Lattice-Based Quantization
Part I

by

Erik Agrell and Thomas Eriksson

Lattice-Based Quantization, Part I

by

Erik Agrell and Thomas Eriksson
Department of Information Theory
Chalmers University of Technology
Göteborg, Sweden

Technical report no. 17
Department of Information Theory
Chalmers University of Technology
Göteborg, Sweden
Oct., 1996

ISSN 0283–1260

ABSTRACT

A training algorithm for the design of lattices for vector quantization is presented. The algorithm uses a steepest descent method to adjust a generator matrix, in the search for a lattice whose Voronoi regions have minimal normalized second moment. Experiments show that the algorithm is stable, in the sense that many independent runs reach equivalent lattices. The obtained lattices reach as low second moments as the best previously reported lattices, or even lower. Specifically, we report lattices in 9 and 10 dimensions with normalized second moments of 0.0716 and 0.0708, respectively, and nonlattice tessellations in 7 and 9 dimensions with 0.0727 and 0.0711, which improves on previously known values. The new 9- and 10-dimensional lattices suggest that Conway and Sloane's conjecture on the duality between the optimal lattices for packing and quantization might be false. A discussion of the application of lattices in vector quantizer design for various sources, uniform and nonuniform, is included.

CONTENTS

I. INTRODUCTION: VECTOR QUANTIZATION AND LATTICES

Lattices are widely recognized as an important tool in the design of vector quantizers, not only for uniform sources. The design can be thought of as two independent problems: the choice of a suitable lattice and the creation of a codebook based on a subset of the lattice. The present report considers the first of these problems, and the second is studied in a companion report, [1].

To select a good lattice, one can of course rely on written sources, such as [2], where many lattices and their properties are tabulated. However, there is reason to believe that the best d-dimensional lattice has not yet been found for every d (see, e.g., figure 3.1). Perhaps there is some knowledge to be gained through an approach completely different from the algebraic methods that have been dominating lattice design? This was the question that triggered the present work, and the answer we found was affirmative.

We propose an algorithm for lattice design that can be used with a minimum of insight into algebra and lattice theory. The algorithm employs a numerical algorithm to iteratively improve a given lattice, in a manner that parallels traditional training methods for the design of unconstrained vector quantizers.

This chapter introduces the background and preliminaries for the work. Section 1.1 is a brief review of the fundamentals of vector quantization and its terminology. In section 1.2, we then apply vector quantization to uniform sources, and explain why a lattice is a commonly employed structure of uniform quantizers. After a summary of some lattice theory in section 1.3, we return to the problem of vector quantizer design in section 1.4. This section, which is essentially a literature survey, presents various strategies to design lattice-based vector quantizers for nonuniform sources, which is not as straightforward as in the uniform case.

The lattice training algorithm is presented in chapter II. In chapter III, experiments with the algorithm are reported, which lead to improvements on previously known results in dimensions 7, 9, and 10. Chapter IV is a summary.

1.1 Vector Quantization

A *vector quantizer* is a general utility for digital representation of multidimensional data. Its input is a real-valued vector $\mathbf{x}$ and its output is one of a finite number of *codevectors* $(\mathbf{c}_1,\cdots,\mathbf{c}_N)$, which is selected to approximate $\mathbf{x}$ as well as possible, according to some

criterion. The codevector $\mathbf{c}_i$ can, through its integer index i, be represented using $^2\log N$ bits. The *rate* R is the number of bits used to quantize one scalar, that is, $R = {}^2\log N/d$, where d is the *dimension* of the quantizer, in other words, the number of components in $\mathbf{x}$ and $\mathbf{c}_i$.

The quantization is governed by a function $Q: \ \mathbb{R}^d \to \mathcal{C}$, where $\mathcal{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_N\}$ is the *codebook.* This function should be chosen to optimize some quality measure for a given source. The standard quality measure is the minimum mean square error, or *distortion,* per vector,

$$D = \int_{\mathbb{R}^d} \|\mathbf{x} - Q(\mathbf{x})\|^2 \, f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{1.1}$$

where $f_{\mathbf{x}}(\mathbf{x})$ is the probability density function of the source vectors $\mathbf{x}$. If the codebook is given, the optimal quantization function is to simply choose the *closest* codevector in the Euclidean sense,

$$Q(\mathbf{x}) = \arg\min_{\mathbf{c} \in \mathcal{C}} \|\mathbf{x} - \mathbf{c}\|^2 \tag{1.2}$$

This rule reduces the problem of vector quantizer design to finding a point constellation for use as a codebook.

Many input vectors $\mathbf{x}$ yield the same output vector $\mathbf{c}_i$. The set of *all* input vectors that are encoded as the same codevector is called a *Voronoi region,*

$$\Omega_k = \left\{ \mathbf{x} \in \mathbb{R}^d : \ \ Q(\mathbf{x}) = \mathbf{c}_k \right\} \tag{1.3}$$

Hence, the function $Q(\cdot)$ partitions d-dimensional source space into N Voronoi regions, without neither gaps nor overlaps. In terms of Voronoi regions, the distortion (1.1) can be separated into the contributions by each codevector:

$$D = \sum_{i=1}^{N} \int_{\Omega_i} \|\mathbf{x} - \mathbf{c}_i\|^2 \, f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{1.4}$$

In the next section, this expression will be specialized to the case of uniform sources.

The most common way to design a vector quantizer is to generate a large set of source samples, a *training database,* and iteratively adjust ("train") an initial codebook, in order to decrease an estimate of the distortion, based on the training database. Among the large number of training algorithms that have been proposed, we mention [3],[1] [4], [5, chs. 5 and 7], and [6].

In this report, an alternative approach for vector quantizer design is studied: lattice-based design. The general idea is to find a lattice with attractive properties and

[1] Lloyds original manuscript, which although unpublished has become famous, is dated 1957.

subsequently shape a subset thereof to the source. The focus of this report is on the lattice itself; truncation and modifications of lattices to suit various sources are discussed in section 1.4, and in the second part of this work [1].

1.2 Quantizer Design for Uniform Sources

This section summarizes the application of vector quantization to uniform sources. Suppose that the source probability density function is uniform within a region Δ,

$$f_{\mathbf{x}}(\mathbf{x}) = \begin{cases} \dfrac{1}{\mathrm{vol}(\Delta)} & \text{if } \mathbf{x} \in \Delta \\ 0 & \text{if } \mathbf{x} \notin \Delta \end{cases} \tag{1.5}$$

where $\mathrm{vol}(\Psi)$ denotes the d-dimensional volume of a region $\Psi \subset \mathbb{R}^d$. Then the distortion (1.4) becomes

$$D = \frac{1}{\mathrm{vol}(\Delta)} \sum_{i=1}^{N} \int_{\Omega_i \cap \Delta} \|\mathbf{x} - \mathbf{c}_i\|^2 \, d\mathbf{x} \tag{1.6}$$

Now we concentrate on what happens when the rate R is high, for a constant dimension d. The region Δ then becomes partitioned into a large number, N, of Voronoi regions, each one contributing a small amount to the overall distortion D. According to a well-known conjecture in quantization theory, first posed by Gersho [7], *almost all the Voronoi regions will be similar to each other* in the optimal vector quantizer. In other words, there exists a *typical body* that, through proper scaling, rotation, reflection, and translation, will approximate most of the Voronoi regions.

We will now, supported by Gersho's conjecture, make the approximation that *all* Voronoi regions are congruent to a typical body Ω_t. Moreover, since the source under consideration is uniform, we assume that all regions have the same size, $\alpha\Omega_t$,[2] where α is a rate-dependent scaling parameter to be determined below. This approximation contains two errors, for any finite rate. Firstly, the regions in (1.6) deviate a little from $\alpha\Omega_t$; secondly, a few of the regions, notably those close to the boundary of Δ, deviate a lot. How these errors are handled implicitly selects one of two concepts for vector quantizer design for uniform sources. The errors can be neglected, which is the basic assumption behind lattice quantization, or they can be considered, which leads into unconstrained quantizer design. In this report, we follow the former approach.

If all Voronoi regions are congruent, the sum in (1.6) is not needed anymore:

---

[2] We will allow the following operation on a set Ψ of vectors: Elementwise multiplication by a scalar a, denoted $a\Psi$, and elementwise addition of a vector $\mathbf{a}$, denoted $\Psi + \mathbf{a}$.

$$D \approx \frac{1}{\mathrm{vol}(\Delta)} N \int_{\alpha\Omega_t} \|\mathbf{x} - \alpha\mathbf{c}_t\|^2 d\mathbf{x}$$
$$= \frac{1}{\mathrm{vol}(\Delta)} N \int_{\Omega_t} \|\alpha\mathbf{y} - \alpha\mathbf{c}_t\|^2 \alpha^d d\mathbf{y} \tag{1.7}$$

The value of α as a function of N can be deduced by considering the total volume that the regions cover. The volume is

$$\mathrm{vol}(\Delta) = N\,\mathrm{vol}(\alpha\Omega_t) = N\alpha^d\,\mathrm{vol}(\Omega_t) \tag{1.8}$$

from which follows that

$$\alpha = \left(\frac{\mathrm{vol}(\Delta)}{N\,\mathrm{vol}(\Omega_t)}\right)^{1/d} \tag{1.9}$$

This value inserted into (1.7) yields

$$D \approx \left(\frac{\mathrm{vol}(\Delta)}{N}\right)^{2/d} \left(\frac{1}{\mathrm{vol}(\Omega_t)}\right)^{1+2/d} \int_{\Omega_t} \|\mathbf{y} - \mathbf{c}_t\|^2 d\mathbf{y}$$
$$= d\,\mathrm{vol}(\Delta)^{2/d} 2^{-2R} G \tag{1.10}$$

where

$$G = \frac{1}{d\,\mathrm{vol}(\Omega_t)^{1+2/d}} \int_{\Omega_t} \|\mathbf{y} - \mathbf{c}_t\|^2 d\mathbf{y} \tag{1.11}$$

is the *normalized second moment* of the typical body Ω_t. This measure is independent of the rate and the source shape. It is also dimensionless and thus insensitive to scaling. Hence, congruent bodies have the same G. The normalization with respect to d is to make easier the comparison between quantizers of different dimensions. This report is devoted to the search for structures with a low value of G.

