Error-Correction of Multidimensional Bursts

Eitan Yaakobi
Department of Computer Science
Technion-Israel Institute of Technology
Haifa 32000, Israel
Email: yaakobi@cs.technion.ac.il

Tuvi Etzion
Department of Computer Science
Technion-Israel Institute of Technology
Haifa 32000, Israel
Email: etzion@cs.technion.ac.il

***Abstract*— A construction for D-dimensional binary codes of size $n_1 \times n_2 \times \cdots \times n_D$ correcting a single D-dimensional box error is presented. If the size of the box error is $b_1 \times b_2 \times \cdots \times b_D$, b_i odd, $1 \leq i \leq D$, and $B = \Pi_{i=1}^{D} b_i$, then the redundancy of the code is at most $\lceil \log_2(n_1 n_2 \cdots n_D) \rceil + B + (D-2)\lceil \log_2 B \rceil + \lceil \log_2 b_1 \rceil$. For a two-dimensional binary array of size $n \times n$ we present a code correcting an error whose shape is a Lee sphere with radius R. The redundancy of the code is at most $\lceil \log_2 n^2 \rceil + 2R^2 + 2R + \lceil 2\log_2(2R+1) \rceil + 1$. This is also the redundancy of a binary code which corrects an arbitrary two-dimensional cluster-error of size $2R+1$. A generalization for D-dimensional code which corrects either D-dimensional error whose shape is a Lee sphere or an arbitrary cluster-error is also given.**

I. Introduction

In current memory devices for advanced storage systems the information is stored in two or more dimensions. In such systems errors usually take the form of multidimensional bursts. Usually, a cluster of errors either will be affected by the position in which the error event occurred or will be of an arbitrary shape. But, since an arbitrary cluster-error is hard to correct efficiently it is common to assume some type of cluster-error. These types of errors can be of specific shapes like rectangles or Lee spheres. We will consider these types of errors as well as arbitrary errors. The main measure to compute the efficiency of a cluster-error correcting code is its redundancy. If we want to design a code which corrects one cluster-error of size B (of an arbitrary or a specific shape) then the redundancy of the code r satisfies $r \geq 2B$. This bound, known as the Reiger bound [11], is attained for binary two-dimensional codes, which correct a rectangular error, constructed recently by Boyarinov [6]. If the volume of the array is N then the redundancy of a code must also satisfy $r \geq \log_2 N + B - 1$ (usually $r - \lceil \log_2 N \rceil \geq B$). The difference $r - \lceil \log_2 N \rceil$ will be called the *excess redundancy* of the code [1], [4]. Abdel-Ghaffar [1] constructed a binary two-dimensional code which corrects a burst with a rectangle shape for which $r = \lceil \log_2 N \rceil + B$. The code has a few disadvantages: very limited size, complicated construction, and there is no obvious generalization for higher dimensions.

In this paper we establish the following results:

1) A construction for two-dimensional codes which correct a cluster-error whose shape is a rectangle of size $b_1 \times b_2$, with excess redundancy at most $b_1 b_2 + \lceil \log_2 b_1 \rceil$ and the size of the code is relatively not limited. The construction has a simple generalization for higher dimensions.
2) An efficient construction for multidimensional codes which correct a cluster error whose shape is a Lee sphere or an arbitrary shape.

In Section II we briefly survey some of the known constructions which are essential to understand our results. In Section III we present a construction for codes which correct a multidimensional box error. In Section IV we discuss correction of a multidimensional Lee spheres and arbitrary errors.

II. Known Constructions

Most of two-dimensional codes known in the literature are designed to correct a single cluster-error of size $b_1 \times b_2$ [1], [4], [6], [7], [10]. In some recent papers [5], [8], [12] it is assumed that the cluster-error can have an arbitrary shape. Two of these constructions are important in our discussion. Abdel-Ghaffar [1] gave a construction of such $n_1 \times n_2$ code with excess redundancy $b_1 b_2$. His construction is a generalization of the optimum one-dimensional codes which correct a single cyclic burst of length b [2], [3]. The existence of such codes was obtained by the following necessary and sufficient conditions.

Theorem 1: If a polynomial $g(x)$ generates an optimum b-burst-correcting code over GF(q), then it can be factored as $g(x) = e(x)p(x)$, where $e(x)$ and $p(x)$ satisfy the conditions:

1) $e(x)$ is a square-free polynomial of degree $b-1$ which is not divisible by x such that h_e and m_e are relatively primes to $q-1$, where h_e and m_e are the period and the degree of the splitting field of $e(x)$, respectively.
2) $p(x)$ is an irreducible polynomial of degree $m \geq b+1$ and period $\frac{q^m-1}{q-1}$ such that m and $q-1$ are relatively primes and $m \equiv 0 \pmod{m_e}$.

