# On successive packing approach to multidimensional ($M$-D) interleaving

Xi Min Zhang* Yun Q. Shi† Sankar Basu‡

**Abstract**

We propose an interleaving scheme for multidimensional ($M$-D) interleaving. To achieved by using a novel concept of basis interleaving array. A general method of obtaining a variety of basis interleaving arrays is presented. Based on the basis interleaving array, we then propose an interleaving technique, called successive packing, to generate the interleaved array of arbitrary size. It is shown that the proposed technique can spread any error burst of $m_0^k \times m_1^k$ within $m_0^n \times m_1^n$ array $(1 \leq k \leq n-1)$ effectively so that the error burst can be corrected with simple random error correcting-code (provided the error correcting-code is available). It is further shown that the technique is optimal for combating a set of arbitrarily-shaped error bursts. Since this algorithm needs to be implemented only once for a given $M$-D array, the computational cost is is low.

**Key words:** Basis interleaving array, Multidimensional interleaving, error burst, random-error-correction codes

## 1 Introduction

With rapid development of the information technology, two-dimensional (2-D) and three-dimensional (3-D) data handling is being widely used. Applications include 2-D and 3-D magnetic and optical data storage, charged-coupled devices (CCDs), 2-D barcodes, and information hiding in digital images and video sequences. Correcting error bursts is an important problem in all of these applications. Thus, the issue of reliability of $M$-D information is an important task, with both theoretical and practical significance. In this paper, we address the protection of multidimensional ($M$-D, $M \geq 2$) digital data. Specifically, we show how to spread the bursts (clusters) of errors in such a way that they can be corrected by simple error correction code.

*ECE Dept, New Jersey Institute of Technology, Newark, NJ 07102.
†ECE Dept, New Jersey Institute of Technology, Newark, NJ 07102.
‡Room 2SA05, IBM T. J. Watson Research Center, New York 10532.

One-dimensional (1-D) interleaving technique has been well documented in the literature (see e.g., [1]). The main idea is to shuffle the code symbols from different codewords so that error bursts encountered in the transmission are spread across multiple codewords when the codewords are reconstructed at the receiving end. Consequently, the error occurring within one codeword may be small enough to be corrected by using simple random-error-correction code. Extending this strategy of one-dimensional (1-D) interleaving technique to the $M$-D situation in order to combat error bursts with some random-error-correction codes has become the most common approach to the correction of error bursts. Some $M$-D interleaving techniques for combating $M$-D error bursts have been proposed in [2, 3, 4, 5]. Among them, Almeida et.al. present 2-D interleaving results for circular-shape error bursts [2]. Their results cannot be generalized to non-circular shaped bursts or to higher dimensions. The United Parcel Service (UPS) combine the 1-D interleaving technique with a writing procedure to protect 2-D barcode [3]. However, this approach cannot effectively spread the 2-D error bursts [6, 7]. Adbel-Ghaffar [4] studies some theoretical aspects of 2-D interleaving, but only presents unproven concepts.

A more comprehensive interleaving technique is discussed in [5], in which an error burst is defined as an arbitrarily-shaped, connected area volume in the multidimensional space. In this method, for each burst size $t_0$, a specific algorithm is implemented, which can optimally correct arbitrarily-shaped error burst of size $t_0$. Furthermore, it is observed that when the burst size $t$ increases, i.e., when $t > t_0$, the algorithm with a set of new parameters needs to be implemented in order to correct the larger error burst of arbitrary shape. Likewise, when the burst size decreases, i.e., $t < t_0$, the interleaving array that is optimal for burst size $t_0$ is not optimal any more. Since, in practice, the sizes of error bursts are not known in advance, application of the technique is somewhat limited. By contrast, the size of a given 2-D array (e.g., the size of image and video frames) is known in many applications. Motivated by these observations, a novel method, called successive packing (SP), to 2-D interleaving, is proposed as a different and complementary technique [7, 8]and optimal performance of SP on square arrays of size $2^n \times 2^n$ is proved. However, the analysis and application of SP is restricted to square arrays of size $2^n \times 2^n$.