The distortion expression (1.10) can be used to estimate the performance of a well-optimized high-rate vector quantizer for a uniform source. Conversely, it can also be used as a tool in the design of such quantizers. The method is to find a d-dimensional body Ω_t with a low G. Every body is not admissible; only bodies that can form a *tessellation*. A tessellation is a partition of $\mathbb{R}^d$ into regions, such that any pair of regions can be transformed into each other through rotation, reflection, and translation.[3] When a tessellation is found that consists of bodies with a low G, the codebook is formed as the intersection of the centroids and Δ. The desired rate determines the scaling of the tessellation. The structure is called a *tessellating quantizer* [8]. In previous studies of

[3] The body with the lowest G is the d-dimensional sphere, but it is not admissible as Ω_t, because it cannot form a tessellation (for $d \geq 2$).

tessellating quantizers, most attention has been devoted to *lattice quantizers*, which constitute an important subset of all tessellation quantizers. Lattices are defined in the next section.

1.3 Lattices

A popular special case of a structure, whose Voronoi regions form a tessellation,[4] is a *lattice*. The following brief summary of lattice theory is intended to be a sufficient background for the quantization problem investigated in this report. For a more extensive treatment, the interested reader is referred to the book by Conway and Sloane [2], which has more or less become the standard textbook on lattice theory.

A lattice is an infinite set of vectors, defined through d linearly independent *basis vectors* $\mathbf{b}_1, \mathbf{b}_2, \cdots \mathbf{b}_d$. The lattice consists of all linear combinations of the basis vectors, with integer coefficients. The matrix whose rows are the basis vectors is called the *generator matrix* of the lattice,

$$\mathbf{B} = \left[\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_d\right]^T \tag{1.12}$$

Formally, we can write the lattice Λ as

$$\Lambda = \left\{\mathbf{x} \in \mathbb{R}^d : \ \left(\mathbf{B}^{-1}\right)^T \mathbf{x} \in \mathbb{Z}^d\right\} \tag{1.13}$$

Hence, any lattice point[5] can be uniquely written as $\mathbf{B}^T\mathbf{u}$, where $\mathbf{u} \in \mathbb{Z}^d$.

Figure 1.1 is an example of a lattice. It is the well-known hexagonal lattice, also called A_2, and can be defined through, e.g., the generator matrix

$$\begin{bmatrix} 2 & 0 \\ 1 & \sqrt{3} \end{bmatrix} \tag{1.14}$$

A_2 is the 2-dimensional case of the lattice A_d, which is defined, along with some other common lattices, in the appendix.

In the design and analysis of lattices, it is often convenient to employ d basis vectors having more than d coordinates. However, throughout this report, $\mathbf{B}$ denotes a square generator matrix. This notation does not restrict generality, since d vectors cannot span more than d dimensions. Hence, for every nonsquare generator matrix $\mathbf{B}'$, there exists a square matrix $\mathbf{B}$ describing an *equivalent* lattice. (More on equivalent lattices below.) Practically, such a $\mathbf{B}$ can be found through, e.g., QR factorization of $\left(\mathbf{B}'\right)^T$ or Cholesky decomposition of $\mathbf{B}'\left(\mathbf{B}'\right)^T$. Some of the following theory draws advantage of $\mathbf{B}$ being

---

[4] When there is no risk of confusion, we will also use "tessellation" to denote an infinite point set whose Voronoi regions form a tessellation. Thus, a lattice is a tessellation.

[5] We use "lattice point" and "lattice vector" interchangeably.

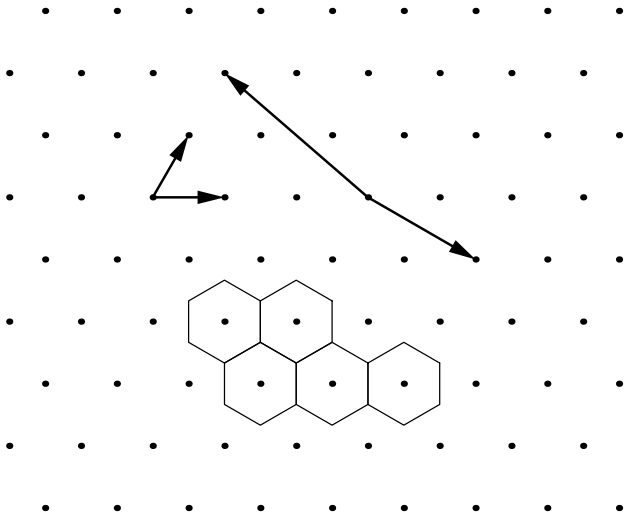

Figure 1.1. Two possible bases for the hexagonal lattice. Some of the Voronoi regions are shown.

square, thus simplifying the notation. For example, both the inverse and the determinant of $\mathbf{B}$ have important interpretations.

Lattice points are evenly distributed in space—there is not a region where the lattice is denser than somewhere else. It is because of this uniformity that lattices are suitable for the quantization of uniform sources. Gersho pointed out, "if you sit on one lattice point and view the surrounding set of lattice points, you will see the identical environment regardless of which point you are sitting on" [9]. Consequently, the Voronoi regions form a tessellation, as mentioned above in section 1.2. Indeed, the Voronoi regions are pure translations of each other, without needing any rotation or reflection. (See figure 1.1 for an example.) This is a chief characteristic of all lattices.

The all-zero vector $\mathbf{0}$ belongs to all lattices. This follows trivially from the definition (1.13). The Voronoi region around $\mathbf{0}$,

$$\Omega = \left\{ \mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|^2 \le \|\mathbf{x} - \mathbf{c}\|^2 \text{ for all } \mathbf{c} \in \Lambda \right\} \tag{1.15}$$

is commonly called *the* Voronoi region of the lattice Λ. It is the standard choice of a typical body (see section 1.2) in the computation of lattice parameters. The volume of Ω is $V = \mathrm{vol}(\Omega) = |\det \mathbf{B}|$.[6] The normalized second moment (1.11) is

$$G = \frac{1}{dV^{1+2/d}} \int_{\Omega} \|\mathbf{x}\|^2 \, d\mathbf{x} \tag{1.16}$$

A complication in the analysis of lattices is that equivalent lattices can be specified through seemingly different generator matrices. Two lattices are considered *equivalent* if

[6] The volume is more commonly given in the form $(\det(\mathbf{B}\mathbf{B}^T))^{1/2}$, which allows for nonsquare generator matrices.

their Voronoi regions (1.15) are congruent. In this case, the two lattices have the same G, and most other lattice parameters agree, too. For example, the generator matrices

$$\begin{bmatrix} -2 & 0 \\ 3 & 1/\sqrt{3} \end{bmatrix}, \begin{bmatrix} \sqrt{3}+1 & \sqrt{3}-1 \\ \sqrt{3}-1 & \sqrt{3}+1 \end{bmatrix}, \text{ and } \begin{bmatrix} 2 & 2-\pi\sqrt{2} \\ 1-1/\sqrt{3}+\pi/\sqrt{6} & 1+1/\sqrt{3}-\pi/\sqrt{2} \end{bmatrix}$$

all specify the A_2 lattice, so these lattices are equivalent to the one given by (1.14).

A lattice can be transformed by scaling, rotation, and reflection, without changing the shape of the Voronoi region.[7] In addition, basis vectors can be selected in many ways within the point set Λ, as illustrated in figure 1.1. It can be shown that the lattices generated by $\mathbf{B}_1$ and $\mathbf{B}_2$ are equivalent if and only if there exist matrices $\mathbf{W}$ and $\mathbf{Q}$ such that

$$\mathbf{B}_2 = \left(\frac{V_2}{V_1}\right)^{1/d} \mathbf{W}\mathbf{B}_1\mathbf{Q} \tag{1.17}$$

where all elements of $\mathbf{W}$ are integers, $\mathbf{W}$ has determinant ±1, and $\mathbf{Q}$ is orthonormal. The coefficient $(V_2/V_1)^{1/d}$ takes care of scaling, $\mathbf{W}$ of basis change, and $\mathbf{Q}$ of rotation and/or reflection. Unfortunately, there has, to our best knowledge, not been published any general algorithm to determine whether two given generator matrices specify equivalent lattices. Of course, if either $\mathbf{W}$ or $\mathbf{Q}$ is known, the other one is obtained by matrix inversion, but to determine both of them simultaneously is still an open problem. It has been suggested to employ a *canonical form* for lattices to solve the problem: if $\mathbf{B}_1$ and $\mathbf{B}_2$ have the same canonical form, they are equivalent; otherwise not. Unfortunately, the algorithms that have been proposed to transform a generator matrix into a canonical form (see, e.g., [10, pp. 65–67] and [11, pp. 184–201]) consider basis changes only, not rotation. If $\mathbf{B}_1$ and $\mathbf{B}_2$ are rotated versions of the same lattice, the canonical forms obtained by such an algorithm will differ. Hence, the problem of identifying equivalent lattices remains.

Finally, for every lattice there is a *dual*. The dual of Λ is another lattice, whose generator matrix is $\left(\mathbf{B}^{-1}\right)^T$. The dual is denoted Λ^*. It has the same degree of symmetry as Λ, but the lattice parameters, such as the normalized second moment G, are normally different.

---

[7] Translation also preserves the Voronoi region, but a translated lattice is normally not a lattice (1.3). It is still, of course, a tessellation.

1.4 Quantizer Design for Nonuniform Sources

We now return to vector quantization. So far, the discussion has been focused upon uniform sources, where lattices are immediately applicable as quantizer structures. While under some circumstances, for example, image data can be modeled as a uniform source [12, p. 33], most applications display different probability density functions. However, lattices have found their use in vector quantization for nonuniform sources, too.[8] In this section, we will review some approaches that have been proposed in the past.

One possibility is, of course, to approximate the probability density function of the source with a uniform function, and design a *lattice quantizer* (section 1.2) accordingly. Much attention has been devoted to the problem of optimizing the size and shape of the uniform function for a given source density; in other words, the problem of scaling and truncation of the lattice. This problem is discussed in [1] and several of its references. The gain in memory and encoding time, compared with a source-optimized codebook, is significant. The price paid is a performance degradation, the severity of which depends on the rate, the dimension, and the probability density of the source. The general trend is that the degradation increases with higher rate and lower dimension, as illustrated for a Gaussian source in figure 3.4 of [1].