A monic polynomial over GF(q) which satisfies condition 1) of Theorem 1 will be called a *b-polynomial*.

Theorem 2: Let $e(x)$ be a b-polynomial over GF(q). Then, for all sufficiently large m relatively prime to $q-1$ such that $m \equiv 0 \pmod{m_e}$, where m_e is the degree of the splitting field of $e(x)$, there exists an irreducible polynomial $p(x)$ of degree m such that $e(x)p(x)$ generates an optimum b-burst correcting code of length $\frac{q^m-1}{q-1}$.

Breitbach, Bossert, Zybalov, and Sidorenko [7] gave three constructions of two-dimensional codes of size $n_1 \times n_2$ which correct rectangular error of size $b_1 \times b_2$. We use ideas from one of the constructions which will be called Construction BBZS.

1-4244-1429-6/07/$25.00

A codeword $\{c_{ij}\}$ of the construction has size $n_1 \times n_2 = 2^{b_2} \times 2^{b_1}$ with $4b_1b_2$ redundancy bits located in positions $\{(i,j) : 0 \le i \le 2b_1 - 1, n_2 - b_2 \le j \le n_2 - 1\} \cup \{(i,j) : n_1 - b_1 \le i \le n_1 - 1, 0 \le j \le 2b_2 - 1\}$ (see Fig. 1). These bits are set initially to be zeroes. Two temporary *component codes* are being used, a vertical code and an horizontal code. We will describe the construction of the vertical code.

For each row $i = 2b_1, \ldots, n_1 - 1$, b_2 parity check bits are generated. p_{im}, $m = 0, 1, \ldots, b_2 - 1$ is computed as

$$p_{im} = \sum_{j=m,m+b_2,m+2b_2,\ldots} c_{ij}. \tag{1}$$

The parity bits p_{im}, $m = 0, 1, \ldots, b_2 - 1$ generate afterward a symbol $\underline{p}_i = (p_{i0}, p_{i1}, \ldots, p_{ib_2-1})$ from the extension field GF(2^{b_2}). The symbols $\underline{p}_i$, $i = 2b_1, \ldots, n_1 - 1$ are considered as the information symbols of a Reed-Solomon (RS) code of length n_1, dimension $n_1 - 2b_1$, and minimum distance $d = 2b_1 + 1$. By the encoding process of the RS code we obtain $2b_1$ redundancy symbols $\underline{p}_i$, $i = 0, \ldots, 2b_1 - 1$, and the $2b_1b_2$ upper right corner redundancy bits of the array are computed in a way that (1) holds for $m = 0, 1, \ldots, b_2 - 1$, and $i = 0, 1, \ldots, 2b_1 - 1$. The encoding process of the horizontal code is done in the same manner, where all the $4b_1b_2$ redundancy bits of the array are assumed to be zeroes. In the decoding process each row generates b_2 parity bits according to (1) such that a word of length n_1 over GF(2^{b_2}) is received (the redundancy bits of the horizontal code are assumed to be zeroes). Assuming that the error occurred in the array can be confined inside a rectangular of size $b_1 \times b_2$, the generated word, of the vertical code, has at most b_1 erroneous symbols, which can be corrected by the decoding procedure of the RS code. The same process is implemented for the horizontal code, and given the combination of errors by the vertical and the horizontal codes it is possible to correct the rectangular error in the array.

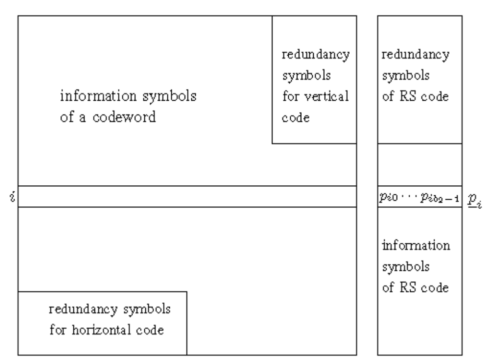

Fig. 1. Construction BBZS

There is no obvious generalization to the construction of Abdel-Ghaffar [1] for multidimensional codes, while immediate generalizations to Construction BBZS cannot support good redundancy or excess redundancy. One way to generalize this construction is to use the optimum cyclic burst-correcting codes of [2], [3] instead of the RS codes. Instead of $4b_1b_2$ redundancy bits we will use $r_1b_2 + b_1r_2$ redundancy bits, where r_1 (r_2) is the redundancy of the vertical (horizontal) code. The excess redundancy of this construction is $2b_1b_2 - 1$ and of its generalization to D-dimensions is $DB - 1$, where B is the volume of the D-dimensional box error. Further improvement of this construction is presented in the next section.