In this paper, we first propose the novel concept of basis interleaving array. Its characteristics are discussed and a method for its construction is proposed. We then propose to generate a large class of interleaving arrays by successive packing of basis interleaving arrays. The performance of the proposed scheme in burst error correction is discussed. The paper is organized as follows. In Section 2, we introduce definitions necessary for the remaining part of the paper. We then present the concept of basis interleaving array for $M$-D interleaving, and the corresponding 2-D method is shown to be optimal. Next, in Section 3, we propose the SP approach to $M$-D interleaving. We show that it works well in a set of error bursts when size of the 2-D array is given, but size of error bursts is not known in advance. Finally, conclusions are drawn in Section 4.

## 2 Basis Interleaved Array

Unless otherwise stated, for the sake brevity of discussion, our presentation will be restricted to one-random-error-correction codes. All results can be extended to $r$-random-error-correction codes with $r > 1$ in a straightforward manner. The philosophy behind interleaving to combat bursts of $M$-D errors is similar to that in the 1-D situation. Loosely speaking, with interleaving, the elements in an $M$-D array are rearranged so that error in the interleaved $M$-D array are separated as far away as possible from in the de-interleaved array. Error burst correction is, thus, facilitated if there is only one error in each codeword in the de-interleaved $M$-D array.

**Definition 2.1.** *Let $C$ be an $M$-D code of $m_0 \times m_1 \cdots \times m_{M-1}$ over GF(q). A codeword of $C$ is an $M$-D array of $m_0 \times m_1 \cdots \times m_{M-1}$, with each element of the $M$-D array assigned with a code symbol.*

Note that $GF(q)$ denotes Galois field over $q$ elements. The simplest field is the binary field, $GF(2) = \{0, 1\}$.

**Definition 2.2.** *In 2-D arrays, the neighbors of element $(x, y)$ are denoted by*

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

*In 3-D arrays, the neighbors of element (x,y,z) are denoted by*

$$(x + 1, y, z), (x - 1, y, z), (x, y + 1, z), (x, y - 1, z), (x, y, z + 1), (x, y, z - 1)$$

*provided those elements exist.*

Natural extensions of Definition 2.2 in higher dimensions apply. For $M$-D arrays, an element has $2M$ neighbors.

**Definition 2.3.** *A burst is a subset of the given $M$-D array $B$, in which any element has at least one neighbor contained in $B$. Its size is defined as the number of elements in $B$.*

**Definition 2.4.** *The distance between any two elements is the length of the shortest path between the two elements. Here, path consists of a squence of neighbors connecting the two elements.*

Since interleaving involves shuffling code symbols so that each element in an error burst is spread into a different codeword, if any two elements within a distinct codeword are separated in the de-interleaved array such that their distance is maximized, then a large error bursts can be hopefully corrected.

Let $A$ be an $M$-D array of size $m_0 \times m_1 \times \cdots \times m_{M-1}$. We re-index each element $s_{i_0,i_1,\cdots,i_{M-1}}$ of $A$ as $s_k$ with $k$ being a function of $i_0$ to $i_{M-1}$ (for instance, for 2-D array, we can have $k = m_1 \times i_0 + i_1$). Now consider a partitionin of A into L blocks with $1 \leq L \leq N$, where $N = m_0 \times m_1 \times \cdots \times m_{M-1}$. That is, each block so generated contains $K = N/L$ elements.

