(Article begins on next page)

Voronoi Regions for Binary Linear Block Codes

Erik Agrell

To appear in IEEE Transactions on Information Theory

Submitted July 1994

Abstract—The Voronoi regions of a block code govern many aspects of the code's performance on a Gaussian channel, and they are fundamental instruments in, for example, error probability analysis and soft decision decoding. This paper presents an efficient method to find the boundaries of the Voronoi regions for an arbitrary binary linear block code. Two theoretical results together lead to the Voronoi regions. First, it is shown that the question of Voronoi neighborship can be reduced into testing a simpler relation, called Gabriel neighborship. Second, a fast method to recognize Gabriel neighbors is proposed. These results are finally employed to describe the Voronoi regions for the Golay codes and several BCH codes, including Hamming codes.

Index Terms—Linear block codes, Euclidean codes, Voronoi regions, soft decision decoding.

The author is with the Department of Information Theory, Chalmers University of Technology, 41296 Göteborg, Sweden.

I. Introduction

The Voronoi region appears in all applications where analog input vectors are represented as elements of a discrete point set, provided that the Euclidean distance is the entity to be minimized. The Euclidean distance is an appropriate distortion measure for a wide class of detection rules, such as MMSE and, under some assumptions, MAP and ML. Especially, the Gaussian channel is strongly connected to the Euclidean distance criterion. If this distortion measure is used by a soft decision decoder [1, pp. 26–30 and 141–180; 2, pp. 464–473], then the set of all demodulator output blocks that would be decoded as the same codeword forms a Voronoi region.

Considerable efforts have been devoted to the problem of computing Voronoi regions for general point sets [3]. One common solution is to project the point set onto the surface of a hypersphere in one higher dimension, and then apply a convex hull algorithm on this transformed point set [4, pp. 95–100 and 257–261; 5, pp. 246–248]. A different approach is to employ linear programming [6, 7]. In [8, p. 41], some additional references on methods for the construction of Voronoi regions are mentioned.

The special case when the point set constitutes a lattice was thoroughly covered by Conway and Sloane [8, esp. Chapters 4 and 21]. Fast methods have been developed for different classes of lattices. Consequently, the Voronoi regions are known for many important lattices.

This paper considers another special case, binary linear block codes. Knowledge of their Voronoi regions is valuable for the theoretical analysis of codes and their geometrical properties. “The shape of the Voronoi region determines almost all properties of [the code] that are important for communications,” as Forney stated [9].

Slepian pioneered in the study of group codes[1] in Euclidean space and gave some general

[1] Slepian’s definition of “group codes,” which allows for non-integer point sets, is different from the algebraic definition that is usual in error-correcting coding [1, p. 10]. However, all binary linear block codes belong to both

theorems on their Voronoi regions [10]. The relation between block codes and lattices was analyzed by Conway and Sloane [11], who also gave a covering summary of soft decision decoding algorithms for various types of both binary codes and lattices. Gao, Rudolph, and Hartmann defined a class of spherical codes, derived from maximal-length (simplex) codes, whose Voronoi regions have a special structure [12]. A decoding algorithm utilizing this structure was proposed for the class. Forney introduced the class of geometrically uniform codes [9], which includes both Slepian group codes (and thereby binary linear block codes) and lattices, and developed some general properties of their Voronoi regions. Recently, Butovitsch presented a soft decision decoding algorithm based on Voronoi regions and described the regions for some codes [13, Parts D and E]. He also gave a useful approximation of Voronoi regions, and applied it in a decoder.

The subject of the present paper is the problem of determining the Voronoi regions for an arbitrary binary linear block code. The outline is as follows. The notation and terminology to be used in the paper is introduced in section II. The two most important concepts are Voronoi neighbors and Gabriel neighbors. Section III is a treatment of binary linear block codes from a geometrical point of view. This leads to the theorem that all hyperplanes that bound the Voronoi region of a codeword correspond to Gabriel neighbors of the codeword. Section IV presents a fast method to recognize Gabriel neighbors. In Section V, these theoretical results are employed to find the Voronoi regions for some codes of common interest. Section VI is a summary.

II. Preliminaries

Consider a set of M points in $\mathbb{R}^n$, $\mathcal{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_M\}$, and the distance measure $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$, where the norm $\|\mathbf{z}\|^2$ equals the squared Euclidean distance $\mathbf{z} \cdot \mathbf{z}$. If all vectors in $\mathbb{R}^n$ are grouped together according to which of the points in $\mathcal{C}$ they are closest to, space will be

types of group codes.

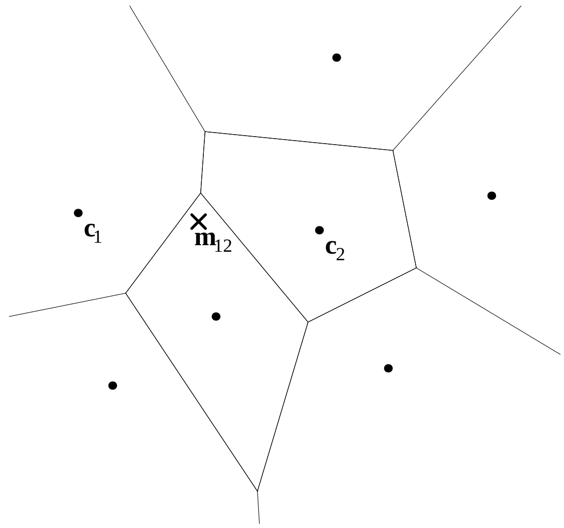

Fig. 1. Seven points in $\mathbb{R}^n$ and their Voronoi regions. The points $\mathbf{c}_1$ and $\mathbf{c}_2$ are Voronoi neighbors, but not Gabriel neighbors.