III. Construction for Multidimensional Arrays

The vertical component code of Construction BBZS finds the rows in which the burst occurred and the shape of the burst up to a cyclic permutation of the columns. Therefore, the horizontal component code only has to determine the first column of the burst. More explicitly, the burst $e = (e_0, e_1, \ldots, e_{b_2-1})$, where $e_i \in$ GF(2^{b_1}), for $0 \le i \le b_2 - 1$ found by the vertical code, can start at any column $0 \le i \le n_2 - b_2$. However, if the first column of the burst is i, then the burst occurred is $e = (e_{i_0}, e_{i_0+1}, \ldots, e_{i_0+b_2-1})$ where $i_0 \equiv i (\text{mod } b_2)$, and indices are taken modulo b_2.

Our new construction is based on two lemmas. The first one is proved here only for the binary case, but there is an alternative version for codes over GF(2^ℓ).

Lemma 1: If $e_2(x) = 1 + x + x^2 + \cdots + x^{b-1}$ and b is an odd integer then $e_2(x)$ is a b-polynomial over GF(2).

Proof: Clearly, $e_2(x)$ is not divisible by x. The derivative of $x^b - 1$ over GF(2) is x^{b-1}, and since g.c.d.$(x^b - 1, x^{b-1}) = 1$, it follows that $x^b - 1$ is a square-free polynomial and hence $e_2(x)$ is also square-free. ∎

By Theorem 2, for the b_2-polynomial $e_2(x) = 1 + x + x^2 + \cdots + x^{b_2-1}$, over GF($2^{b_1}$), there exists an irreducible polynomial $p_2(x)$ of degree $m_2 = r_2 - b_2 + 1$ such that $e_2(x)p_2(x)$ generates an optimum b_2-burst correcting code of length $n_2 = \frac{(2^{b_1})^{m_2} - 1}{2^{b_1} - 1}$ and redundancy r_2. Let $\mathcal{C}^* = \{f(x) \in \text{GF}(2^{b_1})[x] : e_2(x)p_2(x) | f(x), \deg f(x) < n_2\}$, and let $\mathcal{C}_2 = \{f(x) \in GF(2^{b_1})[x] : p_2(x) | f(x), \deg f(x) < n_2\}$. The code $\mathcal{C}_2$ is also of length n_2 and has m_2 redundancy symbols over GF(2^{b_1}). We will show now that $\mathcal{C}_2$ can serve as the horizontal component code, i.e., given the burst occurred up to a cyclic permutation $e = (e_0, e_1, \ldots, e_{b_2-1})$, where $e_i \in GF(2^{b_1})$, for $0 \le i \le b_2 - 1$, it will be possible to determine the first column i of the burst $0 \le i \le n_2 - b_2$.

Lemma 2: Let $e = (e_0, e_1, \ldots, e_{b_2-1})$ be a given burst up to a cyclic permutation which occurred in a transmitted codeword and found by the vertical component code. Then, the horizontal component code $\mathcal{C}_2$ can determine the first column of the given burst.

Proof: We have to prove that if the burst e, or a cyclic shift of e, occurred in two different codewords $f_1(x), f_2(x)$ then two different words will be generated. Since $\mathcal{C}_2$ is a linear code it is sufficient to prove that there is no codeword which is equal to the difference of two bursts which are cyclic shifts of the burst e. Assume the first column of the burst e is i, i.e., the burst is $e' = (e_{i_0}, e_{i_0+1}, \ldots, e_{i_0+b_2-1})$ where $i_0 \equiv i (\text{mod } b_2)$ and indices are taken modulo b_2. The polynomial representing the burst is

$$h_i(x) = x^{\ell_i b_2}(e_{i_0}x^{i_0} + e_{i_0+1}x^{i_0+1} + \cdots + e_{b_2-1}x^{b_2-1} + e_0x^{b_2} + e_1x^{b_2+1} + \cdots + e_{i_0-1}x^{b_2+i_0-1}),$$

where $i_0 \equiv i (\text{mod } b_2)$, and $\ell_i = \lfloor \frac{i}{b_2} \rfloor$. We can write $h_i(x)$ as

$$\begin{aligned} h_i(x) &= x^{\ell_i b_2}\big(e_0 + e_1x + \cdots + e_{b_2-1}x^{b_2-1} + e_0(x^{b_2}-1) + \\ &e_1(x^{b_2+1}-x) + \cdots + e_{i_0-1}(x^{b_2+i_0-1} - x^{i_0-1})\big) = \\ &x^{\ell_i b_2}\big(e_0 + e_1x + \cdots + e_{b_2-1}x^{b_2-1} + \\ &(x^{b_2}-1)(e_0 + e_1x + \cdots + e_{i_0-1}x^{i_0-1})\big) = \\ &x^{\ell_i b_2}\left(e_0 + e_1x + \cdots + e_{b_2-1}x^{b_2-1}\right) \\ &+ x^{\ell_i b_2}(x^{b_2}-1)\left(e_0 + e_1x + \cdots + e_{i_0-1}x^{i_0-1}\right). \end{aligned}$$

