Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes

J. H. CONWAY AND N. J. A. SLOANE, FELLOW, IEEE

Abstract—**For each of the lattices $A_n (n \geq 1)$, $D_n (n \geq 2)$, E_6, E_7, E_8, and their duals a very fast algorithm is given for finding the closest lattice point to an arbitrary point. If these lattices are used for vector quantizing of uniformly distributed data, the algorithm finds the minimum distortion lattice point. If the lattices are used as codes for a Gaussian channel, the algorithm performs maximum likelihood decoding.**

I. Introduction

THE SO-CALLED *root lattices* are the n-dimensional lattices A_n $(n \geq 1)$, $D_n (n \geq 2)$, and $E_n (n = 6, 7, 8)$ defined in Section II. These lattices and their duals give rise to the densest known sphere packing and coverings in dimensions $n \leq 8$, and they can be used as the basis for efficient block quantizers for uniformly distributed inputs and to construct codes for a band-limited channel with Gaussian noise (see [6], [9], [11], [16]). Around each lattice point is its *Voronoi region*, consisting of all points of the underlying space which are closer to that lattice point than to any other. (Voronoi regions are also called Dirichlet regions, Brillouin zones, Wigner–Seitz cells, or nearest neighbor regions.) If the lattice is used as a quantizer, all the points in the Voronoi region around the lattice point $\boldsymbol{x}$ are represented by $\boldsymbol{x}$; while if the lattice is used as a code for a Gaussian channel, all the points in the Voronoi region around $\boldsymbol{x}$ are decoded as $\boldsymbol{x}$. In the preceding paper [6] we found the Voronoi regions for most of the root lattices and their duals, as well as the mean-squared quantization error when these lattices are used to quantize uniformly distributed data.

In the present paper we give very fast and simple algorithms which, for any of the lattices A_n, D_n, E_n, and their duals (as well as many other lattices), find the closest lattice point to an arbitrary point of the space. In other words the algorithms find which Voronoi region the given point belongs to. The algorithms can therefore be used either for vector quantizing or for channel decoding. The running time of the algorithms for D_n, D_n^*, E_n, and E_n^* is proportional to n, while for A_n and A_n^* it is proportional to $n \log n$ and $n^2 \log n$, respectively.

Although considerable work has been done in the past on n-dimensional quantizers and codes (see [1], [3], [8], [9], [14], [16]–[18], and the references given there), these algorithms appear to be new.

II. Definition of Root Lattices and their Duals

If $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n$ are linearly independent vectors in m-dimensional real Euclidean space $\mathbb{R}^m$ with $m \geq n$, the set of all vectors

$$\boldsymbol{x} = u_1 \boldsymbol{a}_1 + \cdots + u_n \boldsymbol{a}_n,$$

where $u_1, \cdots, u_n$ are arbitrary integers, is called an n-dimensional *lattice* Λ. The *dual* (or *reciprocal*) lattice Λ^* consists of all points y in the subspace of $\mathbb{R}^m$ spanned by $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_n$ such that the inner product $\boldsymbol{x} \cdot \boldsymbol{y} = x_1 y_1 + \cdots + x_m y_m$ is an integer for all $\boldsymbol{x} \in \Lambda$. Most of the lattices considered here are contained in their duals, so that there are coset representatives $\boldsymbol{r}_0, \cdots, \boldsymbol{r}_{d-1}$ such that

$$\Lambda^* = \bigcup_{i=0}^{d-1} (\boldsymbol{r}_i + \Lambda). \tag{1}$$

The number d is called the *determinant* of Λ. The *norm* (or energy) of a vector $\boldsymbol{x} = (x_1, \cdots, x_m) \in \mathbb{R}^m$ is

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}} = \left(\sum x_i^2\right)^{1/2}.$$

For further information about lattices see for example [2], [4]–[7], [11], [12], [15], [16].

For $n \geq 1$, A_n is the n-dimensional[1] lattice consisting of the points $(x_0, x_1, \cdots, x_n)$ having integer coordinates that sum to zero. (Thus A_n is an n-dimensional lattice described by $(n+1)$-dimensional coordinates.) If two lattices A and B differ only by a rotation and change of scale, we say they are equivalent and write $A \cong B$. Then A_1 is equivalent to the one-dimensional lattice of integer points, $\mathbb{Z}$, and A_2 is equivalent to the familiar two-dimensional hexagonal lattice (see Fig. 1).

The dual lattice A_n^* consists of the union of $n+1$ cosets of A_n:

$$A_n^* = \bigcup_{i=0}^{n} (\boldsymbol{r}_i + A_n), \tag{2}$$

Manuscript received May 7, 1981; revised November 3, 1981.

J. H. Conway is with the Department of Pure Mathematics and Mathematical Statistics, University of Cambridge, 16 Mill Lane, Cambridge CB2 1SB, England.