For high-dimensional sources, a low-rate lattice quantizer is known to have close to optimal performance. This is because of the *asymptotical equipartition property*, according to which a large class of high-dimensional probability density functions can be well approximated with uniform densities [13, pp. 73, 285], [14]. For example, data drawn from an uncorrelated Gaussian density tend to be uniformly distributed in a thin spherical shell, if the dimension is high [15], whereas the multidimensional Laplacian density can be approximated by a uniform density on the surface of a “pyramid” (hyperoctahedron) [16]. The tendency towards uniform distributions has been successfully employed in several applications. Competitive lattice quantizers have been designed for use in CELP [17] and transform coded [18] speech coding systems. In image coding, Jeong and Gibson have achieved good performance through lattice quantized DCT coefficients [19]. For high rates and low dimensions, on the other hand, the performance degradation of lattice quantizers compared with source-optimized vector quantizers can be quite severe [20, 1].

---

[8] In fact, all applications of lattices that are mentioned in this section are directly generalizable to other types of tessellations as well. We retain the lattice terminology because it is the framework in which most of the research was originally published.

To avoid performance degradation due to nonuniform sources, the quantizer should be matched to the specific source density, but still there exist promising alternatives to the training of an unconstrained codebook. The basic idea is to maintain the local lattice-similarity while making the global structure matched to the source.

One quantizer structure with this aim is the *piecewise uniform* quantizer introduced by Kuhlmann and Bucklew [21, 20]. It is a generalization of the lattice quantizer, where a given (nonuniform) probability density function is approximated with a staircase function. In each region where the density approximation is constant, the codebook is populated by a suitably scaled lattice. Similar structures are obtained by designing two-stage quantizers where the second stage is a lattice [22, 23, 24].

A more general method to improve the performance of a lattice quantizer for nonuniform source densities is to apply a nonlinear transform function to each input vector before quantization, and the inverse function to quantized data. This approach is called *companding* and it is used in many scalar applications. It was suggested for use in vector quantization by Gersho [7], and Bucklew characterized its high-rate performance [25, 26]. Antonini et al. applied companding and lattice vector quantization to wavelet coefficients for image data [27]. The piecewise uniform quantizer is a special case of a companding lattice quantizer, where the transform function is piecewise linear.

An alternative method to modify a lattice quantizer to match a nonuniform source is presented in the sequel of this report, [1], where the advantages of a lattice structure are incorporated into a design algorithm for source-optimized vector quantizers.

In section 1.1 it was assumed that each codevector was encoded with exactly $\log_2 N$ bits. If the codevectors have unequal *a priori* probabilities, the average rate can be reduced by applying an entropy code to the quantizer output. It has been shown that if an entropy code is employed, the *optimal high-rate vector quantizer should have a uniform distribution of codevectors* [7] [28, p. 131] [29, p. 471]. Hence, if Gersho's conjecture (see section 1.2) is true, then a tessellating quantizer is asymptotically optimal when the rate tends to infinity. The optimality does not require the source density to be uniform or even smooth, only that the differential entropy is finite, as proved by Linder and Zeger [8]. It is worth mentioning that a tessellating quantizer with entropy coding performs closer to the rate-distortion bound than the optimal fixed-rate vector quantizer. The argument behind this statement is the following: The optimal fixed-rate vector quantizer is inferior to (has higher average rate than) the same quantizer with an entropy code. And a codebook with nonuniform point density is inferior to a uniform codebook,

as long as entropy coding is being applied. Applications of entropy coded lattice quantization are presented in, e.g., [14, 30, 31].

For high rates, the performance of an entropy-coded tessellating quantizer is proportional to the normalized second moment G of the tessellation. This was shown in [8], through the high-rate approximation

$$D \approx d\,G\,2^{2(h-H)/d} \tag{1.18}$$

in which $h = \int f_{\mathbf{x}}(\mathbf{x}) \log_2 f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}$ is the differential entropy of the source and H is the output entropy of the quantizer, $H = -\sum p_i \log_2 p_i$, where p_i is the probability of the event $Q(\mathbf{x}) = \mathbf{c}_i$. The approximation is asymptotically exact, in the sense that the relative error tends to zero as H approaches infinity. The factor G in (1.18) shows the importance of tessellations with a low G; the gain obtained by improving a tessellation can be expected to propagate directly into the distortion of an entropy-coded quantizer built upon the tessellation. A distortion proportional to G is also a feature of, e.g., high-rate uniform quantizers (1.10) and lattice quantization of Gaussian sources [1, sec. 3.2].

Hence, we turn our attention towards the minimization of G.

II. NUMERICAL OPTIMIZATION OF LATTICES

The history of lattice design is closely interlinked with group theory and error-correcting codes. Almost all lattice design methods that have been proposed arise from the algebraic approach. We now study an alternative method. The basic idea, which was first suggested in [32], is to use an iterative algorithm to adjust a given lattice.[9] We have developed an algorithm that minimizes the normalized second moment by a gradient search procedure. The algorithm is related to algorithms for vector quantizer training, but it operates on a generator matrix instead of individual codevectors. In section 2.1, we regard lattice design for quantization as an optimization problem and adopt a suitable set of variables. The new algorithm is presented in detail in section 2.2, together with the theoretical background. Section 2.3 discusses in general terms what kind of results is expected from the algorithm, and how these results can be interpreted.

2.1 The Optimization Problem

The problem of finding a good lattice for high-rate uniform quantization can be stated as a multivariate minimization problem

$$\min_{\mathbf{B}\in\mathbb{R}^{d\times d}} G \tag{2.1}$$

where G is the normalized second moment (1.16)

$$G = \frac{1}{dV^{1+2/d}} \int_{\Omega} \|\mathbf{e}\|^2 \, d\mathbf{e} \tag{2.2}$$

The problem contains d^2 unknowns, or degrees of freedom, namely, the d^2 elements of the generator matrix, which specify the lattice through the construction (1.13).

To simplify the problem, we recall the concept of equivalent lattices, see section 1.3, especially (1.17). There are many ways to change a generator matrix into one that spans an equivalent lattice, and such changes do not affect the normalized second moment G at all. On the contrary, an iterative optimization algorithm should concentrate on changes that has a potential of improving the lattice. Why attempt a 100-variate optimization problem when 50 variables suffice? If we fix the rotation of the lattice, and the scaling, almost half of the d^2 variables in the generator matrix can be removed from the minimization.

[9] The suggestion in [32] is to use "random walk": Make a random change to one random element of the generator matrix, then encode a training database to evaluate the change. This method is slower than ours by at least a factor M, the database size.

Specifically, the rotation is in (1.17) controlled by an orthonormal $d \times d$ matrix, and the set of all such matrices spans a parameter space of $d(d-1)/2$ dimensions. This number of variables disappear from the minimization problem when the rotation is fixed; one more variable disappears when the scale factor is fixed.[10] Hence, of the d^2 degrees of freedom that a general generator matrix $\mathbf{B}$ possesses, $d(d-1)/2+1$ are irrelevant in the optimization of G, whereas $d^2 - d(d-1)/2 - 1 = (d+2)(d-1)/2$ are important. The irrelevant degrees of freedom can be removed from the generator matrix in several ways; we employ the following straightforward method.

With $b_{i,j}$ denoting row i and column j of $\mathbf{B}$, we impose the constraints

$$\begin{cases} b_{i,j} = 0 & \text{if } i < j \\ \det \mathbf{B} = V \end{cases} \tag{2.3}$$

where V is a constant, upon generator matrices for use with the training algorithm. The volume V can be arbitrarily chosen; we use a volume of 1. Geometrically, these rules lock the first basis vector, $\mathbf{b}_1$, along the first axis of the coordinate system, $\mathbf{b}_2$ in the plane spanned by the first two axes, etc. In addition, the volume of Voronoi regions, V, is locked to unity. These rules are maintained through the vector

$$\mathbf{f} = \left(f_1, \cdots, f_{(d+2)(d-1)/2} \right)^T \tag{2.4}$$

where

$$f_{i(i-1)/2+j} = b_{i,j} \quad \text{for } j = 1, \cdots d-1 \text{ and } i = j, \cdots, d \tag{2.5}$$

are the $(d+2)(d-1)/2$ "free" variables in the optimization. The vector $\mathbf{f}$ determines the generator matrix

$$\mathbf{B} = \begin{bmatrix} b_{1,1} & 0 & \cdots & 0 & 0 \\ b_{2,1} & b_{2,2} & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ b_{d-1,1} & b_{d-1,2} & \cdots & b_{d-1,d-1} & 0 \\ b_{d,1} & b_{d,2} & \cdots & b_{d,d-1} & \prod_{k=1}^{d-1} b_{k,k}^{-1} \end{bmatrix} \tag{2.6}$$

We emphasize that *any* lattice can be presented on this form.[11]

The constraint thus imposed on the form of the generator matrix $\mathbf{B}$ actually serves two purposes in the simplification of the numerical optimization problem (2.1). The

---

[10] The change of basis vectors, as denoted by the matrix $\mathbf{W}$ in (1.17), does not contribute any degrees of freedom in the sense discussed here, since the elements of $\mathbf{W}$ are subject to an integer constraint.

[11] Formally, for any given lattice, there exists an *equivalent* lattice with a generator matrix (2.6).

number of variables are reduced, as discussed above, and it is also worth observing that the objective function itself (2.2) gets a simpler form, namely,

$$G = \frac{1}{d} \int_{\Omega} \|\mathbf{e}\|^2 \, d\mathbf{e} \tag{2.7}$$

because of the volume normalization. This expression contains no determinant, which will lead to a very simple form for the gradient derived in the next section.

2.2 The Lattice Training Algorithm

The training of a lattice basis is in many ways similar to training of an unstructured VQ for a uniform probability density function, but there are also important differences. For instance, the centroid condition for local optimality of a VQ [1, sec. 2.2] is of no help, since the codevectors of *any* lattice are the centroids of their Voronoi regions. This means that we cannot use the generalized Lloyd algorithm in its common form, or, for that matter, no other algorithm that relies on the centroid condition, for the training. Instead, we propose a stochastic gradient algorithm.