partitioned into *Voronoi regions*. The Voronoi region $\mathcal{V}_i$ of a point $\mathbf{c}_i \in \mathcal{C}$ is defined as the set of vectors in $\mathbb{R}^n$ that are closest to $\mathbf{c}_i$, that is,

$$\mathcal{V}_i = \left\{ \mathbf{x} \in \mathbb{R}^n : \quad d(\mathbf{x}, \mathbf{c}_i) \le d(\mathbf{x}, \mathbf{c}), \ \forall \mathbf{c} \in \mathcal{C} \right\}. \tag{1}$$

In case of a tie between two or more points in $\mathcal{C}$, the vector belongs to more than one Voronoi region. However, no vector can belong to the *interior* of more than one Voronoi region. Fig. 1 shows a set of seven points and the two-dimensional Voronoi region of each point.

Each inequality in (1) defines a half-space, a semi-infinite region bounded by a hyperplane. The Voronoi region, as defined by the M inequalities in (1), is the intersection of such half-spaces, that is, a convex polytope in n dimensions. The polytope has usually considerably fewer facets than M, which is to say that several of the inequalities are redundant. The same region can be described by a subset of them,

$$\mathcal{V}_i = \left\{ \mathbf{x} \in \mathbb{R}^n : \quad d(\mathbf{x}, \mathbf{c}_i) \le d(\mathbf{x}, \mathbf{c}), \ \forall \mathbf{c} \in \mathcal{N}_i \right\}. \tag{2}$$

The minimal set $\mathcal{N}_i$ for which the right-hand side equals the Voronoi region (1) is called the set of (Voronoi) *neighbors* of $\mathbf{c}_i$. The point itself, $\mathbf{c}_i$, obviously does not belong to $\mathcal{N}_i$. In Fig. 1, $\mathbf{c}_2$ is an example of a neighbor of $\mathbf{c}_1$.

An equivalent definition is that two points in $\mathcal{C}$ are neighbors if their Voronoi regions have an ($n-1$)-dimensional facet in common, that is if

$$\exists \mathbf{x} \in \mathbb{R}^n : \quad d(\mathbf{x},\mathbf{c}_i) = d(\mathbf{x},\mathbf{c}_j) < d(\mathbf{x},\mathbf{c}), \ \forall \mathbf{c} \in \mathcal{C} \setminus \{\mathbf{c}_i,\mathbf{c}_j\}. \tag{3}$$

Note the use of a strict inequality. That two Voronoi regions share a lower-dimensional face does not imply their being neighbors in the sense defined here.[2] This convention is significant for the classification of degenerate point sets, see Section IV.

A related concept is *Gabriel neighbors* [14; 4, pp. 116–117]. Two points $\mathbf{c}_i$ and $\mathbf{c}_j$ in $\mathcal{C}$ are Gabriel neighbors if

$$d(\mathbf{m}_{ij},\mathbf{c}_i) = d(\mathbf{m}_{ij},\mathbf{c}_j) < d(\mathbf{m}_{ij},\mathbf{c}); \ \forall \mathbf{c} \in \mathcal{C} \setminus \{\mathbf{c}_i,\mathbf{c}_j\}, \tag{4}$$

where $\mathbf{m}_{ij}$ is the vector halfway between $\mathbf{c}_i$ and $\mathbf{c}_j$,

$$\mathbf{m}_{ij} = \tfrac{1}{2}(\mathbf{c}_i + \mathbf{c}_j). \tag{5}$$

An interpretation of this definition is that $\mathbf{c}_i$ and $\mathbf{c}_j$ are Gabriel neighbors if the straight line joining the two points goes directly from $\mathcal{V}_i$ to $\mathcal{V}_j$, without touching a third Voronoi region. For example, $\mathbf{c}_1$ and $\mathbf{c}_2$ in Fig. 1 are not Gabriel neighbors. In consequence with the definition (3) of Voronoi neighbors, Gabriel neighbors (4) are also defined using a strict inequality.

[2] There is no uniform agreement in the computational geometry literature regarding the borderline between neighbors and non-neighbors. The strict inequality in (3), which is used throughout this paper, is necessary for a minimal description of Voronoi regions.

If two points are Gabriel neighbors, then they are also Voronoi neighbors. This is obvious from a comparison between (3) and (4), and it holds for any point constellation. The converse, however, is not necessarily true. For a general constellation, the set of Gabriel neighbors is a subset of the set of Voronoi neighbors.

We now confine the discussion to point sets in which each point $\mathbf{c}_i$ is a codeword in an (n,k) *binary linear block code*. The bit values 0 and 1 are interpreted as coordinates, so that $\mathcal{C} \subseteq \{0,1\}^n$ and $M = 2^k$.[3] Two types of addition will be used, component-wise, on binary vectors: $\mathbf{x} \oplus \mathbf{y}$ denotes modulo-2 integer addition, whereas $\mathbf{x} + \mathbf{y}$ retains its normal real-valued meaning.