Assume the contrary, that the difference between two bursts which are cyclic shifts of the burst e is a codeword. Assume that these two bursts start at columns i and j. Hence, the polynomial $h_i(x) - h_j(x)$ is the codeword

$$\begin{aligned} h_i(x) - h_j(x) &= x^{\ell_i b_2}\left(e_0 + e_1x + \cdots + e_{b_2-1}x^{b_2-1}\right) + \\ &x^{\ell_i b_2}(x^{b_2}-1)\left(e_0 + e_1x + \cdots + e_{i_0-1}x^{i_0-1}\right) - \\ &x^{\ell_j b_2}\left(e_0 + e_1x + \cdots + e_{b_2-1}x^{b_2-1}\right) - \\ &x^{\ell_j b_2}(x^{b_2}-1)\left(e_0 + e_1x + \cdots + e_{j_0-1}x^{j_0-1}\right), \end{aligned}$$

i.e., $h_i(x) - h_j(x)$ is a multiple of $p_2(x)$. $h_i(x) - h_j(x)$ can be written as

$$\begin{aligned} h_i(x) - h_j(x) &= (x^{\ell_i b_2} - x^{\ell_j b_2}) \\ &\left(e_0 + e_1x + \cdots + e_{b_2-1}x^{b_2-1}\right) + \\ &(x^{b_2}-1)\big(x^{\ell_i b_2}\left(e_0 + e_1x + \cdots + e_{i_0-1}x^{i_0-1}\right) \\ &-x^{l_j b_2}\left(e_0 + e_1x + \cdots + e_{j_0-1}x^{j_0-1}\right)\big) = \\ &x^{\ell_j b_2}(x^{(\ell_i-\ell_j)b_2}-1)\left(e_0 + e_1x + \cdots + e_{b_2-1}x^{b_2-1}\right) + \\ &(x^{b_2}-1)\big(x^{\ell_i b_2}\left(e_0 + e_1x + \cdots + e_{i_0-1}x^{i_0-1}\right) - \\ &x^{\ell_j b_2}\left(e_0 + e_1x + \cdots + e_{j_0-1}x^{j_0-1}\right)\big). \end{aligned}$$

This last presentation of $h_i(x) - h_j(x)$ implies that it is dividable by $x^{b_2} - 1$ and hence also by $e_2(x) = \frac{x^{b_2}-1}{x-1}$. Therefore, if $h_i(x) - h_j(x)$ is a codeword in $\mathcal{C}_2$ then $p_2(x)|h_i(x) - h_j(x)$ and also $e_2(x)|h_i(x) - h_j(x)$. Since $p_2(x)$ is an irreducible polynomial and its degree is greater than b_2 it follows that $e_2(x)p_2(x)|h_i(x) - h_j(x)$ and therefore $h_i(x) - h_j(x)$ is also a codeword of $\mathcal{C}^*$, a contradiction since $\mathcal{C}^*$ can correct any burst of length b_2. ■

We will call a code with the properties of $\mathcal{C}_2$ as proved in Lemma 2 a *locator code*. Based on the constructions of [3], [7] and Lemma 2 we can construct an $n_1 \times n_2$ two-dimensional $b_1 \times b_2$-burst correcting code with small excess redundancy.

Let $\mathcal{C}_1$ be an optimum b_1-burst correcting code, over GF(2^{b_2}), of length $n_1 = \frac{(2^{b_2})^{r_1-b_1+1}-1}{2^{b_2}-1}$ and redundancy r_1. Let $\mathcal{C}_2$ be a locator code, over GF(2^{b_1}), of length $n_2 = \frac{(2^{b_1})^{m_2}-1}{2^{b_1}-1}$ and redundancy $m_2 = r_2 - b_2 + 1$. Each codeword of size $n_1 \times n_2$ has $r_1b_2 + m_2b_1 + 1 = \lceil \log_2(n_1n_2) \rceil + b_1b_2 + b_1$ redundancy bits in the following positions:

- $\{(i, j) \ : \ 0 \leq i \leq r_1 - 1, n_2 - b_2 \leq j \leq n_2 - 1\}$ bits for the vertical component code.
- $\{(i(2b_1 - 1) + j(2b_1 - 1)b_1, j) \ : \ 0 \leq i \leq b_1 - 1, 0 \leq j \leq m_2 - 1\}$ bits for the horizontal component code.
- $\{(n_1 - 1, n_2 - 1)\}$.

