$n \to \infty$.) Naturally, we define $|K|/\frac{2^n}{\binom{n/2}{R}}$ as the density of K. Now we can define $\mu_{os}^*(R)$ as the counterpart of $\mu^*(R)$.

The authors of [2] proved (in a somewhat different formulation) that for all fixed R there is a constant $c(R)$ such that $\mu_{os}^*(R) \leq c(R)$. The constant $c(R)$ was not computed explicitly, but a careful reading reveals that it should be at least a^R for some constant $a > 1$. Repeating the proof of Theorem 1.2 for one-sided codes (a minor modification is needed) we can prove the statement of Theorem 1.2 for $\mu_{os}^*(R)$ and (consequently) improve the bound on $\mu_{os}^*(R)$ to order $R \log R$.

Theorem 4.1: Given a pair of positive integers $R > R_1 \geq 1$

$$\mu_{os}^*(R) \leq \frac{y^{R_1}\left(\frac{y}{y-1}\right)^{R-R_1}\binom{R}{R_1}^{-1} x \mu_{os}^*(R_1)}{1 - e^{-x}y^R}$$

holds for any pair of positive constants x and y satisfying $y > 1$ and $1 - e^{-x}y^R > 0$.

Then we have the following.

Corollary 4.2: For all $R \geq 3$

$$\mu_{os}^*(R) \leq e(R\log R + \log R + \log\log R + 1)\mu_{os}^*(1).$$

Notice that here we do not know whether $\mu_{os}^*(1) = 1$.

The minor modification we need in the proof of Theorem 4.1 is due to the fact that the one-side balls have different volumes. It is not hard, however, to overcome this obstacle. By the binomial distribution, the fraction of vertices of $\mathbb{F}_2^n$ with weights more than $\frac{n}{2} + 10R\sqrt{n \log n}$ or less than $\frac{n}{2} - 10R\sqrt{n \log n}$ is $o(1/n^R)$ (10 can be replaced by a smaller number), so it suffices to focus on the vertices with weights between $\frac{n}{2} - 10R\sqrt{n \log n}$ and $\frac{n}{2} + 10R\sqrt{n \log n}$. The one-sided balls centered at these vertices all have volume approximately $\binom{n/2}{R}$. We leave out the details which might serve as an exercise.

ACKNOWLEDGMENT

The authors wish to thank the Associate Editor S. Litsyn for several useful comments.

REFERENCES

[1] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein, *Covering Codes*, ser. North-Holland Mathematical Library. Amsterdam, The Netherlands: North-Holland, 1997, vol. 54.

[2] J. Cooper, R. Ellis, and A. Kahng, "Asymmetric binary covering codes," *J. Comb. Theory Ser. A*, vol. 100, no. 2, pp. 232–249, 2002.

[3] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

Near-Ellipsoidal Voronoi Coding

Stéphane Ragot, *Student Member, IEEE*, Minjie Xie, and Roch Lefebvre, *Member, IEEE*

***Abstract*—In this correspondence, we consider a special case of Voronoi coding, where a lattice Λ in $\mathbb{R}^n$ is shaped (or truncated) using a lattice $\Lambda' = \{(m_1x_1, \ldots, m_nx_n) | (x_1, \ldots, x_n) \in \Lambda\}$ for a fixed $\underline{m} = (m_1, \ldots, m_n) \in (\mathbb{N} \setminus \{0, 1\})^n$. Using this technique, the shaping boundary is near-ellipsoidal. It is shown that the resulting codes can be indexed by standard Voronoi indexing algorithms plus a conditional modification step, as far as Λ' is a sublattice of Λ. We derive the underlying conditions on $\underline{m}$ and present generic near-ellipsoidal Voronoi indexing algorithms. Examples of constraints on $\underline{m}$ and conditional modification are provided for the lattices A_2, D_n $(n \geq 2)$ and $2D_n^+$ $(n$ even $\geq 4)$.**

***Index Terms*—Lattice, lattice codes, lattice indexing, Voronoi coding.**

I. INTRODUCTION

We address the problem of designing (near-)ellipsoidal lattice codes with fast indexing algorithms. The motivation for this work lies in wide-band speech coding. More specifically, we are interested in designing low-complexity high-dimensional algebraic spectrum coding based on a Gaussian mixture model [6], which implies construction of codes to quantize correlated Gaussian vector sources.

Lattices, which are extensively studied in [10], are linear discrete sets of points. Without loss of generality, we will consider here only lattices defined in $\mathbb{R}^n$. A lattice code is defined by selecting a finite subset of a lattice. Lattice codes find important applications, such as coded modulation and vector quantization. They are known to yield potential good performance–complexity tradeoffs and to be asymptotically good in certain conditions.

Given a lattice, two important steps are required to implement a lattice code.

1) Shape the lattice properly (i.e., define the support region of the lattice code) and design the indexing algorithms to label codevectors.

2) Design a procedure to find the closest lattice point *within the code*, that is, the nearest codevector to any arbitrary point.

