans. Image Proc.*, vol. 5, pp. 289–298, Feb. 1996.
- [4] S. F. Simon, “On the sizes of Voronoi cells in entropy-constrained vector quantization,” in *VIII European Signal Processing Conference EUSIPCO-96*, vol. 1, (Trieste, Italy), pp. 347–350, Sept. 1996.
- [5] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*. New York, Berlin, Heidelberg: Springer-Verlag, 2nd ed., 1993.
- [6] J. H. Conway and N. J. A. Sloane, “Fast quantizing and decoding algorithms for lattice quantizers and codes,” *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 227–232, Mar. 1982.
- [7] P. Bratley and B. L. Fox, “Algorithm 659. Implementing Sobol’s quasirandom sequence generator,” *ACM Transactions on Mathematical Software*, vol. 14, pp. 88–100, Mar. 1988.