N. J. A. Sloane is with the Mathematics and Statistics Research Center, Bell Laboratories, Murray Hill, NJ 07974.

[1] The subscript gives the dimension of the lattice.

0018-9448/82/0300-0227$00.75 ©1982 IEEE

where, for $i = 0, 1, \cdots, n$ and $j = n + 1 - i$,

$$r_i = \left(\frac{-j}{n+1}, \frac{-j}{n+1}, \cdots, \frac{-j}{n+1}, \frac{i}{n+1}, \cdots, \frac{i}{n+1} \right)$$
$$= \left(\left(\frac{-j}{n+1} \right)^i, \left(\frac{i}{n+1} \right)^j \right). \qquad (3)$$

Then $A_1^* \cong A_1 \cong \mathbb{Z}$ and $A_2^* \cong A_2$. Also A_3 $(\cong D_3)$ is the face-centered cubic lattice and A_3^* $(\cong D_3^*)$ is the body-centered cubic lattice. It is known that A_2 is the optimal two-dimensional quantizer for a uniformly distributed input, and it is conjectured that A_3^* is optimal in three dimensions.

For $N \geq 2$, D_n consists of the points $(x_1, x_2, \cdots, x_n)$ having integer coordinates with an even sum. The dual D_n^* is the union of four cosets of D_n:

$$D_n^* = \bigcup_{i=0}^{3} (r_i + D_n), \qquad (4)$$

where

$$r_0 = (0^n), \quad r_1 = \left(\tfrac{1}{2}^n\right),$$
$$r_2 = (0^{n-1}, 1), \quad r_3 = \left(\tfrac{1}{2}^{n-1}, -\tfrac{1}{2}\right).$$

Also $D_2 \cong A_1 \oplus A_1 \cong D_2^*$, $D_3 \cong A_3$, and $D_4^* \cong D_4$.

For example the vectors of small norm in D_4 consist of all permutations and sign changes of the following:

type	number	norm^2
$(0, 0, 0, 0)$	1	0
$(\pm 1, \pm 1, 0, 0)$	24	2
$(\pm 2, 0, 0, 0)$	8	4
$(\pm 1, \pm 1, \pm 1, \pm 1)$	16	4
$(\pm 2, \pm 1, \pm 1, 0)$	96	6
$(\pm 2, \pm 2, 0, 0)$	24	8
$\cdots$	$\cdots$	$\cdots$

(5)

(See also [16, Table V].)

There is a second, equivalent definition of D_n^* which is sometimes easier to use: D_n^* consists of the points of the n-dimensional integer lattice $\mathbb{Z}^n$, together with the translate of $\mathbb{Z}^n$ by the vector $(\frac{1}{2}, \frac{1}{2}, \cdots, \frac{1}{2})$, i.e.,

$$D_n^* = \mathbb{Z}^n \cup \left(\left(\tfrac{1}{2}^n\right) + \mathbb{Z}^n \right). \qquad (6)$$

The most convenient definitions of E_6, E_7, and E_8 are the following.[2] E_8 is the union of D_8 and the coset

$$\left(\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}\right) + D_8.$$

In other words E_8 consists of the points $(x_1, \cdots, x_8)$ with $x_i \in \mathbb{Z}$ and Σx_i even, together with the points $(y_1, \cdots, y_8)$ with $y_i \in \mathbb{Z} + \frac{1}{2}$ and Σy_8 even. Also $E_8^* = E_8$.

E_7 is a subspace of dimension 7 in E_8, consisting of the points $(u_1, \cdots, u_8)$ in E_8 with $\Sigma u_i = 0$. Equivalently

$$E_7 = A_7 \cup \left(\left(-\tfrac{1}{2}^4, \tfrac{1}{2}^4 \right) + A_7 \right). \qquad (7)$$

The dual E_7^* is given by

$$E_7^* = E_7 \cup \left(\left(-\tfrac{3}{4}^2, \tfrac{1}{4}^6 \right) + E_7 \right)$$
$$= \bigcup_{i=0}^{3} (s_i + A_7), \qquad (8)$$

where

$$s_i = \left(\left(\frac{-j}{4} \right)^{2i}, \left(\frac{i}{4} \right)^{2j} \right), \qquad i + j = 4. \qquad (9)$$

Finally E_6 is a subspace of dimension 6 in E_8, which we may take for example to consist of the points $(u_1, \cdots, u_8)$ in E_8 with $\Sigma u_i = 0$ and $u_1 + u_8 = 0$. However this lattice and its dual do not appear to be as important as the others for the applications considered here, and we shall not discuss them in as much detail.