The algorithm takes advantage of the problem formulation in the previous section. The strategy is to iterate the vector $\mathbf{f}$ (2.4) in order to decrease G (2.7). Random training vectors are generated with a uniform distribution inside the Voronoi region Ω. For each training vector, the squared distance to the origin is computed, and also the gradient of the squared distance with respect to $\mathbf{f}$. Then $\mathbf{f}$, and thus the generator matrix, is adjusted a small step in the direction of the negative gradient. The adjustment can be made for each individual training vector or for blocks of vectors.

The first question is how to generate the training vectors. Conway and Sloane give an elegant method to generate uniform data within a Voronoi region [33]. First, d independent random numbers are obtained, uniformly distributed between 0 and 1. They constitute a random vector within the d-dimensional unit cube. Calling this vector $\mathbf{z}$, another vector $\mathbf{x} = \mathbf{B}^T\mathbf{z}$ is created. Next, a search algorithm is applied to find the closest codevector to $\mathbf{x}$ in the lattice, denoted $\mathbf{c}^* = \mathbf{B}^T\mathbf{u}$. Finally, the difference vector $\mathbf{e} = \mathbf{x} - \mathbf{c}^*$ is uniformly distributed over the Voronoi region Ω.

To steer the training, we need the gradient of the integrand in (2.7). Hence, we differentiate $\|\mathbf{e}\|^2$ with respect to each component of $\mathbf{f}$:

$$\frac{\partial \|\mathbf{e}\|^2}{\partial b_{i,j}} = \frac{\partial}{\partial b_{i,j}} \sum_{k=1}^{d} e_k^2 = 2 \sum_{k=1}^{d} e_k \cdot \frac{\partial e_k}{\partial b_{i,j}} \tag{2.8}$$

where e_k denotes component k of $\mathbf{e}$. To find the partial derivatives $\partial e_k / \partial b_{i,j}$, $\mathbf{e}$ is first written as a function of $\mathbf{f}$. Defining $\mathbf{y} = (y_1, \cdots, y_d) = \mathbf{z} - \mathbf{u}$, we obtain

$$\mathbf{e} = \mathbf{x} - \mathbf{c}^* = \mathbf{B}^T\mathbf{z} - \mathbf{B}^T\mathbf{u} = \mathbf{B}^T\mathbf{y}, \tag{2.9}$$

or, componentwise,

$$e_k = \sum_{l=1}^{d} b_{l,k}\, y_l \tag{2.10}$$

This gives $\mathbf{e}$ as a function of $\mathbf{B}$, which in turn is a function of $\mathbf{f}$. To continue, we employ (2.6):

$$\begin{aligned} e_k &= \sum_{l=k}^{d} b_{l,k}\, y_l \\ &= \begin{cases} \displaystyle\sum_{l=k}^{d} b_{l,k}\, y_l & \text{if } k < d \\ \displaystyle y_d \prod_{l=1}^{d-1} b_{l,l}^{-1} & \text{if } k = d \end{cases} \end{aligned} \tag{2.11}$$

which is a function of $\mathbf{f}$ only. The derivative is

$$\frac{\partial e_k}{\partial b_{i,j}} = \begin{cases} y_i & \text{if } j = k < d \\ \displaystyle -\frac{y_d}{b_{i,i}} \prod_{l=1}^{d-1} b_{l,l}^{-1} & \text{if } i = j \text{ and } k = d \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

Inserted into (2.8), finally, this yields

$$\frac{\partial \|\mathbf{e}\|^2}{\partial b_{i,j}} = \begin{cases} 2e_j y_i & \text{if } i \neq j \\ \displaystyle 2e_i y_i - \frac{2e_d y_d}{b_{i,i}} \prod_{l=1}^{d-1} b_{l,l}^{-1} & \text{if } i = j \end{cases} \tag{2.13}$$

for all components $b_{i,j}$ of $\mathbf{f}$. Equation (2.13) gives, componentwise, the gradient of $\|\mathbf{e}\|^2$ with respect to $\mathbf{f}$. According to the steepest descent rule, $\mathbf{f}$ should be updated in the direction of the negative gradient.

It is well known that many small changes to a matrix may eventually make it ill-conditioned. This means that an iterative algorithm for lattice design may grow long and almost parallel basis vectors, which may slow the algorithm down and in severe cases cause numerical problems. Fortunately, there is a way to counteract this problem. Again we rely on the theory of equivalent lattices. Through a basis change, a given generator matrix can be replaced by one in which the basis vectors are short and reasonably orthogonal to each other. This process, which is called *reduction*, should be repeated regularly during lattice training. Several reduction algorithms have been proposed, of which the one by Lenstra *et al.* [34] is probably the most popular. The reduction normally destroys the triangular structure of the generator matrix, so reduction is immediately succeeded by rotation in our algorithm. It is worth stressing that neither reduction nor

rotation changes the lattice (except into an equivalent one); it is the *representation* of the lattice that is changed.

Some details must be decided in order to complete the training algorithm. The choices include block- or sample-based training, step size values, etc. In table 2.1, we have formulated one suggestion, a sample-based algorithm with linearly decreasing step size. Note that the diagonal elements of the generator matrix are updated differently from the other elements. The upper triangular part is not updated, and the last diagonal element, $b_{d,d}$, is computed from the other diagonal components as in (2.6). In the execution of the training algorithm, we employ $b_{d,d}$ to simplify the expressions (2.11) and (2.13). Still, $b_{d,d}$ should be regarded purely as a function of other matrix elements, and $b_{d,d}$ does not

TABLE 2.1. THE LATTICE TRAINING ALGORITHM.

Step 1: Initialize the generator matrix $\mathbf{B}$ with values that satisfy (2.3). Set the number of iterations M, the start step size ε_0, and the reduction interval M_r to suitable values. Set $m = 1$.

Step 2: Compute a new training vector as $\mathbf{x} = \mathbf{B}^T\mathbf{z}$, where each component of $\mathbf{z}$ is uniformly distributed in the interval $(0,1)$.

Step 3: Find the lattice vector $\mathbf{c}^* = \mathbf{B}^T\mathbf{u}$ that is closest to the training vector $\mathbf{x}$. Set $\mathbf{y} = \mathbf{z} - \mathbf{u}$ and $\mathbf{e} = \mathbf{B}^T\mathbf{y}$.

Step 4: Update all components of $\mathbf{f}$ as

$$b_{i,j} := b_{i,j} - \varepsilon_m \cdot \frac{\partial \|\mathbf{e}\|^2}{\partial b_{i,j}}$$

where

$$\frac{\partial \|\mathbf{e}\|^2}{\partial b_{i,j}} = \begin{cases} 2e_j y_i & \text{if } i \neq j \\ 2e_i y_i - 2e_d y_d \dfrac{b_{d,d}}{b_{i,i}} & \text{if } i = j \end{cases}$$

and ε_m is a linearly decreasing step size parameter, $\varepsilon_m = \varepsilon_0(1 - m/M)$. The last diagonal element, $b_{d,d}$, is computed by the expression

$$b_{d,d} = \left(\prod_{k=1}^{d-1} b_{k,k} \right)^{-1}$$

in order to maintain (2.3).

Step 5: If m is divisible by M_r, then perform a reduction on $\mathbf{B}$ and subsequently rotate $\mathbf{B}$ into lower triangular form.

Step 6: If the desired number of iterations has been performed, $m = M$, then exit. Otherwise, set $m := m + 1$ and continue from step 2.

enter the vector of optimization variables, $\mathbf{f}$.

Any lattice can serve as initial value for $\mathbf{B}$. We recommend the cubic lattice (see appendix), possibly with a small random disturbance. This lattice is neutral, in the sense that it is not close to any local optimum. Almost any adjustment will reduce the normalized second moment. Metaphorically speaking, the cubic lattice lies on top of the hill. Of course, a better lattice can be used as initialization, if the user wishes to examine this specific lattice, for instance to determine whether it is a local optimum, but this is not a good strategy in the search for a global optimum, especially if the chosen initial lattice is already good. From a point below the hill, you will not see the deep valleys on the other side. The strength with our training algorithm is that it may point at previously unknown lattices.

Three training parameters, ε_0, M, and M_{r}, must be specified in advance for the algorithm. Our standard choice, empirically found, is $\varepsilon_0 = 10^{-3}$, $M = 10^7$, and $M_{\mathrm{r}} = 10^4$. Small changes in the values, tailored to the intended experiment, may yield slightly improved performance (in terms of speed and/or quality), but the algorithm is not too sensitive to these parameter values. In fact, $M_{\mathrm{r}} = \infty$ works well in dimensions up to about 10, which means that low-dimensional lattices can be designed by a simplified version of the algorithm, in which step 5 is omitted.

The computational complexity in the training algorithm is, for high d, dominated by the so-called *closest point problem,* the search for the closest lattice point of the training data, in step 3. Algorithms have been developed by Kannan [35] and Agrell and Eriksson [36]. It has been theoretically proved that the problem is NP-hard, see, e.g., [37], but the complexity is nevertheless not overwhelming. To indicate the order of magnitude, we mention that with one implementation, the average time to find the closest point in a 24-dimensional lattice is 37 milliseconds.

The lattice training algorithm is easily modified to solve other problems that can be formulated in a similar framework. For example, if we search for a lattice under the constraint of a specific structure, all that is needed is the identification of a vector of free optimization variables $\mathbf{f}$ and the gradient of $\|\mathbf{e}\|^2$ with respect this vector. One application of this idea is reported in section 3.2, where the generator matrix (3.1) was refined by such a constrained lattice training algorithm.

2.3 Identification of Lattices

In applications of numerical optimization, exact solutions cannot be expected. The presented training algorithm for lattices is of course no exception: the results are

approximations of true optima, local or global. This section is a discussion of the interpretation of approximate results. An example concludes the section.

The purpose of the lattice training algorithm is to find lattices with the lowest possible value of the normalized second moment G, for a given dimension d. The computation of G for a general lattice involves the determination of every vertex, edge, 2-dimensional face, etc., of the Voronoi region [38]. The complexity of this computation grows dramatically with the dimension, so it is practically feasible only for lattices of moderate dimension. As a less complex alternative, G can be estimated through Monte Carlo integration of (2.7). This method, proposed by Conway and Sloane in [33], is the one we use in this report. We also follow their nomenclature in presenting estimates of G on the form $\hat{G} \pm 2\hat{\sigma}$, where $\hat{\sigma}$ is an estimate of the standard deviation of $\hat{G}$.