The purpose of the next section is to prove that for this special class of point sets, the two neighbor concepts defined above are equal.

III. VORONOI NEIGHBORS ARE GABRIEL NEIGHBORS

If two codewords in a binary linear block code are Voronoi neighbors, then they are also Gabriel neighbors. That is, the relation illustrated by $\mathbf{c}_1$ and $\mathbf{c}_2$ in Fig. 1 cannot exist for a binary linear block code. We prove this indirectly, by considering two codewords that are *not* Gabriel neighbors and showing that they cannot be Voronoi neighbors either. In order to make the proof more accessible, we introduce three lemmas before tackling the main theorem. The first lemma states that the midpoint of the line joining any two codewords does not lie in the interior of any Voronoi region.

Lemma 1: If $\mathbf{c}_i$ and $\mathbf{c}_j$ are two codewords in a binary code, then the vector $\mathbf{m}_{ij}$ halfway between them lies on the boundary of both $\mathcal{V}_i$ and $\mathcal{V}_j$.

[3] To facilitate the simultaneous treatment of codewords as binary sequences and as points in Euclidean space, the usual mapping $\{0 \to 1, 1 \to -1\}$ is *not* applied in this paper.

Proof: Let h_{ij} be the Hamming distance between $\mathbf{c}_i$ and $\mathbf{c}_j$. Then h_{ij} components differ between $\mathbf{c}_i$ and $\mathbf{c}_j$, and the corresponding components of $\mathbf{m}_{ij}$ will be equal to 1/2. Since the components of an arbitrary codeword $\mathbf{c}$ in a binary code are either 0 or 1, the distance from this codeword to $\mathbf{m}_{ij}$ satisfies the bound

$$d\left(\mathbf{m}_{ij}, \mathbf{c}\right) \geq \tfrac{1}{4} h_{ij}. \tag{6}$$

Equality holds for $\mathbf{c}_i$ and $\mathbf{c}_j$, so $\mathbf{m}_{ij}$ cannot be closer to any other codeword than to these two. Thus, both $\mathcal{V}_i$ and $\mathcal{V}_j$ contain $\mathbf{m}_{ij}$, according to the definition (1). ❑

Note that, because of the strict inequality in (4), Lemma 1 does not imply that all pairs of codewords are Gabriel neighbors.

Lemma 2: For a binary code, if $\mathbf{m}_{ij}$ belongs to a third Voronoi region $\mathcal{V}_l$, then

$$(\mathbf{c}_l - \mathbf{c}_i) \cdot \left(\mathbf{c}_l - \mathbf{c}_j\right) = 0. \tag{7}$$

Proof: Lemma 1 and (1) together imply that

$$d\left(\mathbf{c}_l, \mathbf{m}_{ij}\right) = d\left(\mathbf{c}_i, \mathbf{m}_{ij}\right), \tag{8}$$

which after expansion of the squares, cancellation of common terms, and factorization yields (7). ❑

Lemma 3: If three Voronoi regions in a binary linear block code meet at $\mathbf{m}_{ij}$, then a fourth Voronoi region also reaches this vector. The fourth codeword is, if the first three are $\mathbf{c}_i$, $\mathbf{c}_j$, and $\mathbf{c}_l$, equal to

$$\mathbf{c}_m = \mathbf{c}_i + \mathbf{c}_j - \mathbf{c}_l. \tag{9}$$

Proof: A reordering of the terms in (9) yields $\mathbf{c}_m - \mathbf{m}_{ij} = \mathbf{m}_{ij} - \mathbf{c}_l$, which implies that $d\left(\mathbf{m}_{ij}, \mathbf{c}_m\right) = d\left(\mathbf{m}_{ij}, \mathbf{c}_l\right)$. This distance is according to the assumptions also equal to $d\left(\mathbf{m}_{ij}, \mathbf{c}_i\right)$ and

$d(\mathbf{m}_{ij}, \mathbf{c}_j)$, so the four points have the same distance to $\mathbf{m}_{ij}$.

What remains to prove is that the point $\mathbf{c}_m$ belongs to the code. We accomplish this by showing that

$$\mathbf{c}_m = \mathbf{c}_i \oplus \mathbf{c}_j \oplus \mathbf{c}_l \tag{10}$$

when $\mathbf{c}_i$, $\mathbf{c}_j$, and $\mathbf{c}_l$ satisfy Lemma 2. Consider the inner product in (7). It can be expanded as

$$\sum_{b=0}^{n-1} \left(c_l^b - c_i^b\right)\left(c_l^b - c_j^b\right), \tag{11}$$

where c^b denotes the bth component of the vector $\mathbf{c}$. The terms $\left(c_l^b - c_i^b\right)\left(c_l^b - c_j^b\right)$ are all non-negative, because the components are either 0 or 1. Since the sum (11) is equal to zero (Lemma 2), all of the terms must be zero,

$$\left(c_l^b - c_i^b\right)\left(c_l^b - c_j^b\right) = 0; \;\; b = 0, \cdots, n-1. \tag{12}$$

An equivalent way to write this relation is

$$0 \le c_i^b + c_j^b - c_l^b \le 1; \;\; b = 0, \cdots, n-1, \tag{13}$$

which is easily verified by insertion of all possible combinations of bit values $\left(c_i^b, c_j^b, c_l^b\right)$. Thus,

$$\begin{aligned} \mathbf{c}_i + \mathbf{c}_j - \mathbf{c}_l &= \left(\mathbf{c}_i + \mathbf{c}_j - \mathbf{c}_l\right) \bmod 2 \\ &= \mathbf{c}_i \oplus \mathbf{c}_j \oplus \mathbf{c}_l, \end{aligned} \tag{14}$$

which is a codeword. ❑

The algorithm to be introduced in the next section is based upon a generalization of this lemma, namely that the number of Voronoi regions meeting at a midpoint $\mathbf{m}_{ij}$ is always a power of two. This will be shown using Theorem 2.