III. Finding the Closest Point of the n-Dimensional Integer Lattice $\mathbb{Z}^n$

The algorithm for finding the closest point of the integer lattice $\mathbb{Z}^n$ to an arbitrary point $x \in \mathbb{R}^n$ is particularly simple, and serves to introduce the notation. For a real number x, let

$$f(x) = \text{closest integer to } x.$$

In case of a tie, choose the integer with the smallest absolute value. For $x = (x_1, \cdots, x_n) \in \mathbb{R}^n$, let

$$f(x) = (f(x_1), \cdots, f(x_n)).$$

For later use we also define $g(x)$, which is the same as $f(x)$ except that the *worst* component of x—that furthest from an integer—is rounded the *wrong way*. In case of a tie, the component with the lowest subscript is rounded the wrong way.

More formally, for $x \in \mathbb{R}$ we define $f(x)$ and the function $w(x)$ which rounds the wrong way as follows. (Here m is an integer.)

$$\begin{array}{ll}
\text{If } x = 0, & \text{then } f(x) = 0, \\
 & w(x) = 1. \\
\text{If } 0 < m \leq x \leq m + \frac{1}{2}, & \text{then } f(x) = m, \\
 & w(x) = m + 1. \\
\text{If } 0 < m + \frac{1}{2} < x < m + 1, & \text{then } f(x) = m + 1, \\
 & w(x) = m. \\
\text{If } -m - \frac{1}{2} \leq x \leq -m < 0, & \text{then } f(x) = -m, \\
 & w(x) = -m - 1. \\
\text{If } -m - 1 < x < -m - \frac{1}{2}, & \text{then } f(x) = -m - 1, \\
 & w(x) = -m.
\end{array} \qquad (10)$$

(Ties are handled so as to give preference to points of

[2] A slightly different (although equivalent) definition of E_7 was used in [6]. The present definition leads to a simpler algorithm in Section VII.

smaller norm.) We also write

$$x = f(x) + \delta(x),$$

so that $|\delta(x)| \leq 1/2$ is the distance from x to the nearest integer.

Given $\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n$, let k $(1 \leq k \leq n)$ be such that

$$|\delta(x_k)| \leq |\delta(x_i)| \qquad \text{for all } 1 \leq i \leq n$$

and

$$|\delta(x_k)| = |\delta(x_i)| \Rightarrow k \leq i.$$

Then $g(\boldsymbol{x})$ is defined by

$$g(\boldsymbol{x}) = (f(x_1), f(x_2), \cdots, w(x_k), \cdots, f(x_n)).$$

Algorithm 1—To Find the Closest Point of $\mathbb{Z}^n$*to* $\boldsymbol{x}$*:* Given $\boldsymbol{x} \in \mathbb{R}^n$, the closest point of $\mathbb{Z}^n$ is $f(\boldsymbol{x})$. (If $\boldsymbol{x}$ is equidistant from two or more points of $\mathbb{Z}^n$, this procedure finds the one with the smallest norm.)

To see that the procedure works, let $\boldsymbol{u} = (u_1, \cdots, u_n)$ be any point of $\mathbb{Z}_n$. Then

$$\|\boldsymbol{u} - \boldsymbol{x}\|^2 = \sum_{i=1}^{n} (u_i - x_i)^2,$$

which is minimized by choosing $u_i = f(x_i)$ for $i = 1, \cdots, n$. Because of (10) ties are broken correctly, favoring the point with the smallest norm.

IV. Finding the Closest Point of D_n

Algorithm 2—To Find the Closest Point of D_n *to* $\boldsymbol{x}$*:* Given $\boldsymbol{x} \in \mathbb{R}^n$, the closest point of D_n is whichever of $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ has an even sum of components (one will have an even sum, the other an odd sum). If $\boldsymbol{x}$ is equidistant from two or more points of D_n this procedure produces a nearest point having the smallest norm.

This procedure works because $f(\boldsymbol{x})$ is the closest point of $\mathbb{Z}^n$ to $\boldsymbol{x}$ and $g(\boldsymbol{x})$ is the next closest. $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ differ by one in exactly one coordinate, and so precisely one of $\Sigma f(x_i)$ and $\Sigma g(x_i)$ is even and the other is odd. Again (10) implies that ties are broken correctly.

Example: Find the closest point of D_4 to $\boldsymbol{x} = (0.6, -1.1, 1.7, 0.1)$. We compute

$$f(\boldsymbol{x}) = (1, -1, 2, 0)$$

and

$$g(\boldsymbol{x}) = (0, -1, 2, 0),$$

since the first component of $\boldsymbol{x}$ is the furthest from an integer. The sum of the components of $f(\boldsymbol{x})$ is $1 - 1 + 2 + 0 = 2$, which is even, while that of $g(\boldsymbol{x})$ is $0 - 1 + 2 + 0 = 1$, which is odd. Therefore $f(\boldsymbol{x})$ is the point of D_4 closest to $\boldsymbol{x}$.