**Definition 2.5.** *In the above scheme of partitioning followed by reindexing, any element having an index $k$ with $K(d-1) \leq k < Kd$, $1 \leq d \leq L$ is said to belong to the d-th block. An element having index $K(d-1)$ is referred to as the beginning element of the d-th block. All elements belonging to the same block are referred to as the $K$-equivalent elements.*

According to Definition 2.5, we see that $s_{2l}$, $s_{2l+1}$ are 2-equivalent elements; $s_{3l}$, $s_{3l+1}$, $s_{3l+2}$ are 3-equivalent elements. It is obvious that $K_1$-equivalent elements are also $K_2$-equivalent elements if $K_2/K_1$ is integer. Let one block be a codeword with length K, then all elements of a distinct codeword is K-equivalent with each other. Hence, the objective of effective interleaving is transformed to the problem of maximizing the minimum distance between any two $K$-equivalent elements. If, for each $k = 0$ to $M-1$, $m_k$ is prime, then the number of codewords that the corresponding $M$-D array can contain is an integer multiple of $m_n$ $(n < M)$. Motivated by this observation, we propose the concept of *basis interleaving array* next.

## 2.1 Square 2-D basis array

**Definition 2.6.** *Consider a interleaving array $B$ of size $m \times m$, where $m$ is prime. If the minimum distance between any two $m$-equivalent elements attain the maximum, then we call this array as* basis interleaving array.

It is obvious from Definition 2.6 we have square basis arrays of sizes $2 \times 2$, $3 \times 3$, $5 \times 5$, $\cdots$ etc. In [7, 8] an optimal interleaving technique based on the succesive packing of a specific $2 \times 2$ array is presented. In fact, this particular $2 \times 2$ array is an example of a basis interleaving array.

**Theorem 2.1.** *The 2-D array*

$$\begin{bmatrix} S_0 & S_2 \\ S_3 & S_1 \end{bmatrix} \tag{2.1}$$

*is a basis interleaving array.*

*Proof.* In a $2 \times 2$ array, the distance from one corner to its opposite corner is the maximum distance between any two elements. It is obvious that this distance equal to 2. For the two 2-equivalent element pairs $(s_0, s_1)$ and $(s_2, s_3)$, it is easily seen the distance between $s_0$ and $s_1$, as well as the distance between $s_2$ and $s_3$ attains the value 2. Thus, Theorem 2.1 is proved. □

In order to construct a basis interleaving array, it is necessary to know the upper bound of the minimum distance. Note that the number of $m$-equivalent elements of each element is $m-1$ in a square $m \times m$ array. Thus, we need to constitute a 2-D sphere with size $m$ centered around each of the $m$-equivalent elements. The $m$ spheres should be able to tile to a $m \times m$ array without overlapping. Then the maximum radius of this sphere is the upper bound of the minimum distance. This problem was first approached in [9] for $m$ is odd.

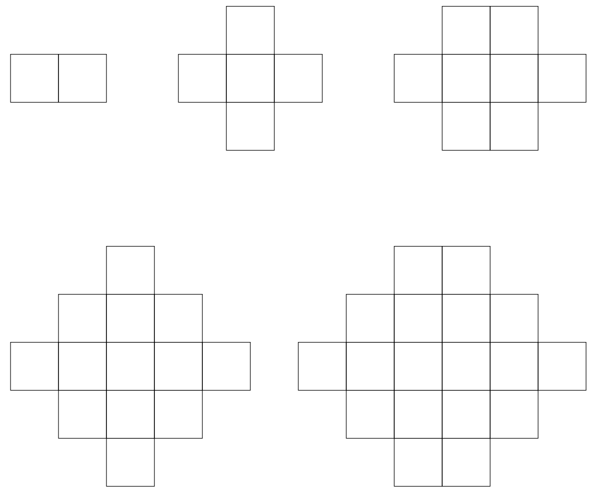

Figure 1: Typical 2-D sphere with size 2, 5, 8, 13, 18.

Later the idea was extended in [5] to even $m$ case. It has been proven that if $m$ is even and $m = t^2/2$, then the upper bound of the radius is $t$; if $m$ is odd and $m = (t^2+1)/2$, then the upper bound is also $t$.