These three lemmas constitute the background for the main theorem of this paper, namely that the Voronoi and Gabriel neighbor concepts are equivalent.

Theorem 1: All Voronoi neighbors in a binary linear block code are Gabriel neighbors.

Proof: Suppose that $\mathbf{c}_i$ and $\mathbf{c}_j$ are *not* Gabriel neighbors. Then Lemmas 2 and 3 hold, so there exist some codewords $\mathbf{c}_l$ and $\mathbf{c}_m$ that satisfy (7) and (9). Consider an arbitrary vector $\mathbf{x}$ in $\mathbb{R}^n$. Its distance to $\mathbf{c}_m$ is

$$\begin{aligned}\|\mathbf{x}-\mathbf{c}_m\|^2 &= \left\|(\mathbf{x}-\mathbf{c}_l)+(\mathbf{c}_l-\mathbf{c}_i)+\left(\mathbf{c}_l-\mathbf{c}_j\right)\right\|^2 \\ &= \|(\mathbf{x}-\mathbf{c}_l)+(\mathbf{c}_l-\mathbf{c}_i)\|^2+\|\mathbf{c}_l-\mathbf{c}_j\|^2 \\ &\quad +2\,(\mathbf{x}-\mathbf{c}_l)\cdot\left(\mathbf{c}_l-\mathbf{c}_j\right)+2\,(\mathbf{c}_l-\mathbf{c}_i)\cdot\left(\mathbf{c}_l-\mathbf{c}_j\right) \\ &= \|\mathbf{x}-\mathbf{c}_i\|^2+\left\|(\mathbf{x}-\mathbf{c}_l)+\left(\mathbf{c}_l-\mathbf{c}_j\right)\right\|^2-\|\mathbf{x}-\mathbf{c}_l\|^2,\end{aligned} \tag{15}$$

where the first equality is due to (9) and the third to (7), or equivalently

$$d(\mathbf{x},\mathbf{c}_m)+d(\mathbf{x},\mathbf{c}_l)=d(\mathbf{x},\mathbf{c}_i)+d\left(\mathbf{x},\mathbf{c}_j\right). \tag{16}$$

This implies that there does not exist any vector $\mathbf{x}\in\mathbb{R}^n$ for which

$$d(\mathbf{x},\mathbf{c}_i)=d\left(\mathbf{x},\mathbf{c}_j\right)<d(\mathbf{x},\mathbf{c});\;\;\forall\mathbf{c}\in\{\mathbf{c}_m,\mathbf{c}_l\} \tag{17}$$

and consequently (3) is not satisfied. The codewords $\mathbf{c}_i$ and $\mathbf{c}_j$ are not Voronoi neighbors. ❏

We emphasize that the two restrictions for this theorem, and thus for the Voronoi region determination in the next section, are that the block code is *linear* and *binary*. Obviously, non-linear codes do not in general satisfy Lemma 3. That non-binary codes do not always follow the theorem is easily demonstrated through an example. Consider the simple ternary linear block code consisting of the three codewords $(0,0,0,0)$, $(1,1,1,2)$, and $(2,2,2,1)$. The third codeword is a Voronoi neighbor of the zero codeword, but not a Gabriel neighbor.

IV. DETERMINATION OF VORONOI REGIONS

Several methods exist to determine the Voronoi regions for an arbitrary finite point set. However, two properties make them intractable for the special case when the point set is formed by a binary block code. First, they are slow. In particular, the time required to examine a point set increases rapidly with the dimension (block length), cf. [15]. Second, they do not handle the case of *degenerate* point sets properly, that is, when more than $n+1$ Voronoi regions meet at the same vector in $\mathbb{R}^n$. This situation has zero probability for a random point set (assuming continuous probability density functions), but it occurs frequently for block codes. For example, the codewords in any binary code lie on vertices of a hypercube. Its center, $(1/2,\cdots,1/2)$, belongs to *all* Voronoi regions, but this does not mean that all codewords are neighbors of each other, see the definition (3). The problem with degenerate point sets is that an arbitrarily small numerical error will make the difference between classifying a pair of points as neighbors or not. This calls for integer computations.

Voronoi regions for any point set are convex polytopes. For a binary code, these polytopes have a special structure. Since all codewords in a binary code lie on the surface of the same hypersphere, Voronoi regions are conical, that is, they have only one vertex, which is at the center of the hypersphere, and extend infinitely in some direction [10]. In other words, if $\mathbf{x}$ belongs to a certain Voronoi region, so does $\alpha\mathbf{x}+(1-\alpha)(1/2,\cdots,1/2)$ for any $\alpha \geq 0$.

A description of the Voronoi regions for a given point set requires the knowledge of their $(n-1)$-dimensional sides (facets), that is, the Voronoi neighbors of all points must be found. To decide whether two given points in an unstructured point set are Voronoi neighbors is, as mentioned, a time-consuming task. A test for Gabriel neighbors is much faster. In the previous section, it was shown that the Gabriel test can replace the Voronoi test, if the point set is a binary linear block code.