To illustrate how ties are handled, suppose

$$\boldsymbol{x} = (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}).$$

In fact (see (5)) $\boldsymbol{x}$ is now equidistant from eight points of D_4, namely $(0,0,0,0)$, any permutation of $(1,1,0,0)$, and $(1,1,1,1)$. The algorithm computes

$$\begin{aligned} f(\boldsymbol{x}) &= (0,0,0,0), \quad \text{sum} = 0, \quad \text{even},\\ g(\boldsymbol{x}) &= (1,0,0,0), \quad \text{sum} = 1, \quad \text{odd}, \end{aligned}$$

and selects $f(\boldsymbol{x})$. Indeed $f(\boldsymbol{x})$ does have the smallest norm of the eight neighboring points.

V. Finding the Closest Point of a Coset or of a Dual Lattice

A procedure Φ for finding the closest point of a lattice Λ to a given point $\boldsymbol{x}$ can be easily converted to a procedure for finding the closest point of a coset $\boldsymbol{r} + \Lambda$ to $\boldsymbol{x}$. For if $\Phi(\boldsymbol{x})$ is the closest point of Λ to $\boldsymbol{x}$,

$$\Phi(\boldsymbol{x} - \boldsymbol{r}) + \boldsymbol{r}$$

is the closest point of $\boldsymbol{r} + \Lambda$ to $\boldsymbol{x}$.

Suppose further that L is a lattice (or in fact any set of points) which is a union of cosets of Λ:

$$L = \bigcup_{i=0}^{d-1} (\boldsymbol{r}_i + \Lambda).$$

Then Φ can be used as the basis for the following procedure for finding the closest point of L.

Algorithm 3—To Find the Closest Point of L (A Union of Cosets of a Lattice) to a Given Point $\boldsymbol{x}$*:* Given $\boldsymbol{x}$, compute

$$y_i = \Phi(\boldsymbol{x} - \boldsymbol{r}_i) + \boldsymbol{r}_i$$

for $i = 0, 1, \cdots, d-1$. Compare each of $y_0, \cdots, y_{d-1}$ with $\boldsymbol{x}$ and choose the closest.

In view of (1), (2), (4), (8) this algorithm reduces the problem of finding the closest point of the dual lattices A_n^*, D_n^* and E_6^*, E_7^*, E_8^* to that of finding the closest point of the original lattices. Alternatively, a faster algorithm for D_n^* is obtained from (6) and Algorithm 1. We illustrate Algorithm 3 by applying it first to D_5^* using definition (4), and then to D_3^* using definition (6).

For the first example we observe from (4) that D_5^* is the union of four cosets of D_5 with coset representatives

$$\begin{aligned} \boldsymbol{r}_0 &= (0, 0, 0, 0, \quad 0),\\ \boldsymbol{r}_1 &= (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \quad \tfrac{1}{2}),\\ \boldsymbol{r}_2 &= (0, 0, 0, 0, \quad 1),\\ \boldsymbol{r}_3 &= (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, -\tfrac{1}{2}). \end{aligned}$$

To find the closest point of D_5^* to

$$\boldsymbol{x} = (0.1, 0, -0.3, 0.4, 0.8)$$

we proceed as follows:

$$\begin{aligned} f(\boldsymbol{x}) &= (0,0,0,0,1), \quad \text{sum} = 1, \quad \text{odd},\\ g(\boldsymbol{x}) &= (0,0,0,1,1), \quad \text{sum} = 2, \quad \text{even}, \end{aligned}$$

therefore

$$y_0 = (0,0,0,1,1).$$

Also

$$\begin{aligned} \boldsymbol{x} - \boldsymbol{r}_1 &= (-0.4, -0.5, -0.8, -0.1, 0.3),\\ f(\boldsymbol{x} - \boldsymbol{r}_1) &= (0,0,-1,0,0), \quad \text{sum} = -1, \quad \text{odd},\\ g(\boldsymbol{x} - \boldsymbol{r}_1) &= (0,-1,-1,0,0), \quad \text{sum} = -2, \quad \text{even}, \end{aligned}$$

therefore

$$\begin{aligned} \boldsymbol{y}_1 &= g(\boldsymbol{x} - \boldsymbol{r}_1) + \boldsymbol{r}_1\\ &= (0.5, -0.5, -0.5, 0.5, 0.5);\\ \boldsymbol{x} - \boldsymbol{r}_2 &= (0.1, 0, -0.3, 0.4, -0.2),\\ f(\boldsymbol{x} - \boldsymbol{r}_2) &= (0,0,0,0,0), \quad \text{sum} = 0, \quad \text{even},\\ g(\boldsymbol{x} - \boldsymbol{r}_2) &= (0,0,0,1,0), \quad \text{sum} = 1, \quad \text{odd}, \end{aligned}$$