Encoding:

All the redundancy bits in a codeword are set initially to be zeroes. The vertical component code and the first set of redundancy bits are computed as in Construction BBZS. The second set of redundancy bits spans over m_2 consecutive columns. In each column there are b_1 redundancy bits in b_1 positions which cover all the b_1 distinct residues modulo b_1. We compute $n_2 - m_2$ symbols of the horizontal component code as in Construction BBZS, with one exception that the r_1b_2 redundancy bits of the first set are assumed to be the computed values in the previous step (they were assumed to be zeroes in Construction BBZS). The remaining m_2 symbols are the redundancy symbols of the horizontal locator code and are computed from the $n_2 - m_2$ information symbols. The only redundancy bit of the third set is the sum of the computed parity bits from the second set.

Decoding:

The decoding is done similarly to the one of Construction BBZS. We will describe the unique parts of our decoding. The vertical code does not find erroneous redundancy bits from the second and the third sets. These bits are spaced in a way that at most one such bit is in error. If such bit is erroneous then from rows of the error-cluster, discovered by the vertical code, we will know the exact row of this bit. In the decoding procedure we will know if such redundancy bit is erroneous by the value of bit $\{(n_1 - 1, n_2 - 1)\}$. Now, by Lemma 2 the horizontal locator code will discover the first column in which an error occurred, and hence the pattern discovered by the vertical component code enables us to correct the errors. If only redundancy bits of the second set or the third set are erroneous (the vertical code does not find errors) then only one of them can be erroneous. This implies that in the computed codeword of the locator code only one symbol can be erroneous. It is also obvious by definition, that the locator code is the Hamming code and hence it can correct this error which implies that the redundancy bit in error can be corrected.

Remark: The parity bits of the second set can be chosen in other ways as long as they form a set of redundancy symbols for the locator code., e.g., they don't have to be in consecutive columns. Such choices can result in other array sizes.

The redundancy of the construction can be further improved if we use as the horizontal locator code a binary code $\mathcal{C}$ of length $2^m - 1$, where $m = r - b_1b_2 + 1$, which locates a burst of length b_1b_2, b_1b_2 odd and $e_2(x) = 1 + x + \cdots + x^{b_1b_2-1}$. This is done simply by taking the b_1 parity symbols which are computed for each column as b_1 consecutive symbols in $\mathcal{C}$ instead of an element in GF(2^{b_1}). We also have to change the second set of redundancy bits. One possible such set is

- $\{(i(2b_1 - 1) + j(2b_1 - 1)b_1, j) \ : \ 0 \leq i \leq b_1 - 1, \ (j + 1)b_1 + i + 1 \leq m\}$.

Note that the parity bits related to these m redundancy bits are m consecutive bits in the binary locator code. The redundancy of the new code is at most $\lceil \log_2(n_1n_2) \rceil + b_1b_2 + \lceil \log_2 b_1 \rceil$.

The construction is generalized to obtain a D-dimensional code of size $n_1 \times n_2 \times \cdots \times n_D$ which corrects a cluster-error whose shape is a box of size $b_1 \times b_2 \times \cdots \times b_D$. Let $B = b_1 b_2 \cdots b_D$; in the first dimension we use a code of length n_1 over GF($2^{\frac{B}{b_1}}$) which corrects a burst error of size b_1. In each of the other $D-1$ dimensions we use a locator code which locates the position of the error and its cyclic permutation in the corresponding direction. In dimension i, $2 \leq i \leq D$, we use a locator code of length n_i over GF($2^{\frac{B}{b_i}}$) which locates the position of a burst of length b_i. The code of the first dimension finds the position of the error in the first dimension and the shape of the error, with a possible shift in each of the other $D-1$ dimensions. The code in dimension i, $2 \leq i \leq D$, finds the location of the position where the burst starts in dimension i. After each code discovers the position where the error starts in its dimension (note, that this can be done in parallel), we have the corresponding shift in the box error found by the first code and the error can be corrected. As before, we can use in dimension i, $2 \leq i \leq D$, a binary code which locates the position of a burst of length B in dimension i. The result is a code with redundancy at most $\lceil \log_2(n_1 n_2 \cdots n_D) \rceil + B + (D-2)\lceil \log_2 B \rceil + \lceil \log_2 b_1 \rceil$.