In this correspondence we deal only with the lattice shaping and indexing problem. This problem is important, since an optimized lattice shaping may bring significant performance gains compared to a baseline shaping (e.g., scalar quantization in source coding applications) [2], [13]. To be more specific, we will focus hereafter on lattice codes defined by ellipsoidal truncation. As mentioned earlier, this restriction is motivated by the need in certain applications to quantize correlated Gaussian vector sources. Other shaping techniques, yielding, for instance, codes defined *on* or *inside* spherical [11], [12] or pyramidal [7], [14], [15] shapes, are not considered.

Manuscript received June 2, 2001; revised December 20, 2002. This work was supported by the Natural Sciences and Engineering Research Council of Canada and VoiceAge Corp. The material in this correspondence was presented in part at the 7th Canadian Workshop on Information Theory, Vancouver, BC, Canada, June 2001.

S. Ragot and R. Lefebvre are with the Department of Electrical and Computer Engineering, University of Sherbrooke, Sherbrooke, QC J1K 2R1 Canada (e-mail: ragot@hermes.usherb.ca; roch.lefebvre@courrier.usherb.ca)

M. Xie was with the University of Sherbrooke. He is now with Polycom, Inc., Burlington, MA 01803 USA (e-mail: mxie@AUSTIN.Polycom.com).

Communicated by R. Zamir, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2003.813484

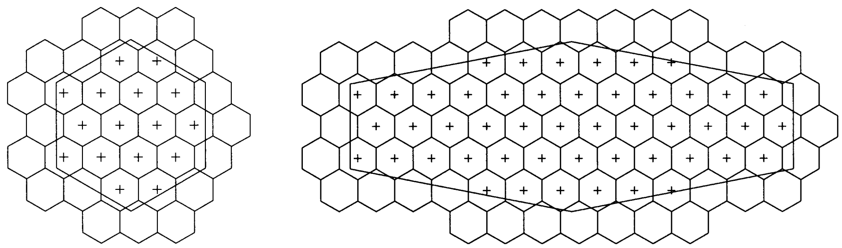

Fig. 1. Example of Voronoi codes for $\Lambda = A_2$. (a) Voronoi code $R = 2$. (b) Near-ellipsoidal Voronoi code $\underline{m} = (12, 4)$.

Ellipsoidal lattice codes have been already proposed in the literature [4], [7]–[9]. However, the lattice $\mathbb{Z}$ used in [7] provides no granular gain. The technique of [8], [9], which generalizes spherical shaping by enumerating lattice points on ellipsoidal shells, has a complexity increasing exponentially with the lattice dimension and the bit rate of the code. To avoid these limitations, we present a special case of Voronoi coding [1], [2] which yields *near*-ellipsoidal lattice codes with fast indexing algorithms and minimal storage. Some early versions of the algorithms presented hereafter were introduced in [4], [5].

The correspondence is organized as follows. We proceed with the notations and basic definitions, and review Voronoi coding in Section II. Near-ellipsoidal Voronoi codes are defined and studied in Section III. As we shall see later, the lattice shaping introduced here is constrained and the related indexing algorithms comprise a step, called *condition modification*, which may be tailored for each lattice. We will use, in particular, the lattices A_2, D_n $(n \geq 2)$ and $2D_n^+$ $(n \text{ even} \geq 4)$ for the purpose of illustrating the related constraints and conditional modification. The conclusions come in Section IV.

II. Preliminaries

A. Notations and Basic Definitions

The notation $\mathbb{N} \setminus \{0, 1\}$ refers to the set of integers greater than 1.

The scalar operators $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ round any input in $\mathbb{R}$ to the nearest integer in $\mathbb{Z}$ toward $-\infty$ and $+\infty$, respectively.

Vectors are denoted with a bar in subscript, while scalars are denoted in italics. The row convention is used for vectors. For $\underline{x}$ in $\mathbb{R}^n$, the ith element of $\underline{x}$ is denoted x_i so that $\underline{x} = (x_1, \ldots, x_n)$.

To describe later Voronoi indexing algorithms, we need to introduce the vector operators $\mathbf{mod}$, $\mathbf{mult}$, and $\mathbf{div}$, which denote the element-wise modulo, multiplication, and division of two vector operands, respectively. For $\underline{x}$ and $\underline{y}$ in $\mathbb{R}^n$, $\underline{j}$ in $\mathbb{Z}^n$ and $\underline{m}$ in $(\mathbb{N} \setminus \{0, 1\})^n$, we have

$$\mathbf{mod}(\underline{j}, \underline{m}) = (j_1 (\bmod\ m_1), \ldots, j_n (\bmod\ m_n)) \tag{1}$$

$$\mathbf{mult}(\underline{x}, \underline{y}) = (x_1 y_1, \ldots, x_n y_n) \tag{2}$$

$$\mathbf{div}(\underline{x}, \underline{y}) = (x_1 / y_1, \ldots, x_n / y_n) \tag{3}$$

where mod is the scalar modulo operator.