therefore

$$\begin{aligned} \boldsymbol{y}_2 &= f(\boldsymbol{x} - \boldsymbol{r}_2) + \boldsymbol{r}_2\\ &= (0,0,0,0,1); \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{x} - \boldsymbol{r}_3 &= (-0.4, -0.5, -0.8, -0.1, 1.3),\\ f(\boldsymbol{x} - \boldsymbol{r}_3) &= (0,0,-1,0,1), \quad \text{sum} = 0, \quad \text{even},\\ g(\boldsymbol{x} - \boldsymbol{r}_3) &= (0,-1,-1,0,1), \quad \text{sum} = -1, \quad \text{odd}, \end{aligned}$$

therefore

$$\begin{aligned} \boldsymbol{y}_3 &= f(\boldsymbol{x} - \boldsymbol{r}_3) + \boldsymbol{r}_3\\ &= (0.5, 0.5, -0.5, 0.5, 0.5). \end{aligned}$$

The final step is to see which of $\boldsymbol{y}_0, \cdots, \boldsymbol{y}_3$ is closest to $\boldsymbol{x}$. We compute

$$\begin{aligned} \|\boldsymbol{x} - \boldsymbol{y}_0\|^2 &= 0.5,\\ \|\boldsymbol{x} - \boldsymbol{y}_1\|^2 &= 0.55,\\ \|\boldsymbol{x} - \boldsymbol{y}_2\|^2 &= 0.3,\\ \|\boldsymbol{x} - \boldsymbol{y}_3\|^2 &= 0.55. \end{aligned}$$

Thus

$$\boldsymbol{y}_2 = (0,0,0,0,1)$$

is the point of D_5^* closest to $\boldsymbol{x}$.

For the second example we use (6) to define the body-centered cubic lattice D_3^* as the union of two cosets of $\mathbb{Z}^3$ with coset representatives

$$\begin{aligned} \boldsymbol{r}_0 &= (0,0,0),\\ \boldsymbol{r}_1 &= (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}). \end{aligned}$$

To find the closest point of D_3^* to

$$\boldsymbol{x} = (0.2, 0.5, 0.8),$$

we compute

$$\begin{aligned} \boldsymbol{y}_0 &= f(\boldsymbol{x}) = (0,0,1),\\ \boldsymbol{x} - \boldsymbol{r}_1 &= (-0.3, 0, 0.3),\\ f(\boldsymbol{x} - \boldsymbol{r}_1) &= (0,0,0),\\ \boldsymbol{y}_1 &= \boldsymbol{r}_1 + f(\boldsymbol{x} - \boldsymbol{r}_1) = (0.5, 0.5, 0.5). \end{aligned}$$

The final step is to find which of $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$ is closer to $\boldsymbol{x}$:

$$\|\boldsymbol{x} - \boldsymbol{y}_0\|^2 = 0.33, \quad \|\boldsymbol{x} - \boldsymbol{y}_1\|^2 = 0.18.$$

Thus $\boldsymbol{y}_1 = (0.5, 0.5, 0.5)$ is the closest point of D_3^* to $\boldsymbol{x}$.

Obviously the second definition of D_n^* leads to the faster algorithm. For any n it is only necessary to compute $f(\boldsymbol{x})$, $f(\boldsymbol{x} - \boldsymbol{r}_1)$, and to calculate the squared norms $\|\boldsymbol{x} - \boldsymbol{y}_0\|^2$ and $\|\boldsymbol{x} - \boldsymbol{y}_1\|^2$.

VI. Finding the Closest Point of E_8

Since E_8 is the union of two cosets of D_8 the discussion in the previous section leads to the following procedure.

Algorithm 4—To Find the Closest Point of E_8 to $\boldsymbol{x}$: Given $\boldsymbol{x} = (x_1, \cdots, x_8) \in \mathbb{R}^8$.

Compute $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$, and select whichever has an even sum of components; call it $\boldsymbol{y}_0$.

Compute $f(\boldsymbol{x} - \tfrac{1}{2})$ and $g(\boldsymbol{x} - \tfrac{1}{2})$, where

$$\tfrac{1}{2} = (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}),$$

and select whichever has an even sum of components; add $\tfrac{1}{2}$ and call the result $\boldsymbol{y}_1$.

Compare $\boldsymbol{y}_0$ and $\boldsymbol{y}_1$ with $\boldsymbol{x}$ and choose the closest.