The standard method to test whether two given points $\mathbf{c}_i$ and $\mathbf{c}_j$ in $\mathcal{C}$ are Gabriel neighbors is to compute $\mathbf{m}_{ij}$ and evaluate the comparisons in (4). Faster methods to find the Gabriel neighbors in an arbitrary point set have been developed [14], but if the point set is a binary linear block code, this fact can be exploited to provide a further complexity reduction. First, we concretize what being Gabriel neighbors means in a binary code.

Theorem 2: Whether two codewords in a binary code are Gabriel neighbors depends only on the positions where both codewords have the same value. If and only if the code does not contain any other codeword that has the same values in these positions, the two codewords are Gabriel neighbors.

Proof: This theorem is an extension of Lemma 1. It was shown that (6) holds with equality for $\mathbf{c} = \mathbf{c}_i$ and $\mathbf{c} = \mathbf{c}_j$. Suppose that there exists a third codeword that agrees with $\mathbf{m}_{ij}$ in all positions where $\mathbf{m}_{ij}$ is not 1/2, that is, in positions where $\mathbf{c}_i$ and $\mathbf{c}_j$ are equal. Then (6) is an equality for this codeword, too, which violates (4). If no such codeword exists, then (4) is satisfied and $\mathbf{c}_i$ and $\mathbf{c}_j$ are Gabriel neighbors. ❑

Because of the symmetrical properties inherent in a linear block code, geometrical problems can be transformed into a form where the zero codeword is involved. Specifically, two codewords $\mathbf{c}_i$ and $\mathbf{c}_j$ are neighbors if and only if $\mathbf{c}_i \oplus \mathbf{c}_j$ and $\mathbf{0} = (0, \cdots, 0)$ are neighbors. Therefore it is sufficient to specialize Theorem 2 into situations where one of the two codewords is $\mathbf{0}$.

Corollary 1: A codeword $\mathbf{c}$ in a binary code is a neighbor of $\mathbf{0}$ if and only if there is no more codeword having zeros where $\mathbf{c}$ has zeros.

The obvious way to perform this test it to generate all codewords in the code, comparing each one of them to $\mathbf{c}$. The computational complexity of this approach is $O\left(n \cdot 2^k\right)$. In the Appendix, a more efficient algorithm is presented. Instead of generating all codewords, the number of codewords having zeros in certain positions is found by linear row operations on the generator matrix $\mathbf{G}$. The matrix is reorganized into another valid generator matrix for the same code, consisting of two parts: all of the last r rows contain the specified zero pattern, and the first $k-r$ rows are such that no linear combination of them can yield this pattern. Then a total of 2^r codewords, including $\mathbf{c}$ and $\mathbf{0}$, will contain the zero pattern. If $r=1$, then $\mathbf{c}$ and $\mathbf{0}$ are neighbors according to Corollary 1; if $r>1$, this is not the case.[4]

The complexity for the test of *one* codeword is $O(n^2k)$, using the algorithm in the Appendix, which is a considerable improvement over $O(n \cdot 2^k)$, except when k is very small. To find *all* facets of the Voronoi region of $\mathbf{0}$, all 2^k-1 non-zero codewords have to be classified as neighbors or not neighbors, so the complexity for the complete determination of one Voronoi region is $O(n^2k \cdot 2^k)$. The complexity for several Voronoi regions is the same as for one, since all Voronoi regions are congruent [10].

A few observations can reduce the complexity. First, Corollary 1 is connected to the distance properties of the code [13, Part D]. It is easily shown that all codewords with a weight equal to $w_{\max} = \max(2d_1(\mathcal{C}), d_2(\mathcal{C})) - 1$ or less, where $d_r(\mathcal{C})$ is the rth *generalized Hamming weight* of the code $\mathcal{C}$ [16], have to be neighbors of $\mathbf{0}$. The number of codewords that satisfies this condition can be quite large for some codes and negligible for others, depending on the weights. The breakpoint is at about $w_{\max} = n/2$. A promising area for future research appears to be to investigate the relation between Hamming weight hierarchy and Voronoi regions further, and to improve the algorithm by incorporating weight information into it. This approach is encouraged by the progress that has been made in using generalized Hamming weights to characterize the complexity of trellises for linear block codes [17].

Second, any known automorphism [18, pp. 229–238] of the code can be exploited to reduce

[4] The value of r is, when the algorithm in the Appendix exits, equal to $k-i+1$.

the number of codewords that need to be examined by the algorithm. For example, if the considered code is cyclic, it is sufficient to test one representative of each cycle. (A cycle is a set of codewords that are cyclic shifts of one another.) This reduces the number of needed Gabriel tests by a factor close to n, since most cycles contain n different codewords, so the total complexity is $O\left(nk \cdot 2^k\right)$. An efficient algorithm for the generation of cycle representatives is given in [19].

The codewords in the code are not stored simultaneously. Instead, they are generated at need, which makes the memory requirement very modest, only $O(nk)$. The major part thereof is for storage of the generator matrix.

V. Voronoi Regions for Some Common Codes

The theory of the preceding sections makes it possible to obtain explicit results on the Voronoi regions for many interesting binary linear codes. This section considers all primitive binary BCH codes of block length n up to 31, including the Hamming codes. Also, a few longer BCH codes and the two binary Golay codes are discussed.