When B is even we have to modify our method in order to obtain similar results. The modifications include binary component codes in all dimensions, all the locator codes locate the position of a cyclic burst of size $B+1$, and we use colorings similar to the ones used in the next section.

Finally, we would like to mention that a generalization for codes over GF(q) is straightforward.

IV. Lee Spheres Cluster Errors

An error event at a position $(i_1, i_2, \ldots, i_D)$ can affect other positions. Assume that the error event can be spread up to radius R. Then any position $(j_1, j_2, \ldots, j_D)$ such that $\sum_{\ell=1}^{D} |j_\ell - i_\ell| \leq R$ might be erroneous. The set of positions $\{(j_1, j_2, \ldots, j_D) \; : \; \sum_{\ell=1}^{D} |j_\ell - i_\ell| \leq R\}$ forms a D-dimensional Lee sphere with radius R [5], [9]. The size of such sphere is $\sum_{i=0}^{\min\{D,R\}} 2^i \binom{D}{i}\binom{R}{i}$ [9]. The idea we are going to use is to transform the surface into another surface in such a way that a Lee sphere in one surface will be transformed into a box in the second surface. After that we will be able to use the encoding and decoding introduced in Section III. We start with the two-dimensional transformation.

Lemma 3: Let M, M^* be infinite two-dimensional surfaces and let T be the transformation from M into M^* defined by $T(i_1, i_2) = \left(\left\lceil \frac{i_1+i_2}{2} \right\rceil, i_2 - i_1\right)$. Then a Lee sphere with radius R in the surface M is located after the transformation T inside a rectangle of size $(R+1) \times (2R+1)$ in M^*.

Proof: A Lee sphere of radius R with center at (i'_i, i'_2) in the surface M includes the set of positions $B_R(i'_i, i'_2) = \{(i_1, i_2) \; : \; |i_1 - i'_1| + |i_2 - i'_2| \leq R\} = \{(i'_1 + R_i, i'_2 + R_j) \; : \; |R_i| + |R_j| \leq R\}$. $B_R(i'_1, i'_2)$ is transformed by T into the set of positions $B^*_R(i'_1, i'_2) = \left\{\left(\left\lceil \frac{i'_1+R_i+i'_2+R_j}{2} \right\rceil, i'_2 + R_j - i'_1 - R_i\right) \; : \; |R_i| + |R_j| \leq R\right\}$ of M^*. Denote, $i'^*_1 = \left\lceil \frac{i'_1+i'_2-R}{2} \right\rceil$, $i'^*_2 = i'_2 - i'_1 - R$, and we have that $B^*_R(i'_1, i'_2)$ is located inside the rectangle $\{(i'^*_1 + i_1, i'^*_2 + i_2) \; : \; 0 \leq i_1 \leq R, 0 \leq i_2 \leq 2R\}$. ■

The transformation T transforms a parallelogram into a rectangle (see Fig. 2 and 3) and hence we will need some adjustment in our encoding and decoding procedures if we want to correct Lee sphere clusters in a rectangular array rather than a parallelogram.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
-5					-5,4									
-4				-4,3	-4,4	-4,5								
-3			-3,2	-3,3	-3,4	-3,5	-3,6							
-2		-2,1	-2,2	-2,3	-2,4	-2,5	-2,6	-2,7						
-1	-1,0	-1,1	-1,2	-1,3	-1,4	-1,5	-1,6	-1,7	-1,8					
0	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9				
1		1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10			
2			2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10	2,11		
3				3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10	3,11	3,12	
4					4,4	4,5	4,6	4,7	4,8	4,9	4,10	4,11	4,12	4,13
5						5,5	5,6	5,7	5,8	5,9	5,10	5,11	5,12	5,13
6							6,6	6,7	6,8	6,9	6,10	6,11	6,12	
7								7,7	7,8	7,9	7,10	7,11		
8									8,8	8,9	8,10			
9										9,9				

Fig. 2. Positions in the parallelogram

0,0	-1,0	-1,1	-2,1	-2,2	-3,2	-3,3	-4,3	-4,4	-5,4
1,1	0,1	0,2	-1,2	-1,3	-2,3	-2,4	-3,4	-3,5	-4,5
2,2	1,2	1,3	0,3	0,4	-1,4	-1,5	-2,5	-2,6	-3,6
3,3	2,3	2,4	1,4	1,5	0,5	0,6	-1,6	-1,7	-2,7
4,4	3,4	3,5	2,5	2,6	1,6	1,7	0,7	0,8	-1,8
5,5	4,5	4,6	3,6	3,7	2,7	2,8	1,8	1,9	0,9
6,6	5,6	5,7	4,7	4,8	3,8	3,9	2,9	2,10	1,10
7,7	6,7	6,8	5,8	5,9	4,9	4,10	3,10	3,11	2,11
8,8	7,8	7,9	6,9	6,10	5,10	5,11	4,11	4,12	3,12
9,9	8,9	8,10	7,10	7,11	6,11	6,12	5,12	5,13	4,13