For example, to find the closest point of E_8 to

$$\boldsymbol{x} = (0.1, 0.1, 0.8, 1.3, 2.2, -0.6, -0.7, 0.9),$$

we compute

$$\begin{aligned} f(\boldsymbol{x}) &= (0,0,1,1,2,-1,-1,1), \quad \text{sum} = 3, \quad \text{odd},\\ g(\boldsymbol{x}) &= (0,0,1,1,2,0,-1,1), \quad \text{sum} = 4, \quad \text{even}, \end{aligned}$$

and take $\boldsymbol{y}_0 = g(\boldsymbol{x})$. Also

$$\begin{aligned} \boldsymbol{x} - \tfrac{1}{2} &= (-0.4, -0.4, 0.3, 0.8, 1.7, -1.1, -1.2, 0.4),\\ f(\boldsymbol{x} - \tfrac{1}{2}) &= (0,0,0,1,2,-1,-1,0), \quad \text{sum} = 1, \quad \text{odd},\\ g(\boldsymbol{x} - \tfrac{1}{2}) &= (-1,0,0,1,2,-1,-1,0), \quad \text{sum} = 0, \quad \text{even}, \end{aligned}$$

and so

$$\begin{aligned} \boldsymbol{y}_1 &= g(\boldsymbol{x} - \tfrac{1}{2}) + \tfrac{1}{2}\\ &= (-0.5, 0.5, 0.5, 1.5, 2.5, -0.5, -0.5, 0.5). \end{aligned}$$

Finally,

$$\|\boldsymbol{x} - \boldsymbol{y}_0\|^2 = 0.65, \quad \|\boldsymbol{x} - \boldsymbol{y}_1\|^2 = 0.95$$

and we conclude that $\boldsymbol{y}_0 = g(\boldsymbol{x})$ is the closest point to $\boldsymbol{x}$.

VII. Finding the Closest Point of A_n, A_n^*, E_7, and E_7^*

Algorithm 5—To Find the Closest Point of A_n to $\boldsymbol{x}$:

Step 1: Given $\boldsymbol{x} \in \mathbb{R}^{n+1}$, compute $s = \Sigma x_i$ and replace $\boldsymbol{x}$ by

$$\boldsymbol{x}' = \boldsymbol{x} - \frac{s}{n+1}(1, 1, \cdots, 1).$$

Step 2: Calculate $f(\boldsymbol{x}') = (f(x_0'), \cdots, f(x_n'))$ and the *deficiency* $\Delta = \Sigma f(x_i')$.

Step 3: Sort the x_i' in order of increasing value of $\delta(x_i')$ (defined in Section III). We obtain a rearrangement of the numbers $0, 1, \cdots, n$, say $i_0, i_1, \cdots, i_n$, such that

$$-\tfrac{1}{2} \leq \delta(x_{i_0}') \leq \delta(x_{i_1}') \cdots \leq \delta(x_{i_n}') \leq \tfrac{1}{2}.$$

Step 4: If $\Delta = 0$, $f(\boldsymbol{x}')$ is the closest point of A_n to $\boldsymbol{x}$.
If $\Delta > 0$, the closest point is obtained by subtracting 1 from the components $f(x'_{i_0}), \cdots, f(x'_{i_{\Delta-1}})$.
If $\Delta < 0$, the closest point is obtained by adding 1 to the components $f(x'_{i_n}), f(x'_{i_{n-1}}), \cdots, f(x'_{i_{n-\Delta+1}})$.

Remarks: We know from Section III that $f(\boldsymbol{x})$ is the closest point of $\mathbb{Z}^{n+1}$ to $\boldsymbol{x}$. The procedure described here finds the closest point of A_n because it makes the smallest changes to the norm of $f(\boldsymbol{x}')$ needed to make $\Sigma f(x'_i)$ vanish.

Step 1 projects $\boldsymbol{x}$ onto $\boldsymbol{x}'$, the closest point of the hyperplane $\Sigma x_i = 0$. Since A_n is by definition contained in this hyperplane it may be possible to assume that $\boldsymbol{x}$ already lies there, in which case Step 1 can be omitted.

The only substantial amount of computation needed is for the sort in Step 3, which takes $O(n \log n)$ steps [10], [13]. However Step 3 can be omitted if $\boldsymbol{x}$ is expected to be close to A_n. In this case Δ will be small, and Steps 3 and 4 can be replaced by the following:

Step 3': If $\Delta = 0$, $f(\boldsymbol{x}')$ is the closest point of A_n to $\boldsymbol{x}$.

If $\Delta > 0$, find the Δ components of $\boldsymbol{x}'$, say $x'_{i_0}, \cdots, x'_{i_{\Delta-1}}$, for which $\delta(x'_i)$ is as small (i.e., as close to $-\frac{1}{2}$) as possible. The closest point of A_n is obtained by subtracting one from the components $f(x'_{i_0}), \cdots, f(x'_{i_{\Delta-1}})$ of $f(\boldsymbol{x})$.