When the lattice training algorithm exits, it has converged to the vicinity of a local minimum. Thus, the training algorithm does not directly point out exact local minima $\mathbf{f}^*$; it terminates anywhere inside a small region around $\mathbf{f}^*$. All $\mathbf{f}$'s inside the region represent the same minimum, and to find this (exact) minimum, we need some kind of "round-off" process. The round-off, which takes place when the training is complete, is guided by the following rule.

Postulate 1: If two lattices represent the same local minimum, the one with most symmetry is the more accurate representative.

The postulate is empirically motivated. Nature favors symmetry. In the past, every lattice that has shown good quantization performance has also possessed a high degree of symmetry, while on the other hand, the (unrounded) lattices generated by the training algorithm have minimum symmetry (that is, reflection in the origin and nothing else). There is a practical reason to encourage symmetry as well: search time. For many lattices, the symmetrical structure has been exploited in the development of very efficient search algorithms [39].

In the comparison of lattices, it is important to remember the possibilities of basis change and rotation. As discussed in section 1.3, equivalent lattices can have generator matrices that look quite different from each other. This effect is due to change of basis and rotation ($\mathbf{W}$ and $\mathbf{Q}$ in (1.17)). In the identification of lattices, these two tools are valuable, as the following example will demonstrate. The example also illustrates how we employ postulate 1.

Example 1: For $d = 5$, one run of the lattice training algorithm gave the following generator matrix:

$$\mathbf{B}_1 = \begin{bmatrix} 1.285 & 0.000 & 0.000 & 0.000 & 0.000 \\ -0.518 & 1.025 & 0.000 & 0.000 & 0.000 \\ -0.255 & 0.517 & 1.149 & 0.000 & 0.000 \\ -0.514 & -0.261 & -0.579 & 0.811 & 0.000 \\ 0.513 & 0.263 & -0.572 & -0.003 & 0.815 \end{bmatrix}$$

The normalized second moment G was estimated to 0.075624 ± 0.000010. Direct inspection of the generator matrix does not immediately suggest any symmetries. To see the structure of this lattice better, we create another generator matrix through (1.17) with a basis change given by

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

a rotation given by

$$\mathbf{Q} = \begin{bmatrix} 0.449011 & 0.448323 & 0.447087 & 0.446851 & 0.444785 \\ -0.893513 & 0.224658 & 0.226453 & 0.227533 & 0.219344 \\ 0.001054 & 0.496921 & 0.500431 & -0.500429 & -0.502203 \\ -0.001532 & 0.708240 & -0.705966 & -0.002575 & -0.000121 \\ 0.004514 & -0.001350 & -0.003817 & 0.705774 & -0.708411 \end{bmatrix}$$

and a scaling of $V_2/V_1 = 0.5$. The new generator matrix,

$$\mathbf{B}_2 = \left(\frac{V_2}{V_1}\right)^{1/d} \mathbf{W}\mathbf{B}_1\mathbf{Q} = \begin{bmatrix} 1.000 & 0.002 & -0.000 & -0.001 & 0.005 \\ 0.002 & 0.999 & 0.002 & 0.001 & -0.001 \\ -0.000 & 0.002 & 1.002 & 0.001 & -0.004 \\ -0.001 & 0.001 & 0.001 & 1.002 & -0.003 \\ 0.502 & 0.501 & 0.500 & 0.500 & 0.498 \end{bmatrix}$$

specifies a lattice that is equivalent to $\mathbf{B}_1$. The structure underlying this matrix is clearly visible, and postulate 1 allows us to create a more accurate representation of the found minimum by rounding off the elements:

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1/2 & 1/2 & 1/2 & 1/2 & 1/2 \end{bmatrix}$$

This generator matrix is well known. It produces the very symmetrical D_5^* lattice (A.3), which is the best 5-dimensional lattice currently known, in terms of low normalized second moment, G. Its G value is known exactly [40]; it is $2^{-48/5} \cdot 2641/45 \approx 0.075625$, which falls well within the interval estimated for $\mathbf{B}_1$. ❑

III. EXPERIMENTS

The work presented in this report was inspired by the need for better lattices for quantization. As discussed in chapter I, the performance of a lattice quantizer is, under some circumstances, characterized by the normalized second moment of the lattice, G. Figure 3.1 summarizes the best classical lattices [40, 41, 42, 43] along with Conway and Sloane's conjectured lower bound [41]. With "classical" we mean lattices for which the normalized second moment has been reported previously. The figure hints a potential for improvement, especially in 9 and 10 dimensions. 10-dimensional quantization has received a lot of attention in speech coding [44, 45], where suboptimal structures such as split VQ and multistage VQ have been much employed. A good 10-dimensional lattice might provide an attractive alternative in this application.

In section 3.1, the results obtained through lattice training are presented. When the dimension d is 9 or 10, we find lattices with normalized second moments considerably lower than the values previously known for these dimensions. If the new lattices are indeed optimal, they disprove a famous conjecture by Conway and Sloane. The structure of the new lattices draws our attention to a class of nonlattice tessellations, which is

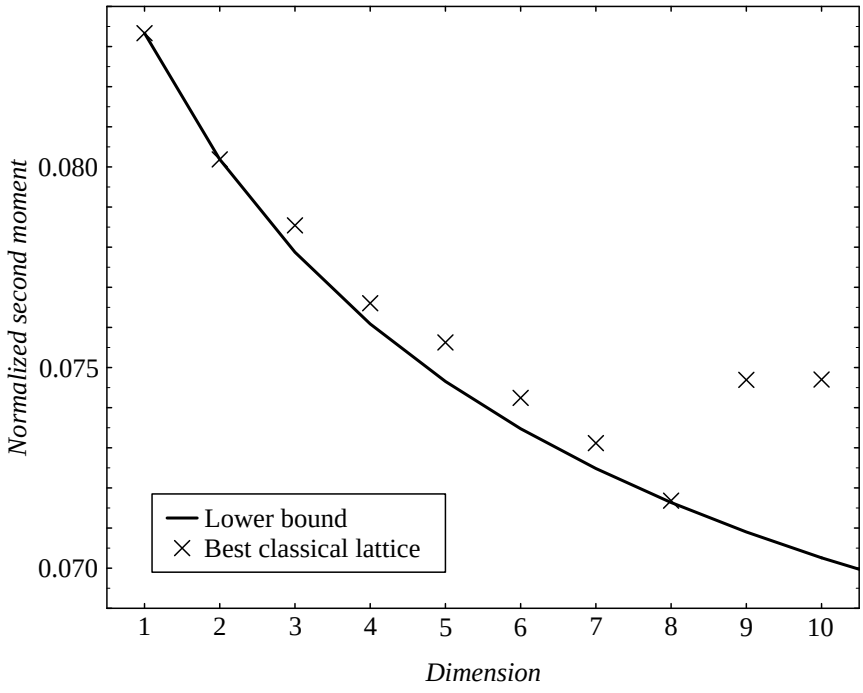

Figure 3.1. The lowest normalized second moments previously known, in dimensions 1–10. This diagram is supplemented with our results in subsequent figures.

examined in section 3.2. We report tessellations that are better than all previously studied tessellations, including lattices, in 7 and 9 dimensions. We tried to focus the chapter on the results of the training algorithm, without expanding the work into an essay on lattice theory. Hence, some theoretical background and definitions are left to references.

3.1 The Best Lattices Found

There is no theoretical limitation on the number of dimensions that the lattice training algorithm can handle, only a practical one. The algorithm, in its present form, typically requires 2 hours in 3 dimensions, 4 hours in 10, and 25 hours in 20. With other training parameters, time can, of course, be bought to the price of accuracy. To evaluate the algorithm, especially to assess its ability to converge into local minima with low normalized second moments G, we considered it important to run the algorithm several times in each dimension, and identify each output lattice through the methodology of section 2.3, in a manner similar to example 1. This required some manual work on each lattice, the amount of which ranged from seconds to hours for the identification of a single lattice.

The considerations above led to the following experiment setup. The training algorithm was run 10 times in each dimension from 2 to 10, and the trained lattices were identified. Most of this section is devoted to results from this experiment. To study some further features of the algorithm, we also designed 100 3-dimensional lattices and one 20-dimensional one.

In table 3.1, the 10 lattices obtained in dimensions 2–10 are listed, grouped according to which local minimum they represent. Normalized second moments G were estimated for one (randomly selected) member of each group. Most of the groups represent one of the "classical" lattices, among which the most notable are $\mathbb{Z}^d$, A_d, and A_d^* for $d \geq 1$; D_d and D_d^* for $d \geq 3$; and E_d and E_d^* for $6 \leq d \leq 8$. They are all defined in the appendix and their properties (normalized second moment, etc.) can be found in [2, chs. 4 and 21]. A few groups do not represent any known and named lattice; these lattices are characterized below. For comparison, the previously best known G values[12] and Conway and Sloane's lower bound[13] are included in the table.

The 10 runs for each dimension turned out to converge into just a few different local minima; more than three local minima were not found for any dimension. Of these minima, one, called the *principal minimum* for a given d, always got significantly more

[12] Among the "best known" lattices, only the ones in dimensions 1–3 have been proven optimal.

[13] The values were computed using a series expansion of the recursive integral equation in [41].

TABLE 3.1. THE LATTICES OBTAINED BY THE TRAINING ALGORITHM, GROUPED ACCORDING TO LOCAL MINIMA.

d	Trained lattices				Previously best known		Lower bound
	Number of local minima	Hits in each minimum	G	Name of minimum	G	Name	G
2	1	10	0.080180 ± 0.000010	A_2	0.080188	A_2	0.080188
3	1	10	0.078540 ± 0.000010	A_3^*	0.078543	A_3^*	0.077875
4	2	9 1	0.076602 ± 0.000010 0.077551 ± 0.000010	D_4 A_4^*	0.076603	D_4	0.076087
5	2	9 1	0.075624 ± 0.000010 0.075796 ± 0.000010	D_5^* —	0.075625	D_5^*	0.074654
6	2	7 3	0.074240 ± 0.000010 0.074342 ± 0.000010	E_6^* E_6	0.074244	E_6^*	0.073475
7	2	9 1	0.073121 ± 0.000010 0.073234 ± 0.000010	E_7^* E_7	0.073116	E_7^*	0.072484
8	1	10	0.071681 ± 0.000010	E_8	0.071682	E_8	0.071636
9	3	8 1 1	0.071626 ± 0.000002 0.071634 ± 0.000002 0.071640 ± 0.000003	— — —	0.074693	D_9^*	0.070902
10	1	10	0.070814 ± 0.000010	D_{10}^+	0.074701	D_{10}^*	0.070258

hits than the others. It can be seen in table 3.1 that in all dimensions, the principal minimum turned out to be equivalent to the best known d-dimensional lattice—or better! In none of the studied dimensions, our lattice training algorithm failed to reach a performance that has been attained through other design methods. In dimensions 9 and 10, we found lattices that have not been considered for quantization before. Both these cases are discussed in detail below. In dimensions 2–8, the principal minima were equivalent to best known results, which lends confidence to our training algorithm as well as to previous investigations. We do not claim that the principal minimum found by the lattice training algorithm is always the global minimum, but we have not yet seen a counterexample.