Fig. 3. Transformation to rectangle

The construction is generalized to D dimensions. The transformation T will work between two D-dimensional surfaces. For each entry $(i_1, i_2, \ldots, i_D)$ in the D-dimensional surface M, $T(i_1, i_2, \ldots, i_D) = (i^T_1, i^T_2, \ldots, i^T_D)$, where $i^T_1 = \left\lceil \frac{i_1+i_2}{2} \right\rceil$, $i^T_2 = \left\lceil \frac{-i_1+i_2+i_3}{2} \right\rceil, \ldots, i^T_j = \left\lceil \frac{(-1)^{j+1} i_1 + (-1)^j i_2 + \cdots - i_{j-1} + i_j + i_{j+1}}{2} \right\rceil$, for $1 \leq j \leq D-1$ and $i^T_D = \Sigma_{j=1}^{D} (-1)^{D-j} i_j$. We invoke first the two-dimensional transformation on the first two coordinates, then on the second and the third coordinates and so on.

Lemma 4: Let M be an infinite D-dimensional surface. Then a Lee sphere with radius R in the surface M is located after the transformation T inside a D-dimensional box of size $\underbrace{(R+1) \times (R+1) \times \cdots \times (R+1)}_{D-1 \text{ times}} \times (2R+1)$ in M^*.

As a consequence of this transformation we can use the construction of Section III for correction of D-dimensional box error. If we assume that our codewords are D-dimensional arrays rather than D-dimensional parallelograms we can use an array located inside the parallelogram. The redundancy will be increased by less than $(D+1)\log_2 D - D + 1$ bits.

An improvement for two dimensions is done by using the following construction. Lee spheres with radius R are used to tile the two-dimensional surface. Such tiling is given in [5], [9]. In each diagonal direction we have strips of Lee spheres. We color (with integers 0,1,2, and so on) the positions of the array in a way that in one of the diagonal directions all relative positions of the Lee spheres in the same strip have the same number; and in the other direction they are congruent modulo $b^* = 2R^2 + 2R + 1$ which is the size of a sphere (see Fig. 4). Another property is that the b^* numbers in a Lee sphere are consecutive integers. Such coloring is obtained by assigning the color $(R+1)i_1 + Ri_2$ to position (i_1, i_2), $0 \leq i_1, i_2 \leq n-1$. For each diagonal direction we choose a binary component code. For one diagonal we choose an optimum b^*-burst-correcting code. The k-th bit of a codeword in this code is the parity of all positions with color number k. For the other diagonal direction we choose a locator code which locates the position of a burst (or its cyclic shift) of length b^*. The codewords of this code are encoded similarly to the ones in Section III. Appropriate redundancy bits are chosen similarly to the constructions in Section III. If the code has size $n \times n$ then its redundancy is at most $\lceil \log_2 n^2 \rceil + b^* + \lceil 2\log_2(2R+1) \rceil$ and if each codeword is a rhombus of size n^2 then the redundancy is at most $\lceil \log_2 n^2 \rceil + b^* + \lceil \log_2 b^* \rceil$.