If $\Delta < 0$, find the $|\Delta|$ components of $\boldsymbol{x}'$, say $x'_{i_n}, x'_{i_{n-1}}, \cdots, x'_{i_{n-\Delta+1}}$, for which $\delta(x'_i)$ is as large (i.e., as close to $\frac{1}{2}$) as possible. The closest point of A_n is obtained by adding 1 to the components of $f(x'_{i_n}), \cdots, f(x'_{i_{n-\Delta+1}})$ of $f(\boldsymbol{x})$.

In any case $|\Delta|$ cannot exceed $n/2$. However if Δ is expected to be large the first version of the algorithm is preferable.

The closest point of A_n^* can be found by Algorithm 3, using the fact that A_n^* is the union of $n+1$ cosets of A_n (see (2)).

For example the hexagonal lattice A_2 is shown in Fig. 1, together with ordinary two-dimensional coordinates (u_1, u_2) for the points. The three-dimensional coordinates (x_0, x_1, x_2) with $x_0 + x_1 + x_2 = 0$ that we have used are obtained by multiplying (u_1, u_2) on the right by the matrix

$$M = \begin{pmatrix} 1 & 0 & -1 \\ \frac{1}{\sqrt{3}} & \frac{-2}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix}.$$

Conversely the u-coordinates may be obtained from the x-coordinates by

$$(u_1, u_2) = (x_0, x_1, x_2) \cdot \tfrac{1}{2} M^{\text{tr}}.$$

For example the points $(0,0)$, $(1,0)$, $(1/2, \sqrt{3}/2)$, $(-1/2, \sqrt{3}\,2)$ have x-coordinates $(0,0,0)$, $(1,0,-1)$, $(1,-1,0)$, $(0,-1,1)$, respectively. To find the closest point of A_n to the point P with coordinates

$$(u_1, u_2) = (0.4, -0.4)$$

we first find the x-coordinates of P, which are

$$\boldsymbol{x} = (x_0, x_1, x_2) = (0.169, 0.462, -0.631).$$

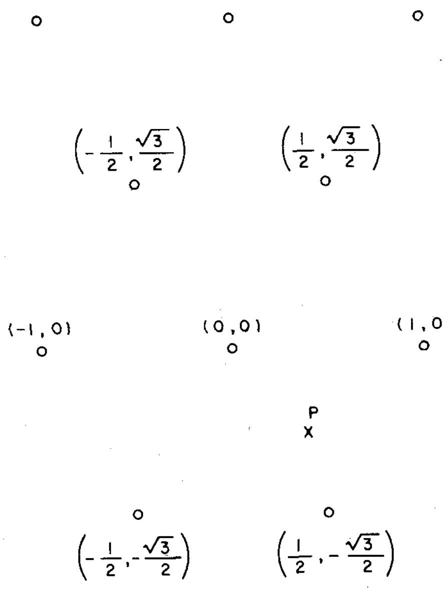
$\left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$
$\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$
(-1, 0)
(0, 0)
(1, 0)
P
$\left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$
$\left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$

Fig. 1. The hexagonal lattice A_2.

Step 1 of the algorithm can be omitted, since $x_0 + x_1 + x_2 = 0$ holds automatically. Step 2 produces

$$f(\boldsymbol{x}) = (0, 0, -1),$$

with deficiency $\Delta = -1$. At Step 3 we find

$$\delta(x_0) = 0.169 < \delta(x_2) = 0.369 < \delta(x_1) = 0.462.$$

At Step 4 we add 1 to $f(x_1)$, obtaining

$$(0, 1, -1)$$

which is the closest point of A_2. The u-coordinates for this point are

$$(0, 1, -1) \cdot \frac{1}{2} M^{\text{tr}} = \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$$

(see Fig. 1).

Since $A_3 \cong D_3$, Algorithm 2 is preferable to Algorithm 5 for finding the closest point of the face-centered cubic lattice. Finally E_7 and E_7^* can be handled via the algorithm for A_7, using (7), (8), and Algorithm 3.

Acknowledgment

During the early stages of this work we had some helpful discussions with Allen Gersho.

References

[1] T. Berger, "Optimum quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-18, no. 6, 759–765, Nov. 1972.

[2] N. Bourbaki, *Groupes et Algèbres de Lie, Chapitres 4, 5 et 6*. Paris: Hermann, 1968.

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

[4] J. W. S. Cassels, *An Introduction to the Geometry of Numbers*. New York: Springer-Verlag, 1971.

[5] J. H. Conway and N. J. A. Sloane, "On the enumeration of lattices of determinant one," *J. Number Theory*, to be published.

[6] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inform. Theory*, this issue on pp. 211–226.

[7] H. S. M. Coxeter, *Regular Polytopes*, 3rd ed. New York: Dover, 1973.