In this work, we will consider only full-rank lattices in $\mathbb{R}^n$. In the general case, a lattice in $\mathbb{R}^n$ is denoted Λ and is defined as

$$\Lambda = \{k_1 \underline{v}_1 + \cdots + k_n \underline{v}_n | \underline{k} \in \mathbb{Z}^n\} \tag{4}$$

where $\{\underline{v}_i\}_{1 \leq i \leq n}$ is a set of linearly independent vectors in $\mathbb{R}^n$. These vectors, when stacked on top of each other, form a matrix, called a generator matrix $M(\Lambda)$ of Λ. It is important to note that we use the row convention for $M(\Lambda)$. In other words, if $\underline{k}$ is in $\mathbb{Z}^n$, $\underline{x} = \underline{k} M(\Lambda)$ generates a point in Λ and $\underline{x} M(\Lambda)^{-1}$ retrieves the related basis expansion $\underline{k}$. The lattice can then be written as

$$\Lambda = \{\underline{k} M(\Lambda) | \underline{k} \in \mathbb{Z}^n\}. \tag{5}$$

In a lattice, all Voronoi regions are congruent [10], and we can consider only the region related to the origin, denoted $V(\Lambda)$.

We will use, more specifically, for Λ the lattices A_2, D_n, and $2D_n^+$. These lattices are all defined and specified by a generator matrix in [10]. However, for the sake of completeness and clarity, the generator matrices used herein are all defined in the Appendix. Note that we use only lower triangular generator matrices, which will be an important property to simplify later the near-ellipsoidal Voronoi indexing algorithms.

Following [3], if Λ' is a sublattice of Λ, $|\Lambda/\Lambda'|$ refers to the order of the lattice partition Λ/Λ'.

B. Voronoi Coding

Given a lattice Λ in $\mathbb{R}^n$, a Voronoi code [1] can be defined as $C = \Lambda \cap \left(2^R\, V(\Lambda) + \underline{a}\right)$, where R is an integer greater than 1 and $\underline{a}$ is an appropriate offset in $\mathbb{R}^n$ set to fix ties (i.e., to ensure no point of Λ is on the boundary of the shaping region). The code size of C is $|\Lambda / 2^R \Lambda| = 2^{nR}$. This definition differs slightly from the original. In [1], the Voronoi code C is translated by $-\underline{a}$ and is usually zero mean; this difference has, however, no influence on the Voronoi indexing algorithms. The extension of Voronoi coding to lattice shaping by geometrically similar sublattices [2] is not considered here.

With this definition, a Voronoi code can be viewed directly as the truncation of Λ by the Voronoi region $V(\Lambda)$ scaled by 2^R and translated by $\underline{a}$, as illustrated in Fig. 1(a). It is also possible to interpret the shaping region as $V(2^R \Lambda) + \underline{a}$, where $2^R \Lambda$ appears to be a shaping lattice [2]. The advantage of this point of view is that C can then be viewed as the set of (minimum-norm) coset leaders of the partition $\Lambda / 2^R \Lambda$ [2]. Voronoi shaping may also be interpreted as a modulo operation with lattice operands [10].

Voronoi shaping yields index-optimized lattice codes since there exist some elegant indexing algorithms for C which rely on lattice decoding and modular arithmetics [1].

III. Near-Ellipsoidal Voronoi Codes

A. Definition (With Restrictions on $\underline{m}$)

Given a lattice Λ in $\mathbb{R}^n$, a near-ellipsoidal Voronoi code is defined as $C = \Lambda \cap (V(\Lambda') + \underline{a})$, where

$$\Lambda' = \{(m_1 x_1, \ldots, m_n x_n) | \underline{x} \in \Lambda\}. \tag{6}$$

The vector $\underline{m}$ in $(\mathbb{N} \setminus \{0, 1\})^n$ is constrained here so that Λ' is a sublattice of Λ. The offset $\underline{a}$ in $\mathbb{R}^n$ is chosen to fix ties. The region $V(\Lambda')$, which is the Voronoi region of Λ', corresponds to the region $V(\Lambda)$ scaled in each dimension according to the elements of $\underline{m}$. This lattice shaping is illustrated in Fig. 1(b).

Property: The size of C is $\prod_{i=1}^{n} m_i$.

Proof: The proof follows directly from [2]. If Λ' is a sublattice of Λ, the size of C is $|\Lambda/\Lambda'|$. It is easy to verify from the definition of Λ' that

$$\Lambda = \bigcup_{\substack{\underline{k} \in \mathbb{Z}^n \text{ such that} \\ 0 \le k_i < m_i \text{ for } 1 \le i \le n}} \Lambda' + \underline{k} M(\Lambda). \tag{7}$$

Therefore there are $\prod_{i=1}^{n} m_i$ cosets of Λ' in Λ, i.e.,

$$|\Lambda/\Lambda'| = \prod_{i=1}^{n} m_i. \qquad \square$$

B. Admissible Voronoi Modulos

We address here the problem of finding all $\underline{m}$ in $(\mathbb{N} \setminus \{0, 1\})^n$ so that the lattice Λ', defined in (6), is a sublattice of Λ. The solution we propose is based on the following theorem.