Some examples of 2-D spheres are shown in Fig. 1. Notice that spheres of size $m$ do not exist for values of $m$ equal to $3, 4, 6, 7, 9, 10, 11, \cdots$. Thus, the upper bound of the minimum distance is the radius of the largest sphere with size less than $m$. According to this observation, the upper bound is 2 for $2 \le m \le 4$; the upper bound is 3 for $5 \le m \le 7$; upper bound is 4 for $8 \le m \le 12$, etc.

In the following we present a method of constructing square basis interleaving arrays.

**Procedure 2.1.** *Let $A$ be a 2-D array of size $m \times m$ ($m \ge 2$), and let $d_r$ be the upper bound of the distance between any two elements. We express coordinate $(i, j)$ of each element toroidally i.e., modulo integer $m$. We first attribute the elements of first row as $s_0, s_m, s_{2m}, \cdots, s_{m(m-1)}$, which are in the location $(0,0), (0,1), \cdots, (0, m-1)$. Then, let $X = 1$ and $Y = d_r - 1$. For each element with location (i, j), add 1 to the subscript of this element and put it in the location (i+X, j+Y). For example, $s_1$ is put in the location (X, Y). Repeat this procedure until all of the positions are occupied.*

*Example 2.1.1*: Consider the case of $2 \times 2$ array. According to [5], we have $d_r = 2$. Thus, $X = 1, Y = 1$. Using the above procedure with $X = 1, Y = 1$, we have constructed the array as in Fig.2. It can be seen that it is exactly same with our $2 \times 2$ basis array above.

| | |
|---|---|
| $s_0$ | $s_2$ |
| $s_3$ | $s_1$ |

Figure 2: $2 \times 2$ basis interleaving array

*Example 2.1.2*: Consider the case of $3 \times 3$ array. According to [5], we have $d_r = 2$. Thus,

$X = 1, Y = 1$. Using the above procedure with $X = 1, Y = 1$, we have constructed the array as in Fig. 3.

| | | |
|---|---|---|
| $s_0$ | $s_3$ | $s_6$ |
| $s_7$ | $s_1$ | $s_4$ |
| $s_5$ | $s_8$ | $s_2$ |

Figure 3: $3 \times 3$ basis interleaving array

*Example 2.1.3*: Consider the case of $5 \times 5$ array. According to [5], we have $d_r = 3$. Thus $X = 1, Y = 2$. Using the above procedure with $X = 1, Y = 2$, we have constructed the array as in Fig. 4.

| | | | | |
|---|---|---|---|---|
| $s_0$ | $s_5$ | $s_{10}$ | $s_{15}$ | $s_{20}$ |
| $s_{16}$ | $s_{21}$ | $s_1$ | $s_6$ | $s_{11}$ |
| $s_7$ | $s_{12}$ | $s_{17}$ | $s_{22}$ | $s_2$ |
| $s_{23}$ | $s_3$ | $s_8$ | $s_{13}$ | $s_{18}$ |
| $s_{14}$ | $s_{19}$ | $s_{24}$ | $s_4$ | $s_9$ |

Figure 4: $5 \times 5$ basis interleaving array

**Theorem 2.2.** *If the integer $m$ is prime, then the square array constructed by Procedure 2.1 is a basis interleaving array.*

*Proof.* According to Procedure 2.1, we first generate the positions for elements $s_0, s_m, s_{2m}, \cdots, s_{m(m-1)}$. Then we generate the positions of their $m$-equivalent elements respectively. It can be seen that the corresponding distance of two co-positional $m$-equivalent elements in each of the $m$-equivalent set is the same. For example, the distance between $s_0$ and $s_k$ is equal to the distance between $s_{2m}$ and $s_{2m+k}$, where $k < m$. Therefore, if we can prove Theorem 2.2 for the $m$-equivalent set beginning with $s_0$, then Theorem 2.2 is proved. Based on the same reasoning, if we can prove that the distance between $s_0$ and any its $m$-equivalent elements is greater than or equal to $d_r$, then it holds for any $s_k$ with $0 < k < m$.