To begin with, consider the $(n, n-1)$ single-parity code, which consists of all binary vectors of length n with an even weight. If any two bits of a codeword are inverted, the result is another codeword. Thus, the interpretation of Theorem 2 for this simple code is that two codewords are neighbors if and only if their distance is 2, regardless of the block length n.

TABLE I

THE FACETS OF THE VORONOI REGION OF THE ZERO CODEWORD IN THE (15,7) BCH CODE

Codeword	*Number of equivalent shifts*	*Neighbor of* **0**?
000000111010001	15	Yes
000001001110011	15	Yes
000010100110111	15	Yes
000011010010101	15	Yes
000101110111111	15	No
000110011111011	15	Yes
000111101011001	15	Yes
001001001001001	3	Yes
001011010101111	15	Yes
011011011011011	3	No
111111111111111	1	No

In the (7,4) Hamming code, the 16 codewords were compared using Theorem 2. This showed that each codeword has 14 neighbors. A given codeword is a neighbor of all the others except the antipodal one. Thus, the Voronoi regions for this code are 14-sided 7-dimensional polytopes, described by (2), where $\mathcal{N}_i = \mathcal{C} \setminus \{\mathbf{c}_i, \mathbf{c}_i \oplus (1,\cdots,1)\}$, for all i.

Table I shows the results for the (15,7) BCH code, which has a minimum distance of 5. All codewords except the zero codeword $\mathbf{0}$ are listed, exploiting the cyclic property to compress the table. The relation to $\mathbf{0}$ (neighbor or not neighbor) is given for the codewords. Of the 127 non-zero codewords, 108 were found to be neighbors of $\mathbf{0}$. Table II, in which the codewords are collected according to their Hamming weights, shows that the set of neighbors consists of all codewords with a weight less than or equal to 9, and no others.

TABLE II

THE NUMBER OF NEIGHBORS OF **0** WITH A GIVEN HAMMING WEIGHT, FOR THE (15,7) BCH CODE

Hamming weight	*Number of neighbors*	*Number of non-neighbors*
5	18	0
6	30	0
7	15	0
8	15	0
9	30	0
10	0	18
15	0	1
All	108	19

The middle column in this type of table, which gives the weight distribution of the neighbors, is called the *local distance profile*. It can be used to compute an upper bound on the error probability for a soft decision decoder that is tighter than the usual union bound [9]. Denoting the number of neighbors with weight w with L_w, the error probability for equiprobable signaling over a discrete-time channel with additive white Gaussian noise is bounded by

$$P_e \leq \frac{1}{2} \sum_{w=1}^{n} L_w \operatorname{erfc}\left(\frac{\sqrt{w\, SNR}}{2\sqrt{2}} \right), \tag{18}$$

where SNR is the ratio between signal and noise power. The usual union bound would use the full distance profile, that is the sum of the second and third columns, instead of L_w. For the (15,7) BCH code, the difference between the two bounds is minor, since most codewords are included in the local distance profile, but subsequent tables will demonstrate codes for which this is not the case. Other bounds related to the union bound, such as Berlekamp's *tangential union bound* [20], may be improved in a similar fashion.

The relation between neighbors and weights for the (15,11) and (31,26) Hamming codes is shown in Tables III and IV, respectively. Note that it is not possible to conclude whether two codewords in the (31,26) Hamming code are neighbors or not by considering the distance between them alone. Codewords with a Hamming distance of 6 can be either type, according to Table IV.

TABLE III

THE (15,11) HAMMING CODE

Hamming weight	*Number of neighbors*	*Number of non-neighbors*
3	35	0
4	105	0
5	168	0
6	0	280
7	0	435
8	0	435
9	0	280
10	0	168
11	0	105
12	0	35
15	0	1
All	308	1739

TABLE IV

THE (31,26) HAMMING CODE

Hamming weight	*Number of neighbors*	*Number of non-neighbors*
3	155	0
4	1085	0
5	5208	0
6	13888	8680
7	0	82615
8	0	247845
9	0	628680
10	0	1383096
11	0	2648919
12	0	4414865
13	0	6440560
14	0	8280720
15	0	9398115
16	0	9398115
17	0	8280720
18	0	6440560
19	0	4414865
20	0	2648919
21	0	1383096
22	0	628680
23	0	247845
24	0	82615
25	0	22568
26	0	5208
27	0	1085
28	0	155
31	0	1
All	20336	67088527

TABLE V

THE (23,12) GOLAY CODE

Hamming weight	*Number of neighbors*	*Number of non-neighbors*
7	253	0
8	506	0
11	1288	0
12	1288	0
15	0	506
16	0	253
23	0	1
All	3335	760

TABLE VI

THE (24,12) GOLAY CODE

Hamming weight	*Number of neighbors*	*Number of non-neighbors*
8	759	0
12	2576	0
16	0	759
24	0	1
All	3335	760

The Voronoi regions for the (23,12) Golay code all have 3335 facets, see Table V. Adding a parity bit to create the (24,12) Golay code (Table VI) does not topologically alter the Voronoi regions, even though the inter-codeword distances are changed.

In both the (31,16) and the (31,21) BCH codes, summarized in Tables VII and VIII, there exist neighbors that have a greater weight than some non-neighbors. The (15,5), (31,6), and (31,11) BCH codes are not tabulated here, because their Voronoi regions have always $M-2$ facets. All non-trivial pairs of codewords are neighbors, just as in the (7,4) Hamming code.