Theorem: Λ' is a sublattice of Λ if and only if

$$Q = M(\Lambda)\mathrm{diag}\,(m_1, \ldots, m_n) M(\Lambda)^{-1}$$

is a matrix of integers.

Proof: The proof is simple. We assume first that Λ' is a sublattice of Λ. We start by observing that the rows of $M(\Lambda)$ are in Λ, and the rows of $M(\Lambda)\mathrm{diag}\,(m_1, \ldots, m_n)$ are in Λ'. Since Λ' is a sublattice of Λ, the rows of $M(\Lambda)\mathrm{diag}\,(m_1, \ldots, m_n)$ are also in Λ. Each row of $M(\Lambda)\mathrm{diag}\,(m_1, \ldots, m_n)$ multiplied by $M(\Lambda)^{-1}$ will thus give the expansion in a basis of Λ in terms of integer coordinates. Consequently, $Q = M(\Lambda)\mathrm{diag}\,(m_1, \ldots, m_n)M(\Lambda)^{-1}$ is a matrix of integers.

To prove the converse part, we now assume that Q is a matrix of integers. A point $\underline{\lambda}'$ in Λ' may be expanded as $\underline{\lambda}' = \underline{k}M(\Lambda')$. This can be also written as $\underline{\lambda}' = \underline{k}M(\Lambda)\mathrm{diag}\,(m_1, \ldots, m_n)$ from the definition of Λ'. Using the assumption, $\underline{\lambda}' M(\Lambda)^{-1}$ will be a vector of integers, which implies that $\underline{\lambda}'$ is obtained by an integral combination of a basis of Λ. Thus, Λ' is a sublattice of Λ. $\square$

By using the above theorem, we can derive the conditions on $\underline{m}$ for Λ' to be a sublattice of Λ. Note that the criterion is invariant with respect to any scaled integer orthogonal transformation of Λ: if $M(\Lambda)$ is a generator matrix of Λ, any matrix of the form $cTM(\Lambda)$ specifying the lattice $c\Lambda$, where c is a scaling factor and T an integer orthogonal matrix $(\det T = \pm 1)$, yields identical constraints on $\underline{m}$. However, in general,[1] two generator matrices $M(\Lambda_1)$ and $M(\Lambda_2)$ of two equivalent lattices Λ_1 and Λ_2 yield different conditions on $\underline{m}$.

Corollary: The admissible $\underline{m}$ are points from $(\mathbb{N} \setminus \{0, 1\})^n \cap A$, where A is an integer lattice which is a function of Λ.

Proof: We denote by A the set of vectors $\underline{m}$ in $\mathbb{R}^n$ such that

$$Q = M(\Lambda)\mathrm{diag}\,(m_1, \ldots, m_n)M(\Lambda)^{-1}$$

[1]For instance, the constraints on $\boldsymbol{m}$ differ for the two equivalent lattices $\boldsymbol{\Lambda}_1 = \mathbb{Z}^2$ and $\boldsymbol{\Lambda}_2 = \boldsymbol{D}_2$.

TABLE I
ADMISSIBLE VORONOI MODULOS $\boldsymbol{m}$ (BY DEFINITION $\boldsymbol{m}$ IS ALSO CONSTRAINED TO BE IN $(\mathbb{N} \setminus \{0, 1\})^n$

Lattice	Constraints on $\underline{m}$
A_2	m_1 and m_2 have the same parity (i.e. $\underline{m} \in D_2$)
D_n $(n \ge 2)$	$m_1, \cdots, m_n$ all share the same parity (i.e. $\underline{m} \in 2D_n^*$)
$2D_n^+$ (n even ≥ 4)	$m_1, \cdots, m_n$ all share the same parity and $\sum_{i=1}^{n} m_i$ is a multiple of 4 $(\underline{m} \in 2D_n^+)$

is an integer matrix. Without loss of generality, we force $M(\Lambda)$ to be lower triangular. Then it is easy to show that Q is lower triangular and has diagonal elements $m_1, \ldots, m_n$, i.e.,

$$Q = \begin{bmatrix} m_1 & & & \\ \alpha_{2,1} & m_2 & & \\ \vdots & \ddots & \ddots & \\ \alpha_{n,1} & \cdots & \alpha_{n,n-1} & m_n \end{bmatrix}. \tag{8}$$

It then follows that if $\underline{m} \in A$, $\underline{m}$ is an integer vector. Moreover, it is easy to check that the set A is nonempty and has an additive group structure. We can then conclude that the set A is an integer lattice, i.e., a sublattice of $\mathbb{Z}^n$. The proof is complete by noting that admissible vectors $\underline{m}$ are defined in $(\mathbb{N} \setminus \{0, 1\})^n \cap A$. $\square$