We first consider the case when $d_r$ is odd. Due to toroidal labeling of the coordinates, the coordinates of $s_k$ is $(kX, kY \mod m)$. The distance between $s_0$ and $s_k$ is $kX + kY \mod m$. Thus, the problem is transformed to proving the following inequality:

$$kX + kY \mod m \geq d_r \tag{2.2}$$

If $kY < m$ then we have $kY \mod m = kY$. Thus, $kX + kY \mod m = k(X + Y)$. Since $X + Y = d_r$, it is obvious that $k(X + Y) > d_r$. Now, let us consider the case that $kY > m$.

Then for integers $l_1$ and $l_2$ we can write

$$kY = l_1 m + l_2.$$

Using the above equation, we can obtain

$$k = \frac{l_1 m + l_2}{Y}.$$

Since $X = 1$, $Y = d_r - 1$, we have

$$kX + kY \mod m = \frac{l_1 m + l_2}{d_r - 1} + l_2.$$

Hence, the inequality (2.2) becomes

$$\frac{l_1 m + l_2}{d_r - 1} + l_2 \geq d_r, \tag{2.3}$$

which is satisfied if

$$l_1 m \geq d_r^2 - (l_2 + 1)d_r - l_2.$$

According to a result in [5], we have $m \geq \frac{d_r^2+1}{2}$. Thus, if

$$l_1 \frac{d_r^2 + 1}{2} \geq d_r^2 - (l_2 + 1)d_r - l_2$$

holds, then inequality (2.3) holds. The above inequality clearly holds for $l_1 > 1$. If $l_1 = 1$, then it follows that

$$\frac{d_r^2 + 1}{2} \geq d_r^2 - (l_2 + 1)d_r - l_2.$$

Thus, the value of $l_2$ should satisfy the following condition

$$l_2 > \frac{d_r - 2}{2}.$$

In order to find the range of allowable values of $l_2$ when $l_1 = 1$, let us decompose $m$ as

$$\begin{aligned} m &= \frac{d_r^2 + 1}{2} \\ &= (d_r - 1)\frac{d_r + 1}{2} + 1 \end{aligned} \tag{2.4}$$

Now, if we let $k = \frac{d_r+1}{2} + 1$ then

$$\begin{aligned} kY &= (d_r - 1)\frac{d_r + 1}{2} + 1 + (d_r - 2) \\ &= m + (d_r - 2) \end{aligned} \tag{2.5}$$

Hence we get $l_2 = d_r - 2$. According to the above procedure, $d_r - 2$ is the minimum of $l_2$ when $l_1 = 1$. Thus, the theorem is proved for the $d_r$ odd case. Similar reasoning applies when $d_r$ is even. □

In Procedure 2.1, we proposed a technique for generating the basis interleaving array. Next, we generalize this method to any $m \times m$ array such that the minimum distance between any two $m$-equivalent elements attains the maximum value.

**Procedure 2.2.** *Let $A$ be a 2-d array of size $m \times m$ ($m \geq 2$), and let $d_r$ be the upper bound on the distance between any two elements of the array. We label the coordinates of the array toroidally on $m$ i.e., we replace the corordinate $(i, j)$ with their values modulo $m$. We first attribute the elements of first row as $s_0, s_m, s_{2m}, \cdots, s_{m(m-1)}$, which are in the locations $(0,0), (0,1), \cdots, (0, m-1)$. Then let $X = 1$, $Y$ satisfy the condition $d_r - 1 \leq Y \leq d_r$, with $Y$ and $m$ relatively prime. For each element with location $(i, j)$, add 1 to the subscript of the element and put it in the location (i+X, i+Y). For example, $s_1$ is put in the location $(X, Y)$. Repeat this procedure until all of the positions are occupied.*

**Theorem 2.3.** *For any integer $m > 1$, the square array constructed by Procedure 2.2 attains the maximum in the sense of minimum distance between any two $m$-equivalent elements.*

Proof of Theorem 2.3 is similar to proof of Theorem 2.2. Notice that Construction 2.1 in [5] is a special case of Procedure 2.2, where $Y = b_r$ for $m = \frac{b_r^2+1}{2}$, and $Y = b_r + 1$ for $m = \frac{b_r^2}{2}$.