Figure 3.2 summarizes the results obtained by lattice training, in relation to the previously known results of figure 3.1. Lattice training carries the normalized second moment much closer to the bound for $d = 9$ and 10. In 10 dimensions, the gain over the best classical lattice, D_{10}^*, is 0.23 dB, a gain that (1.18) indicates can be interpreted as the SNR difference between corresponding lattice quantizers.

We now comment on the results in each dimension; first we give a brief summary of dimensions from 2 to 8, then a more detailed presentation of dimensions 9 and 10, which is where our lattice results improve on previous knowledge. For $d = 2$ and 3, all trials

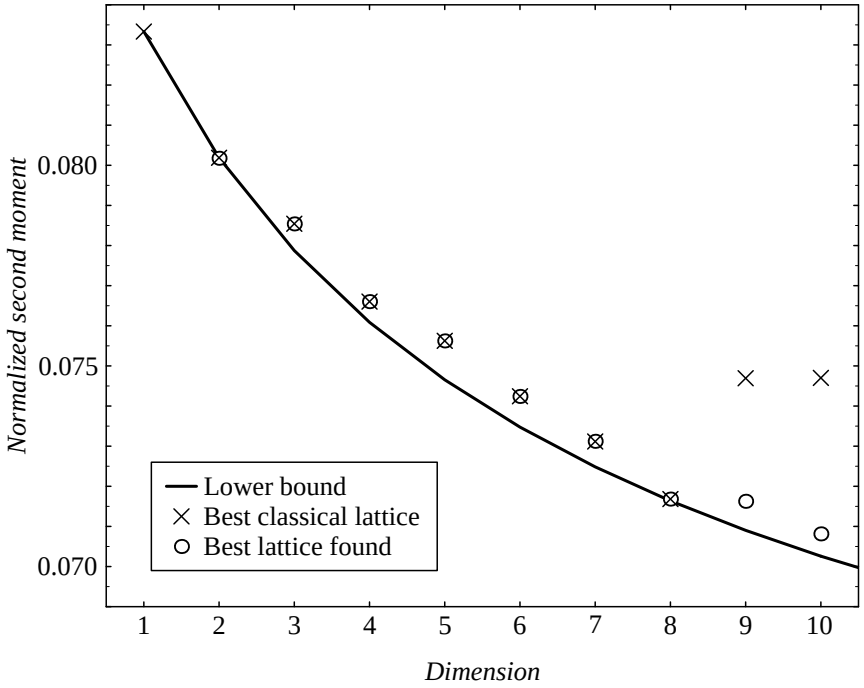

Figure 3.2. A comparison between classical lattices and the principal minima given by the training algorithm.

converged to the same local minimum, A_2 and A_3^*. For $d = 4$ and 5, we found two local minima, of which the principal ones (D_4 and D_5^*) were reached in 9 out of 10 attempts. The suboptimal local minima are A_4^* and an unnamed 5-dimensional sublattice of D_6^*.[14] For $d = 6$ and 7, the principal minimum is E_d^*, and E_d is a secondary minimum. For $d = 8$, the only local minimum found is E_8, which is known as a very good and very symmetrical lattice [41].

Before proceeding to the lattices found in dimensions 9 and 10, we pause to make an observation on the less frequent local minima in table 3.1. Since some local minima only received one hit in 10 attempts, there may well exist other lattices that are also locally optimal, even though none of the 10 runs arrived there. If we wish to estimate the exact number of local minima in a given dimension, 10 trials are apparently insufficient; if, on the other hand, we are more concerned with finding the one global optimum in each dimension, the trend of better local minima getting more hits, as the table indicates, is encouraging. As an example of a thorough search for local minima, we executed the algorithm 100 times in 3 dimensions. All of them converged to A_3^*; hence, it is likely that this is the only 3-dimensional local minimum. Especially, the *face-centered cubic*

[14] The 5-dimensional locally optimal lattice can be obtained as the intersection of D_6^* and a hyperplane perpendicular to the vector $(1,1,1,1,1,1)^T$. This specification of the new lattice is analogous to the definition of E_7 as a sublattice of D_8^*, see the appendix.

lattice, A_3, was never reached. Its G is known to be slightly higher than that of A_3^*; apparently, A_3 is not even locally optimal.

Nine-dimensional lattices are special. The lattices we reached are more irregular than the ones in other dimensions, and none of them were found in the literature. Moreover, they do not appear to be *integer lattices* [2, p. 47] for any scaling, which makes them unique amongst the presently best known lattices. While confusing at first, this irregularity was to some extent explained when we studied nonlattice tessellation (see section 3.2). It turned out that there is a 9-dimensional nonlattice tessellation (yes, a highly regular tessellation!) that is considerably better than all known lattices. If the optimal tessellation for $d = 9$ is not a lattice at all, then the irregular lattices we observe may be attempts by the training algorithm to approximate the nonlattice structure within a lattice constraint.

The generator matrix of the 9-dimensional principal minimum is

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 0.573 \end{bmatrix} \tag{3.1}$$

and the normalized second moment of the lattice was estimated to 0.071622 ± 0.000003. In (3.1), we have interpreted the output of the algorithm according to postulate 1, but for this lattice, the postulate does not give exact values of all matrix elements. One element, 0.573, was left unrounded. We will explain later why no value can be replaced for 0.573 to increase the symmetry of the lattice.

Since postulate 1 yields exact values of all but one element of the generator matrix of this locally optimal lattice, a single-variate optimization can be formulated to increase the accuracy of this one variable. We modified the lattice training algorithm of chapter II for this purpose. The lattice structure was constrained to (3.1) with an unknown variable substituted for the lower right element. A derivative similar to (2.13) was calculated and the algorithm was run to optimize the single variable. The result, based on 100 runs of this single-variate training algorithm, was 0.57321 ± 0.00014, where the interval is again given on the form $\pm 2\hat{\sigma}$, using an estimate $\hat{\sigma}$ of the standard deviation. Hence, the three decimals in (3.1) can be considered significant.

The matrix (3.1) generates a peculiar lattice, in which the lattice points lie closer together along the dth coordinate that along the others. This follows from the fact that the nonzero lattice points closest to the origin are $\pm(0,0,\cdots,0,1.146)$.[15] Just these two points; for example, $(1.146,0,0,\cdots,0)$ is not a lattice point. The distance to the two closest points is 1.146; in other directions, the distance to any lattice point is 2 or greater. Hence, the Voronoi region of this lattice is flat. In is tempting to conclude that the lattice therefore must have a relatively high normalized second moment G, since this measure characterizes how round the Voronoi region is (see section 1.2), but the conclusion is severely wrong. A lower G is not known among 9-dimensional lattices. This lattice apparently compensates its weird 9th coordinate with being extremely round in the first 8 dimensions. The projection of the lattice orthogonal to $(0,0,\cdots,0,1.146)$ is D_8^+, better known as E_8, whose Voronoi region is very round, as mentioned above. The geometry of the 9th coordinate also explains why symmetry arguments will not suffice to identify the local minimum represented by (3.1) completely. To use an analogy, a cylinder cannot be made more symmetrical by changing its height.

The two suboptimal local minima that the algorithm converged into one time each for $d = 9$ display similar irregularities. One of them reminds much of (3.1) above, in that it has a pair of vectors being significantly shorter than any other lattice vector, and the projection orthogonal to them is E_8. The last local optimum has two pairs of extra short vectors, and the projection orthogonal to both of them is E_7^*.

The obtained 10-dimensional lattices, finally, are all equivalent to the lattice generated by

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 \end{bmatrix} \tag{3.2}$$

This lattice is well known, it is called D_{10}^+, but it has, to our knowledge, not been considered for quantization earlier. Its normalized second moment lies very close to the

[15] To see that these points belong to the lattice, use the construction $\mathbf{B}^T\mathbf{u}$ with $\mathbf{u} = \mp(-3,1,1,\cdots,1,-2)$.

TABLE 3.2. LATTICES THAT CHALLENGE CONWAY AND SLOANE'S CONJECTURE. SHADED CELLS DENOTE BEST KNOWN VALUES IN THEIR DIMENSION.

Lattice	G	δ of dual
(3.1)	0.071622 ± 0.000003	0.02857 ± 0.00002
Λ_9^*	0.071769 ± 0.000006	$1/16\sqrt{2} \approx 0.04419$
D_{10}^+	0.070813 ± 0.000003	$1/32 = 0.03125$
Λ_{10}^*	0.071339 ± 0.000009	$1/16\sqrt{3} \approx 0.03608$

lower bound; we estimated $\hat{G} = 0.070813 \pm 0.000003$. More on the D^+ family, including a very fast search method, is discussed in the next section.

There is a famous conjecture regarding the relation between optimal lattices for "quantization" and "packing."[16] It is based on the observation that the best known d-dimensional lattices for the two purposes were always each other's duals. In 1982, Conway and Sloane conjectured that this duality would be true for the optimal lattices in any dimension [40]. The conjecture was supported in [33] and, although some doubts were expressed in [2, p. 62], no counterexample has been presented to date. We now claim that our 9- and 10-dimensional discoveries both are counterexamples to the conjecture, and present evidence in the form of G and δ values.