C. Examples of Constraints on $\underline{m}$

For the lattice A_2, the condition becomes

$$Q = M(A_2)\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \quad M(A_2)^{-1} = \begin{bmatrix} m_1 & 0 \\ \dfrac{m_1 - m_2}{2} & m_2 \end{bmatrix} \tag{9}$$

where $M(A_2)$ is set as in the Appendix. We obtain that m_1 and m_2 must have the same parity (either even or odd). Using the generator matrices $M(D_n)$ and $M(2D_n^+)$ defined in the Appendix, we obtain for D_n that $m_1, m_2, \cdots, m_n$ must have the same parity; the same constraint applies for $2D_n^+$, but $m_1 + \cdots + m_n$ must also be a multiple of 4. These results are summarized in Table I.

D. Indexing Algorithms: Generic Framework

Voronoi coding, as introduced in [1], is an elegant lattice shaping technique yielding generic indexing algorithms. It is desirable for near-ellipsoidal Voronoi codes to keep this attribute, and to minimize algorithmic changes to the original framework of [1]. We will show here that, as long as $\underline{m}$ is an admissible Voronoi modulo, a near-ellipsoidal Voronoi code C specified in a lattice Λ by $\underline{m}$ and $\underline{a}$ can be indexed using the algorithms presented in Fig. 2. Note that the index $\underline{k}$ in Fig. 2 is not scalar,[2] but rather a vector of integers which satisfy $0 \le k_i < m_i$ for $i \in \{1, \ldots, n\}$. Consequently, there are $\prod_{i=1}^{n} m_i$ possible values for $\underline{k}$.

[2]To form a scalar index $\boldsymbol{K}$, the elements of a near-ellipsoidal Voronoi index $\underline{\boldsymbol{k}}$ can be easily multiplexed (e.g., $\boldsymbol{K} = \boldsymbol{k}_1 \prod_{i=2}^{n} \boldsymbol{m}_i + \boldsymbol{k}_2 \prod_{i=3}^{n} \boldsymbol{m}_i + \cdots + \boldsymbol{k}_{n-1}\boldsymbol{m}_n + \boldsymbol{k}_n$). The size of $\boldsymbol{K}$ is $\lceil \log_2 (\prod_{i=1}^{n} \boldsymbol{m}_i) \rceil$ bits.

Encoding algorithm: codevector $\underline{\lambda} \in \Lambda \to$ index $\underline{k}$

1. Compute $\underline{j} = \underline{\lambda} M(\Lambda)^{-1}$
2. (a) Compute $l_2, \cdots, l_n$ from $\underline{j}$, $\underline{m}$ and Q } conditional modification step
   (b) Compute $\underline{j}'$ from $\underline{j}$, $l_2, \cdots, l_n$ and Q
3. Compute $\underline{k} = \mathbf{mod}(\underline{j}', \underline{m})$

Decoding algorithm: index $\underline{k} \to$ codevector $\underline{\lambda} \in \Lambda$

1. Compute $\underline{x} = \underline{k} M(\Lambda)$
2. Compute $\underline{y} = \mathbf{div}(\underline{x} - \underline{a}, \underline{m})$
3. Find the nearest neighbor $\underline{z}$ of $\underline{y}$ in Λ
4. Compute $\underline{\lambda} = \underline{x} - \mathbf{mult}(\underline{m}, \underline{z})$

Fig. 2. Indexing algorithms for a near-ellipsoidal Voronoi code.

1) Decoding Algorithm: We will begin with the decoding algorithm of Fig. 2 which maps an index $\underline{k}$ into a code vector $\underline{\lambda}$. This algorithm is a straightforward generalization of [1]. The detailed steps 1–4 are summarized below as follows:

$$\underline{\lambda} = \underline{x} - Q_{\Lambda'}(\underline{x} - \underline{a}) \tag{10}$$

where $Q_{\Lambda'}$ denotes the nearest neighbor search operation in the lattice Λ' and $\underline{x} = \underline{k} M(\Lambda)$. It follows from (10) that all $\underline{\lambda}$ generated by decoding will be in the near-ellipsoidal Voronoi code C. It is now important to verify that two different indexes $\underline{k}$ and $\underline{k}'$ will generate two different code vectors in C. This follows directly from the fact that (10) may be interpreted as a lattice-modulo operation applied on Λ. To conclude this part, it can be verified[3] that for the decoding to produce points in Λ, Λ' has to be a sublattice of Λ.

2) Encoding Algorithm: Conversely, the encoding algorithm of Fig. 2 maps a near-ellipsoidal Voronoi codevector $\underline{\lambda}$ into an index $\underline{k}$. We describe here the principle of the conditional modification step and show why it is helpful for the generator matrix $M(\Lambda)$ to be specified as a lower triangular matrix.