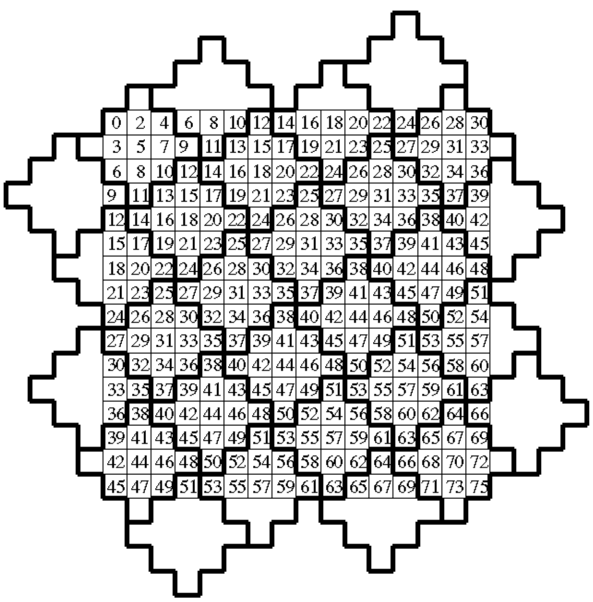

Fig. 4. Coloring of tiling

A similar idea can be used for D-dimensional code of size $n \times n \times \cdots \times n$ correcting Lee sphere error with radius one. We use D components codes, each one is an optimum $(2D+1)$-burst-correcting code of length $n^* = 1 + \frac{D(D+1)}{2}(n-1)$. Each code uses a different coloring of the array. For a position $(i_1, i_2, \ldots, i_D)$, $0 \leq i_\ell \leq n-1$, $1 \leq \ell \leq D$, for the k-th coloring, $1 \leq k \leq D$, we assign the color $\Sigma_{\ell=1}^{D} \ell \cdot i_{k+\ell-1}$, where indices are residues modulo D between 1 and D. Again, for each coloring, all the numbers in a Lee sphere are consecutive integers. Bit s of the k-th component code is a parity bit for all the bits colored with the integer s, by the k-th coloring, in the D-dimensional codeword. Note that each color in each coloring forms a perfect code [9] when we consider the coloring in $\mathbb{Z}^D$ and not just in the array. The excess redundancy in this case is quadratic in D, compared to exponential in D if we use the transformation T and the code which corrects D-dimensional box error. Further improvement can be made if we use for some dimensions locator codes instead of optimum $(2D+1)$-burst-correcting codes.

Finally, we want to design a code which corrects an arbitrary D-dimensional cluster-error of size b. If b is odd then the cluster is located inside a Lee sphere with radius $\frac{b-1}{2}$. If $b = 2R$ then after applying the transformation T the cluster is located inside a D-dimensional box of size $\underbrace{(R+1) \times \cdots \times (R+1)}_{D-1 \text{ times}} \times (2R)$. For a two-dimensional cluster we can use the same tiling as for Lee spheres with radius $R = \frac{b}{2}$. If $b = 3$ and $D \geq 3$ then we can use the codes based on correction of Lee spheres errors with radius one. These ideas are improved by using bursts with limited weight. Also, some improvements in the redundancies of some of the constructions are made by using some more ideas. These results and complete description of some of the briefly mentioned ideas will appear in the full version of this paper.

References

[1] K. A. S. Abdel-Ghaffar, "An information- and coding-theoretic study of bursty channels with applications to computer memories", *Ph.D. dissertation, California Inst. Technol. Pasadena, CA*, June 1986.

[2] K. A. S. Abdel-Ghaffar, "On the existence of optimum cyclic burst-correcting codes over GF(q)", *IEEE Trans. on Inform. Theory*, vol. IT-34, pp. 329-332, March 1988.

[3] K. A. S. Abdel-Ghaffar, R. J. McEliece, A. M. Odlyzko, H. C. A. van Tilborg, "On the existence of optimum cyclic burst-correcting codes", *IEEE Trans. on Inform. Theory*, vol. IT-32, pp. 768-775, Nov. 1986.

[4] K. A. S. Abdel-Ghaffar, R. J. McEliece, H. C. A. van Tilborg, "Two-dimensional burst identification codes and their use in burst correction", *IEEE Trans. on Inform. Theory*, vol. IT-34, pp. 494-504, May 1988.

[5] M. Blaum, J. Bruck, A. Vardy, "Interleaving schemes for multidimensional cluster errors", *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 730–743, March 1998.

[6] I. M. Boyarinov, "Two-dimensional array codes correcting rectangular burst errors", *Prob. of Infor. Transmission,* vol. 42, pp. 26–43, June 2006.

[7] M. Breitbach, M. Bossert, V. Zyablov, V. Sidorenko, "Array codes correcting a two-dimensional cluster of errors", *IEEE Trans. on Inform. Theory*, vol. IT-44, pp. 2025-2031, September 1998.

[8] T. Etzion, A. Vardy, "Two-dimensional interleaving schemes with repetitions: Constructions and bounds", *IEEE Trans. Inform. Theory*, vol. IT-48, 428–457, February 2002.

[9] S. W. Golomb and L. R. Welch, "Perfect codes in the Lee metric and the packing of polyominos", *SIAM Journal Applied Math.*, vol. 18, pp. 302–317, 1970.

[10] H. Imai, "Two-dimensional Fire codes", *IEEE Trans. on Inform. Theory*, vol. IT-19, pp. 796-806, 1973.

[11] S. H. Reiger, "Codes for correction of 'clustered' errors", *IRE Trans. on Inform. Theory*, vol. IT-6, pp. 16-21, 1960.

[12] M. Schwartz and T. Etzion, "Two-dimensional cluser-correcting codes", *IEEE Trans. on Inform. Theory*, vol. IT-51, pp. 2121-2132, June 2005.