The best known lattices for packing in 9 and 10 dimensions are called Λ_9 and Λ_{10}, respectively [2, chs. 1 and 6]. The best known lattices for quantization are now given by (3.1) and (3.2), the latter being D_{10}^+. In table 3.2, we show that our new lattices have a lower G than Λ_9^* and Λ_{10}^*, thus showing that Λ_9^* and Λ_{10}^* are not optimal, as the conjecture would imply. We also show that the duals of the new lattices are not optimal packings, which, if it were true, would have been a second way to satisfy the conjecture.

Our new lattice results thus strongly indicate that the conjecture is false, but they do not prove it. A proof would be complete the day one of four shaded values in table 3.2 would be proved optimal. For now, we have to be content with a "counter-conjecture": $d = 9$ is the lowest dimension for which the optimal lattices for quantization and packing are *not* duals.

The present study ends with $d = 10$, but the algorithm is able to design lattices in considerably higher dimensions than this. The only thing that limits the number of dimensions is, as far as we have found, the available time. As an example, a 20-

[16] In the lattice literature, the *quantization* problem is to minimize G, and the *packing* problem is to maximize [2, Ch. 1]

$$\delta = \frac{1}{V}\left(\inf_{\mathbf{x}\notin\Omega}\|\mathbf{x}\|\right)^d$$

dimensional lattice was designed. It took 25 hours, and the normalized second moment of the resulting lattice was estimated to 0.067594 ± 0.000005. The lower bound for $d = 20$ is 0.066457.

3.2 The Best Tessellations Found

In the study of the trained lattices, we were struck by the similarities between the best found lattices in 8 and 10 dimensions. In this section, we generalize the pattern and discover very good nonlattice tessellations in 7 and 9 dimensions. Traditionally, the study of tessellation for quantization applications has been heavily dominated by lattices, see, e.g., [2, p. 61]. This is, to our knowledge, the first time that nonlattice tessellations have shown any competitive performance in relation to lattices.

Compare (3.2) and (A.7): the pattern is obvious. On the other hand, the best found lattice in 9 dimensions (3.1) is different, not very much, but still significantly. The key to this mystery lies in the D^+ family. It is defined as the union of D_d and a translation of D_d [2, pp. 46 and 119]:

$$D_d^+ = D_d \cup \left(D_d + (1/2,\cdots,1/2)^T\right) \tag{3.3}$$

where D_d is defined in the appendix. When d is even, D_d^+ is a lattice, whereas for odd values of d, D_d^+ is a nonlattice tessellation. We will return to the geometrical properties

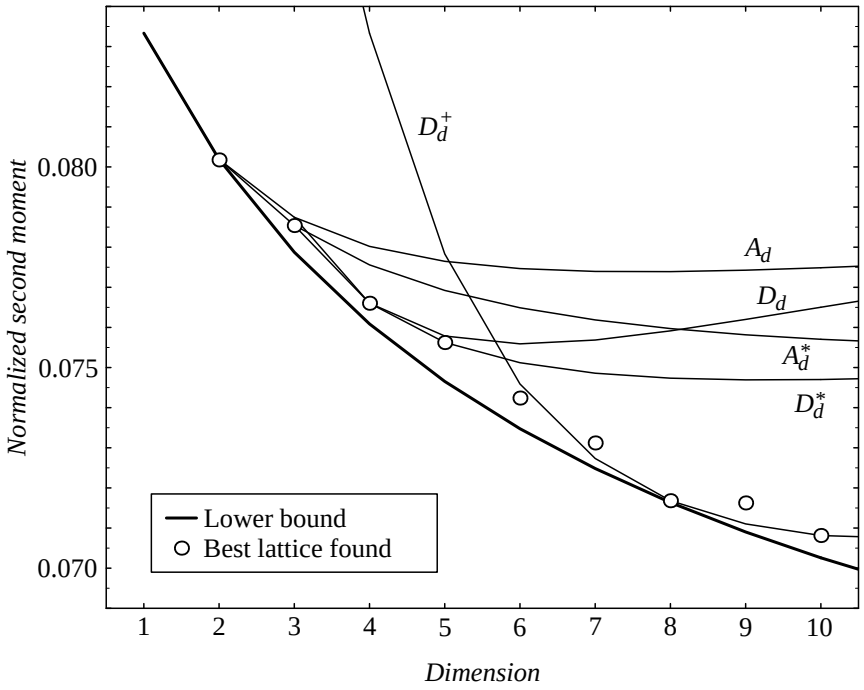

Figure 3.3. The performance of the D^+ tessellation, versus lattices and the lower bound. Note the improvement in 7 and 9 dimensions.

of D_d^+, and the peculiar 9th dimension later in the section. For now, we turn to the main point of interest, namely, the normalized second moment.

Figure 3.3 shows the results in dimensions from 2 to 10. For comparison, the figure includes the conjectured lower bound [41] and the best known lattices, including the new results from the previous section. The performances of the lattices A_d, A_d^*, D_d, and D_d^* [40] are also shown. D_d^+ gives significantly lower normalized second moments than these four lattice families for $d \geq 6$; the curve for D_d^+ indeed passes through the best lattices in 8 and 10 dimensions, as expected. In 7 and 9 dimensions, however, the best known lattices are inferior to D_d^+, which performs considerably closer to the lower bound. That the best lattices have higher normalized second moments than other tessellations in certain dimensions has never before been observed.

In table 3.3, the results in dimensions from 7 to 10 are summarized in terms of normalized second moment. The new contributions of this report are indicated by a shaded background.

Consider figure 3.2 again and observe G for the best lattices, that is, study the pattern formed by the circles. Without too much imagination, the pattern can be described as somewhat zigzag: lattices are in general worse for odd dimensions than for even, compared with the lower bound. This property can, to some extent, be explained by the relation between lattices and nonlattice tessellations. To put it simple, good tessellations are more often lattices in even dimensions than in odd. This observation motivates a closer look upon the geometry of D_d^+.

It is not hard to show that when d is even, D_d^+ is a lattice, with the generator matrix

$$\mathbf{B} = \begin{bmatrix} 2 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1/2 & 1/2 & 1/2 & \cdots & 1/2 \end{bmatrix} \tag{3.4}$$

The situation is more complicated when d is odd. To begin with, (3.4) does *not* generate

TABLE 3.3. THE BEST KNOWN NORMALIZED SECOND MOMENT G FOR LATTICES AND TESSELLATIONS IN DIMENSIONS 7–10. SHADED CELLS DENOTE NEW RESULTS.

d	Best known lattice	Best known tessellation	Lower bound
7	0.073116	0.072734 ± 0.000003	0.072484
8	0.071682	0.071682	0.071636
9	0.071622 ± 0.000003	0.071103 ± 0.000003	0.070902
10	0.070813 ± 0.000003	0.070813 ± 0.000003	0.070258

D_d^+. Instead, it generates a peculiar lattice, which has many characteristics in common with the best found 9-dimensional lattice (3.1) discussed above. E. g., $\pm(0,0,\cdots,0,1)$ are lattice points, so the lattice points lie closer together along the dth coordinate than along the others. For $d = 9$, (3.4) generates Λ_9^*, which was discussed in connection with table 3.2.

So what is the generator matrix of D_d^+ for odd values of d? The answer is that there is none. As mentioned, D_d^+ is not a lattice when d is odd. However, it is still a tessellation, because all its Voronoi regions are congruent. Half of them are upside-down (more precisely, reflected in a point), which disqualifies the tessellation from being a lattice. In a lattice, all Voronoi regions are translations of each other, without scaling, rotation, or reflection, see section 1.3.

Conway and Sloane [2, p. 120] summarize some parameters of the tessellation D_d^+. We conclude this section by adding two facts to their list, without further comments. The *covering radius* is $R = \sqrt{3}/2$ $(d = 3)$, 1 $(4 \le d \le 8)$, $\sqrt{d/8}$ (even $d \ge 8$), or $\sqrt{2d-1}/4$ (odd $d \ge 8$). For even d, the lattice D_d^+ is *geometrically self-dual* [2, p. xix]: the dual is equivalent to the lattice itself.

IV. SUMMARY AND CONCLUSIONS

The contribution of this report is an algorithm for numerical minimization of the normalized second moment of lattices. The algorithm shows the following features:

- It works on a constrained generator matrix with the minimum degrees of freedom needed to encompass all lattices, but only one rotation and scaling of each.
- It is a stochastic gradient algorithm.
- It can operate in dimensions from 2 to at least 20.
- It has initially the ability to escape shallow local minima, but the ability decreases with training time.
- Its output is accurate enough to make possible the identification of exact generator matrices for locally optimal lattices.
- It converges to relatively few local minima. Of the local minima, it tends to favor the better ones.
- In dimensions 2–8, it rediscovers the lattices that have previously been reported as best known.
- In dimension 9, it discovers a new lattice (3.1), which is considerably better than any lattice previously known. It has an uncommon structure for locally optimal lattices.
- In dimension 10, it shows that a significant improvement can be attained by employing D_{10}^{+} instead of the lattices that have been considered before.
- It suggests that the famous duality conjecture by Conway and Sloane may be false.
- It directs us towards the D_d^{+} tessellation, which was shown to perform better than any known lattice tessellation in dimensions 7 and 9. This is the first time that lattices do not hold all second moment world records for tessellations.

As a graphical summary, we conclude this report with figure 4.1, which presents our values (circles and dots) compared with what was previously known.

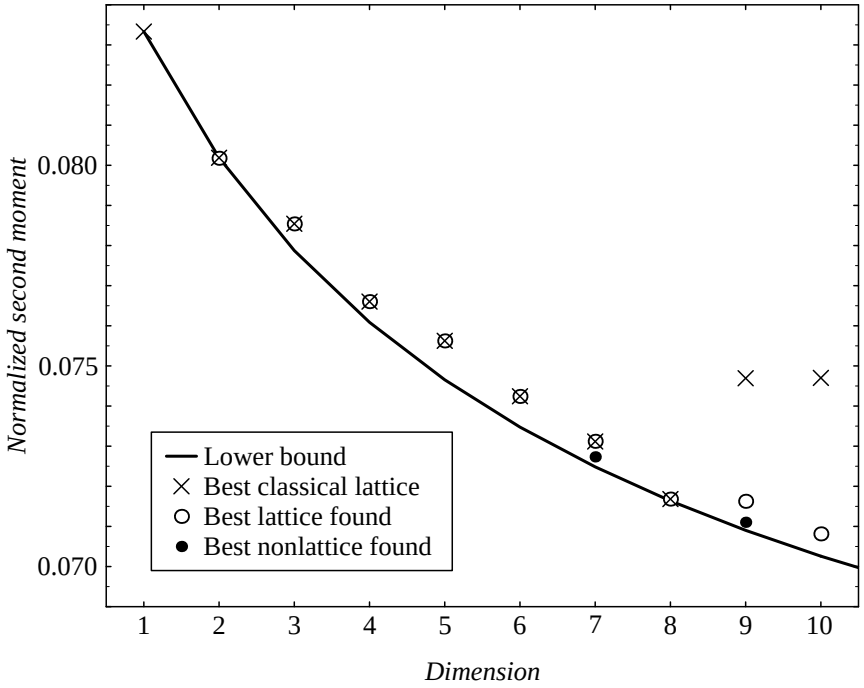

Figure 4.1. A comparison between classical lattices and the lattices and tessellations found through training.