Given $\underline{\lambda}$ in a near-ellipsoidal Voronoi code C specified in Λ by $\underline{m}$ and $\underline{a}$, we can interpret $\underline{\lambda}$ as a codevector generated by decoding using an index $\underline{k}$. Using steps 1 and 4 of the decoding algorithm, we can state that there exists an index $\underline{k}$ and $\underline{z} \in \Lambda$ such that

$$\underline{\lambda} = \underline{k} M(\Lambda) - \mathbf{mult}(\underline{m}, \underline{z}). \tag{11}$$

Since $\underline{z} \in \Lambda$, there exists also $\underline{l}$ such that $\underline{z} = \underline{l} M(\Lambda)$. Then (11) becomes

$$\underline{\lambda} = \underline{k} M(\Lambda) - \underline{l} M(\Lambda) \mathrm{diag}\,(m_1, \ldots, m_n). \tag{12}$$

Consequently, the intermediate index $\underline{j} = \underline{\lambda} M(\Lambda)^{-1}$ in step 1 of the encoding can be expanded as

$$\underline{j} = \underline{k} - \underline{l} Q \tag{13}$$

where

$$Q = M(\Lambda) \mathrm{diag}\,(m_1, \ldots, m_n) M(\Lambda)^{-1}. \tag{14}$$

To retrieve automatically the exact index $\underline{k}$ from $\underline{\lambda}$, we need to eliminate somehow the extra term $\underline{l} Q$. This is done in steps 2 and 3 of the encoding algorithm by applying a conditional modification and an element-wise modulo. The principle of the conditional modification is explained next.

If $M(\Lambda)$ is specified as a lower triangular matrix, the matrix Q is lower triangular and has diagonal elements $m_1, \ldots, m_n$, as described in (8). Moreover, if $\underline{m}$ is constrained such that Λ' is a sublattice of Λ, Q has integer elements. Using (8), we can write (13) as a system of equations

$$\begin{cases} j_1 = k_1 - (m_1 l_1 + \alpha_{2,1} l_2 + \cdots + \alpha_{n,1} l_n) \\ j_2 = k_2 - (m_2 l_2 + \alpha_{3,2} l_3 + \cdots + \alpha_{n,2} l_n) \\ \vdots \\ j_{n-1} = k_{n-1} - (m_{n-1} l_{n-1} + \alpha_{n,n-1} l_n) \\ j_n = k_n - m_n l_n. \end{cases} \tag{15}$$

The elements of $\underline{k}$ can then be computed recursively from k_n to k_1.

- Equation (15) gives $k_n = j_n + m_n l_n$. Therefore,

  $$k_n = j_n (\mathrm{mod}\ m_n)$$

  which satisfies $0 \le k_n < m_n$.
- Equation (15) also gives $l_n = (k_n - j_n)/m_n$. From the already calculated k_n, we obtain

  $$l_n = (j_n (\mathrm{mod}\ m_n) - j_n)/m_n = -\lfloor j_n / m_n \rfloor.$$

- Equation (15) gives

  $$k_{n-1} = j_{n-1} + m_{n-1} l_{n-1} + \alpha_{n,n-1} l_n$$

  which can be rearranged as

  $$k_{n-1} = j'_{n-1} + m_{n-1} l_{n-1}$$

  where $j'_{n-1} = j_{n-1} + \alpha_{n,n-1} l_n$ can be evaluated from the already calculated l_n. Therefore,

  $$k_{n-1} = j'_{n-1} (\mathrm{mod}\ m_{n-1})$$

  which satisfies $0 \le k_{n-1} < m_{n-1}$.

$$\vdots$$

[3] See step 4 of the decoding algorithm: the term $\mathbf{mult}(\underline{\boldsymbol{m}}, \underline{\boldsymbol{z}})$ must be in $\boldsymbol{\Lambda}$ for $\underline{\boldsymbol{\lambda}}$ to also be in $\boldsymbol{\Lambda}$.

This recursive procedure eventually results in the following equations:

$$\begin{cases} l_n = -\lfloor j_n/m_n \rfloor \\ l_{n-1} = -\lfloor (j_{n-1} + \alpha_{n,\,n-1} l_n)/m_{n-1} \rfloor \\ \vdots \\ l_2 = -\lfloor (j_2 + (\alpha_{3,\,2} l_3 + \cdots + \alpha_{n,\,2} l_n)/m_2 \rfloor \end{cases} \tag{16}$$

and

$$\begin{cases} j'_n = j_n \\ j'_{n-1} = j_{n-1} + \alpha_{n,\,n-1} l_n \\ \vdots \\ j'_2 = j_2 + (\alpha_{3,\,2} l_3 + \cdots + \alpha_{n,\,2} l_n) \\ j'_1 = j_1 + (\alpha_{2,\,1} l_2 + \cdots + \alpha_{n,\,1} l_n). \end{cases} \tag{17}$$

The elements of $\underline{k}$ are given by

$$k_i = j'_i (\mathrm{mod}\, m_i), \qquad \text{for } 1 \le i \le n. \tag{18}$$

The conditional modification step of Fig. 2 consists of (16) and (17).