[8] N. C. Gallagher, Jr., and J. A. Bucklew, "Some recent developments in quantization theory," in *Proc. 12th Annu. Sympos. System Theory*, Virginia Beach, VA, May 19–20, 1980.

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

[10] D. E. Knuth, *The Art of Computer Programming*, vol. 3, *Sorting and Searching*. Reading, MA: Addison-Wesley, 1973.

[11] J. Leech and N. J. A. Sloane, "Sphere packing and error-correcting codes," *Canad. J. Math.*, 23, 718–745, 1971.

[12] H. V. Niemeier, "Definite quadratische Formen der Dimension 24 und Diskriminante 1," *J. Number Theory*, 5, 142–178, 1973.

[13] E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[14] K. D. Rines and N. C. Gallagher, Jr., "The design of multidimensional quantizers using prequantization," in *Proc. 18th Allerton Conf. Commun., Control, and Computing*, Univ. of Illinois, Monticello, IL, Oct. 8–10, 1980, pp. 446–453.

[15] N. J. A. Sloane, "Binary codes, lattices and sphere packings," in *Combinatorial Surveys* (Proc. 6th British Combinatorial Conference), P. J. Cameron, Ed. London and New York: Academic, 1977, pp. 117–164.

[16] ——, "Tables of sphere packings and spherical codes," *IEEE Trans. Inform. Theory*, vol. IT-27, no. 3, 327–338, May 1981.

[17] Y. Yamada, S. Tazaki, and R. M. Gray, "Asymptotic performance of block quantizers with difference distortion measures," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 1, 5–14, Jan. 1980.

[18] P. Zador, "Topics in the asymptotic quantization of continuous random variables," *IEEE Trans. Inform. Theory*, this issue on pp. 139–149.

The Design of Two-Dimensional Quantizers using Prequantization

KERRY D. RINES, MEMBER, IEEE, AND NEAL C. GALLAGHER, JR., MEMBER, IEEE

Abstract—**The theoretical advantages of two-dimensional quantization over univariate quantization have been studied in the literature. However, in many cases there is no known implementation for the two-dimensional quantizer that can operate in real time. A new approach to the design of two-dimensional quantizers is presented. This technique, called prequantization, is used to design two-dimensional quantizers that operate in real time. The importance of prequantization is demonstrated by the design of the optimum uniform two-dimensional (hexagonal) quantizer. Additional examples are given to illustrate the flexibility of this design approach.**

I. Introduction

THE USE OF two-dimensional quantizers for encoding analog sources has been of increasing interest in recent years. Two-dimensional quantizers can offer advantages in the design of both optimum and suboptimum quantizers. These advantages may be offset by the difficulty in implementing many two-dimensional quantizers. In this paper we present a new approach to the design of two-dimensional quantizers called prequantization. We show that for a number of examples prequantization simplifies the quantizer implementation and/or improves the quantizer performance.

The design of two-dimensional quantizers for optimum quantization is one area of interest. Consider the random sequence $x_1, x_2, x_3, \cdots$ where the x_i are all independent and identically distributed. The traditional approach to quantizing this sequence is to perform the quantization one sample at a time using a one-dimensional quantizer. Much of the early work in quantization theory has addressed this problem. As a result the design and implementation of optimum one-dimensional quantizers is straightforward. In addition these quantizers are often able to operate at high source rates. These properties make one-dimensional quantization an attractive choice for quantizing the above sequence. The advantage of quantizing the independent identically distributed (i.i.d.) sequence in two or more dimensions is discussed by Zador [1]. Simply stated, these results indicate that the minimum obtainable per sample distortion decreases as the quantizer dimension is increased. Therefore, the potential exists to improve the performance of digital encoders by replacing one-dimensional quantizers with two-dimensional quantizers.

Zador's results include derivations of both the upper and lower bounds on the distortion obtained when using an optimum quantizer. Unfortunately, these results do not provide insight into the structure of the quantizer. The design and implementation of optimum two-dimensional quantizers remains a largely unsolved problem. Recently the design of two-dimensional quantizers has been addressed. Computer algorithms for designing optimum quantizers of two or more dimensions have been presented

Manuscript received Feb. 19, 1980; revised March 12, 1981. This work was supported by the Air Force Office of Scientific Research under Grant AFOSR 78-3605.

K. D. Rines was with the School of Electrical Engineering, Purdue University, West Lafayette, IN. He is now with The Analytic Sciences Corporation, McLean Operation, 8301 Greensboro Drive, Suite 1200, McLean, VA, 22102.

N. C. Gallagher, Jr., is with the School of Electrical Engineering, Purdue University, West Lafayette, IN 47907.

0018-9448/82/0300-0232$00.75 ©1982 IEEE