TABLE VII

THE (31,16) BCH CODE

Hamming weight	*Number of neighbors*	*Number of non-neighbors*
7	155	0
8	465	0
11	5208	0
12	8680	0
15	13888	4371
16	13888	4371
19	0	8680
20	0	5208
23	0	465
24	0	155
31	0	1
All	42284	23251

TABLE VIII

THE (31,21) BCH CODE

Hamming weight	*Number of neighbors*	*Number of non-neighbors*
5	186	0
6	806	0
7	2635	0
8	7905	0
9	18910	0
10	35092	6510
11	41664	43896
12	0	142600
13	0	195300
14	0	251100
15	0	301971
16	0	301971
17	0	251100
18	0	195300
19	0	142600
20	0	85560
21	0	41602
22	0	18910
23	0	7905
24	0	2635
25	0	806
26	0	186
31	0	1
All	107198	1989953

The tables show, not surprisingly, that for a fixed n, the percentage of the codewords that are neighbors decreases with increasing k. This behavior is summarized, for the BCH codes, in Figs. 2 and 3. As endpoints in the diagrams, the $(n,1)$ repetition code and the (n,n) universe code, which fills the binary code space, are included.

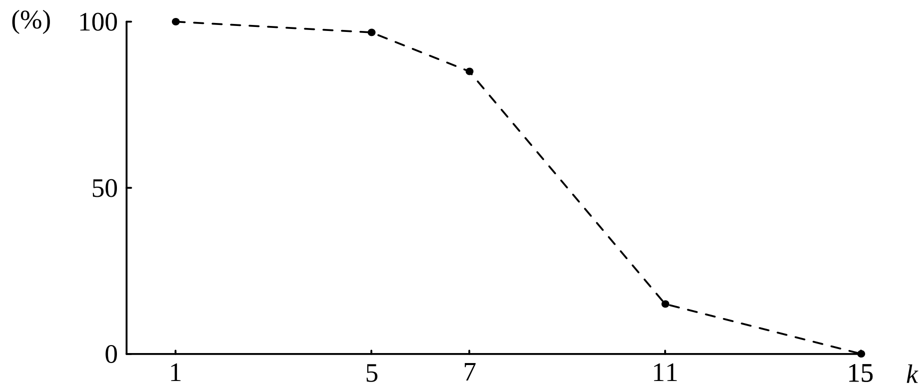

Fig. 2. The probability that two arbitrary codewords in a primitive BCH code with $n = 15$ are neighbors.

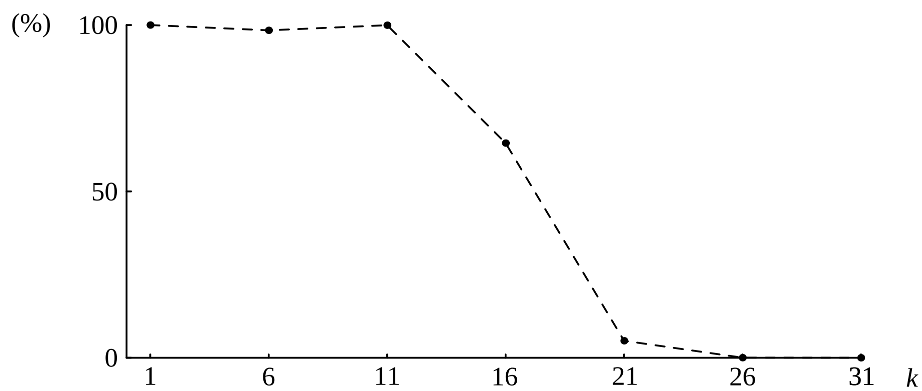

Fig. 3. The probability that two arbitrary codewords in a code with $n = 31$ are neighbors.

The transition between few and many neighbors is surprisingly fast. For k values up to approximately one third of n, practically all pairs of codewords are neighbors of each other, and for k values above two thirds, almost none of them are. Fig. 4, which shows the corresponding curve for some primitive $(63, k)$ BCH codes, suggests that the same tendency holds for larger block lengths n, too. This curve also demonstrates the algorithm's capability to handle large codes. Voronoi regions were determined for code sizes up to $k = 30$. It took a couple of hours to examine the 1 billion codewords of the (63,30) BCH code, using a modern workstation.

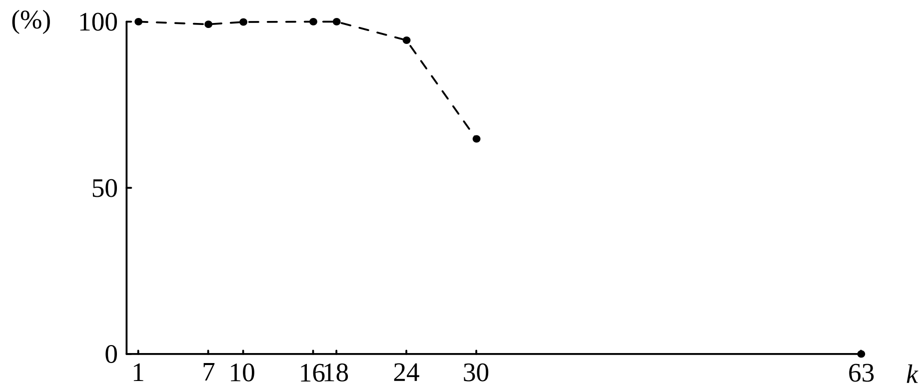

Fig. 4. The probability that two arbitrary codewords in a code with $n = 63$ are neighbors.