Note that $\underline{j}'$ is defined here as $j'_n = j_n$ and

$$j'_i = j_i + \alpha_{i+1,\,i} l_{i+1} + \cdots + \alpha_{n,\,i} l_n, \qquad \text{for } 1 \le i < n.$$

The quantity l_1 is not calculated in (16), because it is not needed in (17) and (18). If $\underline{j}'$ was defined as $j'_n = j_n + m_n l_n$ and

$$j'_i = j_i + m_i l_i + \alpha_{i+1,\,i} l_{i+1} + \cdots + \alpha_{n,\,i} l_n, \qquad \text{for } 1 \le i < n$$

i.e., $\underline{j}' = \underline{j} + \underline{l}Q$, the quantity

$$l_1 = -\lfloor (j_1 + (\alpha_{2,\,1} l_2 + \cdots + \alpha_{n,\,1} l_n)/m_1 \rfloor$$

would have to be computed, but the index $\underline{k}$ calculated in (18) would be identical.

E. Examples of Conditional Modification

We present here several examples to illustrate the conditional modification, i.e., (16) and (17), for $\Lambda = A_2, D_n\ (n \ge 2)$ and $2D_n^+$ (n even ≥ 4). Another example—for the lattice Λ_{16}—can be found in [16]. For the conditional modification to work correctly, we assume that $\underline{m}$ satisfies the constraint such that the matrix Q defined in (14) and specified in (8) is a matrix of integers.

1) Conditional Modification for A_2: For A_2, the matrix Q is computed in (9). The conditional modification is then given by

i.	Compute $l_2 = -\lfloor j_2/m_2 \rfloor$
ii.	Set $j'_1 = j_1 + (m_1 - m_2) l_2/2$ and $j'_2 = j_2$

2) Conditional Modification for D_n: For D_n, the matrix Q is given by

$$Q = \begin{bmatrix} m_1 & & & \\ \frac{m_1 - m_2}{2} & m_2 & & \\ \vdots & & \ddots & \\ \frac{m_1 - m_n}{2} & & & m_n \end{bmatrix} \tag{19}$$

where the missing elements in Q are zeros. We obtain the following sequence of operations for the conditional modification:

i.	Compute $l_i = -\lfloor j_i/m_i \rfloor$ for $i \in \{2, \ldots, n\}$
ii.	Set $j'_1 = j_1 + \sum_{i=2}^{n} (m_1 - m_i) l_i/2$
	Set $j'_i = j_i$ for $i \in \{2, \ldots, n\}$

3) Conditional Modification for $2D_n^+$: For $2D_n^+$

$$Q = \begin{bmatrix} m_1 & & & & \\ \frac{m_1 - m_2}{2} & m_2 & & & \\ \vdots & & \ddots & & \\ \frac{m_1 - m_{n-1}}{2} & & & m_{n-1} & \\ s & \frac{m_2 - m_n}{2} & \cdots & \frac{m_{n-1} - m_n}{2} & m_n \end{bmatrix} \tag{20}$$

where $s = (m_1 - m_2 - \cdots - m_{n-1} - (n-3)m_n)/4$ and the missing elements in Q are zeros . The conditional modification can be implemented as follows:

i.	Compute $l_n = -\lfloor j_n/m_n \rfloor$
	Set $j'_n = j_n$
ii.	For $i \in \{2, \ldots, n-1\}$:
	—compute $l_i = -\left\lfloor \left(j_i + \frac{m_i - m_n}{2} l_n \right) \Big/ m_i \right\rfloor$
	—set $j'_i = j_i + (m_i - m_n) l_n/2$
iii.	Set $j'_1 = j_1 + \sum_{i=2}^{n-1} (m_1 - m_i) l_i/2 + s l_n$

IV. Conclusion

In this correspondence, we introduced a special case of Voronoi coding where a near-ellipsoidal Voronoi code is defined in a lattice Λ by a modulo vector $\underline{m}$ and an offset $\underline{a}$. We showed how standard Voronoi indexing could be generalized to encode the related codes using the algorithms of Fig. 2. The main contribution of this correspondence concerns the definition of constraints on $\underline{m}$ and the derivation of a conditional modification step, which is summarized in (16) and (17). The matrix Q, given in (14) and (8), was shown to play a key role in determining the admissible modulo $\underline{m}$ and the design of the conditional modification step. Note that this step is not required when the modulo vector $\underline{m}$ has identical components, i.e., in the case of near-spherical Voronoi coding.

The proposed codes have several advantages. They generalize the standard Voronoi codes of [1]. Their indexing algorithms require minimal data storage (only the generator matrix of Λ, the offset $\underline{a}$, and the modulo vector $\underline{m}$). Furthermore, the complexity of these algorithms is essentially given by the complexity involved in the closest lattice point search and in the conditional modification.