APPENDIX: THE CLASSICAL LATTICES

In this appendix, the lattices $\mathbb{Z}^d$, A_d, D_d, and E_d are briefly defined. A much more thorough treatise on their structure and properties is found in [2, Ch. 4].

The *cubic lattice* $\mathbb{Z}^d$ is the Cartesian product of d 1-dimensional lattices. The generator matrix depends, as discussed in section 1.3, on rotation and choice of basis vectors. One generator matrix for $\mathbb{Z}^d$ is the $d \times d$ identity matrix. The lattice is its own dual.

The lattice A_d can be defined as a sublattice of the cubic lattice: A_d consists of the points of $\mathbb{Z}^{d+1}$ that lie on a hyperplane orthogonal to $(1,1,\cdots,1)^T$. A rotated version of A_d is generated by the matrix[17]

$$\begin{bmatrix} \alpha & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & \alpha \end{bmatrix} \tag{A.1}$$

where $\alpha = \sqrt{d+1} + 2$. A_2 is the hexagonal lattice, see figure 1.1. One choice of a generator matrix of the dual A_d^* is (A.1) with $\alpha = \sqrt{d+1} - d$.

The lattice D_d is defined for $d \geq 3$. It consists of every second point in $\mathbb{Z}^d$, namely, those points whose coordinate sum is even. A generator matrix is

$$\begin{bmatrix} 2 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{A.2}$$

Its dual, D_d^* has the generator matrix

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 1/2 & 1/2 & \cdots & 1/2 & 1/2 \end{bmatrix} \tag{A.3}$$

[17] For consistency, we give square generator matrices for all lattices, which is why some of our definitions may appear unfamiliar. Especially in literature with focus on theory rather than application, it is common to present the lattices A_d, E_6, E_7, and their duals using nonsquare generator matrices.

The E_d family is defined for $d = 6$, 7, and 8 only. E_6 is a sublattice of A_7^*. With A_7^* generated by (A.1), E_6 is the lattice being orthogonal to any of the basis vectors of A_7^*. A generator matrix for E_6 and its dual is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \alpha \\ 0 & 1 & 0 & 0 & 0 & \alpha \\ 0 & 0 & 1 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 1 & 0 & \alpha \\ 0 & 0 & 0 & 0 & 1 & \alpha \\ 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 3\alpha/2 \end{bmatrix} \tag{A.4}$$

with $\alpha = \sqrt{3}$ for E_6 and $\alpha = 1/\sqrt{3}$ for E_6^*. E_7 is the sublattice of D_8^* (A.3) being orthogonal to $(1,1,\cdots,1)^T$. As a generator matrices for E_7, we can use

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{A.5}$$

and for E_7^*,

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{A.6}$$

E_8, finally, is equivalent to D_8^+, see section 3.2. The lattice, which is also equivalent to its dual E_8^*, can be generated by the matrix

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 & 1/2 \end{bmatrix} \tag{A.7}$$

REFERENCES

[1] T. Eriksson and E. Agrell, "Lattice-Based Quantization, Part II," Tech. Report 18, Dept. of Information Theory, Chalmers Univ. of Technology, Göteborg, Sweden, 1996.

[2] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 2nd ed. New York, NY: Springer-Verlag, 1993.

[3] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[4] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84–95, Jan. 1980.

[5] T. Kohonen, *Self-Organization and Associative Memory*, 2nd ed. Berlin, Germany: Springer-Verlag, 1988.

[6] J. K. Flanagan, D. R. Morrell, R. L. Frost, C. J. Read, and B. E. Nelson, "Vector quantization codebook generation using simulated annealing," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1759–1762, Glasgow, Scotland, U.K., May 1989.

[7] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, no. 4, pp. 373–380, July 1979.

[8] T. Linder and K. Zeger, "Asymptotic entropy-constrained performance of tessellating and universal randomized lattice quantization," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 575–579, Mar. 1994.

[9] A. Gersho, "On the structure of vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 2, pp. 157–166, Mar. 1982.

[10] H. Cohn, *A Second Course in Number Theory*. New York, NY: John Wiley & Sons, 1962.

[11] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY: John Wiley & Sons, 1988.

[12] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[13] T. Berger, *Rate Distortion Theory. A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[14] T. R. Fischer, "Geometric source coding and vector quantization," *IEEE Trans. Inform. Theory*, vol. 35, no. 1, pp. 137–145, Jan. 1989.

[15] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the I.R.E.*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[16] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-32, no. 4, pp. 568–583, July 1986.

[17] C. Lamblin, J. P. Adoul, D. Massaloux, and S. Morissette, "Fast CELP coding based on the Barnes-Wall lattice in 16 dimensions," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 61–64, Glasgow, Scotland, U.K., May 1989.

[18] M. Xie and J.-P. Adoul, "Embedded algebraic vector quantizers (EAVQ) with application to wideband speech coding," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 240–243, Atlanta, GA, May 1996.

[19] D. G. Jeong and J. D. Gibson, "Lattice vector quantization for image coding," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1743–1746, Glasgow, Scotland, U.K., May 1989.

[20] D. G. Jeong and J. D. Gibson, "Uniform and piecewise uniform lattice vector quantization for memoryless Gaussian and Laplacian sources," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 786–804, May 1993.

[21] F. Kuhlmann and J. A. Bucklew, "Piecewise uniform vector quantizers," *IEEE Trans. Inform. Theory*, vol. 34, no. 5, pp. 1259–1263, Sept. 1988.

[22] P. F. Swazek, "Unrestricted multistage vector quantizers," *IEEE Trans. Inform. Theory*, vol. 38, no. 3, pp. 1169–1174, May 1992.

[23] T. Eriksson, "Multistage vector quantization with dynamic bit allocation," in *Signal Processing VII: Theories and Applications*, vol. 1, M. J. J. Holt, C. F. N. Cowan, P. M. Grant, and W. A. Sandham, Eds., pp. 383–386, *Proc. EUSIPCO*, Edinburgh, Scotland, U.K., Sept. 1994.

[24] J. Pan and T. R. Fischer, "Two-stage vector quantization—lattice vector quantization," *IEEE Trans. Inform. Theory*, vol. 41, no. 1, pp. 155–163, Jan. 1995.

[25] J. A. Bucklew, "Companding and random quantization in several dimensions," *IEEE Trans. Inform. Theory*, vol. IT-27, no. 2, pp. 207–211, Mar. 1981.

[26] J. A. Bucklew, "A note on optimal multidimensional companders," *IEEE Trans. Inform. Theory*, vol. IT-29, no. 2, p. 279, Mar. 1983.

[27] M. Antonini, M. Barlaud, and T. Gaidon, "Adaptive entropy constrained lattice vector quantization for multiresolution image coding," in *Proceedings of SPIE*, vol. 1818, pt. 2, P. Maragos, Ed., pp. 441–457, *Proc. Visual Communications and Image Processing*, Boston, MA, Nov. 1992.

[28] R. M. Gray, *Source Coding Theory*. Boston, MA: Kluwer Academic Publishers, 1990.

[29] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer Academic Publishers, 1992.

[30] M. Antonini, M. Barlaud, and P. Mathieu, "Image coding using lattice vector quantization of wavelet coefficients," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2273–2276, Toronto, Ontario, Canada, May 1991.

[31] Z. Mohd-Yusof and T. R. Fischer, "An entropy-coded lattice vector quantizer for transform and subband image coding," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 289–298, Feb. 1996.

[32] K. Sayood and S. J. Blankenau, "A fast quantization algorithm for lattice quantizer design," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1168–1171, New York, NY, Apr. 1988.

[33] J. H. Conway and N. J. A. Sloane, "On the Voronoi regions of certain lattices," *SIAM Journal on Algebraic and Discrete Methods*, vol. 5, no. 3, pp. 294–305, Sept. 1984.

[34] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982.

[35] R. Kannan, "Improved algorithms for integer programming and related lattice problems," in *Proc. Annual ACM Symposium on Theory of Computing*, pp. 193–206, Boston, MA, Apr. 1983.

[36] E. Agrell and T. Eriksson, unpublished work, 1996.

[37] L. Babai, "On Lovász' lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[38] E. Viterbo and E. Biglieri, "Computing the Voronoi cell of a lattice: the diamond-cutting algorithm," *IEEE Trans. Inform. Theory*, vol. 42, no. 1, pp. 161–171, Jan. 1996.

[39] 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, no. 2, pp. 227–232, Mar. 1982.

[40] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory*, vol. IT-28, no. 2, pp. 211–226, Mar. 1982.

[41] J. H. Conway and N. J. A. Sloane, "A lower bound on the average error of vector quantizers," *IEEE Trans. Inform. Theory*, vol. IT-31, no. 1, pp. 106–109, Jan. 1985.

[42] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, Nov. 1985.

[43] J. D. Gibson and K. Sayood, "Lattice quantization," in *Advances in Electronics and Electron Physics*, vol. 72, P. W. Hawkes and B. Kazan, Eds. Boston, MA: Academic Press, pp. 259–330, 1988.

[44] K. K. Paliwal and B. S. Atal, “Efficient vector quantization of LPC parameters at 24 bits/frame,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 661–664, Toronto, Ontario, Canada, May 1991.

[45] P. Hedelin, “Single stage spectral quantization at 20 bits,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 525–528, Adelaide, Australia, Apr. 1994.