VI. SUMMARY

The paper contains results on two levels. It tabulates the Voronoi regions for several binary linear block codes, mostly BCH codes, and it presents an efficient method to determine such Voronoi regions. The study of different block codes displays a strong connection between the ratio k/n and the percentage of codeword pairs that are neighbors of each other. Another conclusion is that the codewords that bound the Voronoi region of a given codeword are not necessarily those that are closest to this codeword.

The employed algorithm is based primarily on two theoretical results for binary linear block codes. First, two codewords are Voronoi neighbors if and only if they are Gabriel neighbors. Second, a simple reorganization of the generator matrix shows whether two codewords are Gabriel neighbors.

APPENDIX:
AN ALGORITHM TO DECIDE WHETHER $\mathbf{c}$ BOUNDS THE VORONOI REGION OF $\mathbf{0}$

Input: $\mathbf{g}_1, \cdots, \mathbf{g}_k$, the rows of a $k \times n$ generator matrix $\mathbf{G}$. Destroyed in the algorithm.

$\mathbf{c}$, a codeword to test.

Exits: NEIGHBOR if $\mathbf{c}$ and $\mathbf{0}$ are neighbors, otherwise NOT NEIGHBOR.

Algorithm:

```
Set i = 1.
For p = the position of all zeros in c:
    Set firstone = TRUE.
    For j = i to k:
        If g_j has a one in position p:
            If firstone:
                Set pivot = j.
                firstone = FALSE.
            else:
                Set g_j = g_j ⊕ g_pivot.
            end.
        end.
    end.
    If not firstone:
        Swap g_i and g_pivot.
        Set i = i + 1.
        If i = k: Exit(NEIGHBOR).
    end.
end.
Exit(NOT NEIGHBOR).
```

ACKNOWLEDGMENTS

The author would like to thank Per Hedelin, Stefan Dodunekov, and Mikael Skoglund, whose helpful comments led to significant improvements of this paper.

REFERENCES

[1] G. C. Clark, Jr. and J. B. Cain, *Error-Correction Coding for Digital Communications*. New York, NY: Plenum Press, 1981.

[2] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley Publishing Company, 1983.

[3] F. Aurenhammer, “Voronoi diagrams—A survey of a fundamental geometric data structure,” *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, Sept. 1991.

[4] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Chichester, England, U.K.: John Wiley & Sons, 1992.

[5] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. New York, NY: Springer-Verlag, 1985.

[6] R. L. Joshi and P. G. Poonacha, “A new MMSE encoding algorithm for vector quantization,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 645–648, Toronto, Ontario, Canada, May 1991.

[7] E. Agrell, “A method for examining vector quantizer structures,” in *Proc. IEEE International Symposium on Information Theory*, p. 394, San Antonio, TX, Jan. 1993.

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

[9] G. D. Forney, Jr., “Geometrically uniform codes,” *IEEE Trans. Inform. Theory*, vol. 37, no. 5, pp. 1241–1260, Sept. 1991.

[10] D. Slepian, “Group codes for the Gaussian channel,” *Bell Syst. Tech. J.*, vol. 47, no. 4, pp. 575–602, Apr. 1968.

[11] J. H. Conway and N. J. A. Sloane, “Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice,” *IEEE Trans. Inform. Theory*, vol. IT-32, no. 1, pp. 41–50, Jan. 1986.

[12] J. Gao, L. D. Rudolph, and C. R. P. Hartmann, “Iteratively maximum likelihood decodable spherical codes and a method for their construction,” *IEEE Trans. Inform. Theory*, vol. 34, no. 3, pp. 480–485, May 1988.

[13] P. Butovitsch, “Classification of signal sets—The classification capability of multi-layer perceptrons and soft-decision decoding of error-correcting codes,” Ph.D. Thesis, Department of Signals, Sensors and Systems, Royal Institute of Technology, Stockholm, Sweden, 1994.

[14] J. W. Jaromczyk and G. T. Toussaint, “Relative neighborhood graphs and their relatives,” *Proc. IEEE*, vol. 80, no. 9, pp. 1502–1517, Sept. 1992.

[15] A. Bowyer, “Computing Dirichlet tessellations,” *The Computer J.*, vol. 24, no. 2, pp. 162–166, May 1981.

[16] V. K. Wei, “Generalized Hamming weights for linear codes,” *IEEE Trans. Inform. Theory*, vol. 37, no. 5, pp. 1412–1418, Sept. 1991.

[17] G. D. Forney, Jr., “Dimension/length profiles and trellis complexity of linear block codes,” *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1741–1752, Nov. 1994.

[18] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, the Netherlands: North-Holland Publishing Company, 1977.

[19] P. E. Allard, S. G. S. Shiva, and S. E. Tavares, “A note on the decomposition of cyclic codes into cyclic classes,” *Inform. Contr.*, vol. 22, no. 1, pp. 100–106, Feb. 1973.

[20] E. R. Berlekamp, “The technology of error-correcting codes,” *Proc. IEEE*, vol. 68, no. 5, pp. 564–593, May 1980.