Nonetheless, these codes also have some limitations. The constraints on admissible scalings are lattice specific; therefore, the optimization of $\underline{m}$ for a given Gaussian vector source depends on Λ. Besides, for the code shape to be really near-ellipsoidal, the shape of a Voronoi region $V(\Lambda)$ of Λ should be as spherical as possible.[4] Consequently, the lattice $\mathbb{Z}$ would not be useful. Last but not least, the nearest neighbor search in Voronoi codes is tricky and is usually solved at low complexity with suboptimal strategies.

Note that we did not consider here how to optimize the offset $\underline{a}$ to fix ties—this problem was already addressed in [1], [2]. An application to wide-band speech coding can be found in [6] where the lattices D_{16}, $2D_{16}^+$, and Λ_{16} were used.

[4]This requirement is also true for the near-spherical Voronoi codes of [1], [2].

APPENDIX
GENERATOR MATRICES FOR THE LATTICES USED HEREIN

The lattices used herein are full rank, therefore, the following generator matrices are square matrices:

$$M(A_2) = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}. \tag{21}$$

For $n \geq 2$

$$M(D_n) = \begin{bmatrix} 2 & & & \\ 1 & 1 & & \\ \vdots & & \ddots & \\ 1 & & & 1 \end{bmatrix}. \tag{22}$$

For n even ≥ 4

$$M(2D_n^+) = \begin{bmatrix} 4 & & & & \\ 2 & 2 & & & \\ \vdots & & \ddots & & \\ 2 & & & 2 & \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}. \tag{23}$$

The missing coefficients in $M(D_n)$ and $M(2D_n^+)$ correspond to zeros.

ACKNOWLEDGMENT

The authors would like to acknowledge the contribution of Prof. J.-P. Adoul in the early development of near-ellipsoidal Voronoi codes. They are grateful to Prof. F. R. Kschischang for a valuable comment during the CWIT'2001 Workshop. They also wish to thank an anonymous reviewer for helpful suggestions to improve the presentation of the text—in particular, the derivation of the conditional modification step—as well as for a question which motivated the Corollary.

REFERENCES

[1] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 820–824, Nov. 1983.

[2] G. D. Forney, "Multidimensional constellations. II. Voronoi constellations," *IEEE Trans. Select. Areas Commun.*, vol. 7, pp. 941–958, Aug. 1989.

[3] ——, "Coset codes. I. Introduction and geometrical classification," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1123–1151, Sept. 1988.

[4] M. Xie, "Quantification vectorielle algébrique et codage de parole en bande élargie," Ph.D. dissertation, Univ. Sherbrooke, Sherbrooke, QC, Canada, Feb. 1996.

[5] S. Ragot, M. Xie, and R. Lefebvre, "Near-ellipsoidal lattice quantization by generalized Voronoi shaping," in *Proc. 7th IEEE Canadian Workshop on Information Theory (CWIT)*, Vancouver, BC, Canada, June 2001.

[6] S. Ragot, H. Lahdili, and R. Lefebvre, "Wideband LSF quantization by generalized Voronoi codes," in *Proc. Eurospeech*, Aalborg, Denmark, Sept. 2001, pp. 2319–2322.

[7] T. R. Fischer, "Geometric source coding and vector quantization," *IEEE Trans. Inform. Theory*, vol. 35, pp. 137–145, Jan. 1989.

[8] M. Barlaud, P. Solé, J. M. Moureaux, M. Antonini, and P. Gauthier, "Elliptical codebook for lattice vector quantization," in *Proc. IEEE Int. Conf. Acousitcs, Speech, and Signal Processing*, vol. 5, Apr. 1993, pp. 590–593.

[9] J. M. Moureaux, M. Antonini, and M. Barlaud, "Counting lattice points on ellipsoids: Application to image coding," *Electron. Lett.*, vol. 31, no. 15, pp. 1224–1225, July 1995.

[10] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. New York: Springer-Verlag, 1999.

[11] J.-P. Adoul and M. Barth, "Nearest neighbor algorithm for spherical codes from the Leech lattice," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1188–1202, Sept. 1988.

[12] M. Xie and J.-P. Adoul, "Embedded algebraic vector quantizer (EAVQ) with application to wide-band speech coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, Atlanta, GA, May 7–10, 1996, pp. 240–243.

[13] T. D. Lookabaugh and R. M. Gray, "High-resolution quantization theory and the vector quantizer advantage," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1020–1033, Sept. 1989.

[14] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 568–583, July 1986.

[15] M. Barlaud, P. Solé, T. Gaidon, M. Antonini, and P. Mathieu, "Pyramidal lattice vector quantization for multiscale image coding," *IEEE Trans. Image Processing*, vol. 3, pp. 367–381, July 1994.

[16] S. Ragot, "New techniques of algebraic vector quantization based on Voronoi coding—Application to AMR-WB+ coding," Ph.D. dissertation (in French), Univ. Sherbrooke, Sherbrooke, QC, Canada, May 2003.