## 2.2 Rectangular basis interleaving array

Next, we generalize the results of the previous section to rectangular basis arrays. We have the following result.

**Theorem 2.4.** *Let $m \times n$ be a rectangular basis array. If $m < n$, then the upper bound of the minimum distance between any two $n$-equivalent elements is the same as the minimum distance of any two $m$-equivalent elements in a $m \times m$ basis array. If $m > n$, then the upper bound of the minimum distance between any two $m$-equivalent elements is the same as the minimum distance of any two $n$-equivalent elements in a $n \times n$ basis array.*

*Proof.* To prove Theorem 2.4, we first prove that the upper bound of its minimum distance cannot be greater than the corresponding square basis array. Then we show that the equality can be obtained.

Let us assume for definiteness that the minimum distance is greater than the corresponding square basis array, and the minimum distance of any two $n$-equivalent elements greater than the minimum distance of $m \times m$ basis array. Let us then truncate the $m \times n$ array to $m \times m$ array. According to our assumption, the newly obtained $m \times m$ array will have the minimum distance larger than the $m \times m$ basis interleaving array, which contradicts the definition of basis interleaving array. Hence, the minimum distance of the $n$-equivalent elements in the $m \times n$ array cannot be larger than the corresponding $m \times m$ square basis interleaving array. To obtain the same minimum distance as the square basis interleaving array, we can change Procedure 2.1 slightly to generate the rectangular basis interleaving array. □

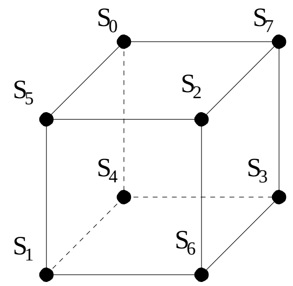


Figure 5: 3-D $2 \times 2 \times 2$ basis interleaving array.

**Procedure 2.3.** *Let $A$ be a 2-D array of size $m \times n$, and let $d_r$ be the upper bound of the distance for the corresponding square interleaving array. If $m < n$, we re-express $i$ of row coordinates $(i, j)$ toroidally as $i \mod m$. We first attribute the elements of first row as $s_0, s_n, s_{2n}, \cdots, s_{(m-1)n}$, which are in the location $(0,0), (0,1), \cdots, (0, m-1)$. Then let $X = d_r - 1$, $Y = 1$. For each element with location (i, j), add 1 to the subscript of the element and put it in the location (i+X, j+Y). For example, $s_1$ is put in the location (X, Y). Repeat this procedure until all of the positions are occupied. If $m > n$, an analogous procedure is followed.*

It is easy to see by using arguments as in the proof of Theorem 2.2, that Procedure 2.3 yields the same minimum distance as the corresponding square array.

## 2.3 3-D basis interleaving array

In this section, we attempt to briefly describe via some examples how to extend the results of the previous section to higher dimensions.

**Definition 2.7.** *Consider a 3-D interleaving array $B$ of size $l \times m \times n$, where $l, m, n$ are prime, and $l \leq m \leq n$. If the minimum distance between any two mn-equivalent elements attains the maximum, then we call this array a basis interleaving array.*

For a square 3-D array of size $m \times m \times m$ with minimum distance $d_r$, it has been proven that $m$ is bounded by

$$m \geq \frac{d_r^3 + 2d_r}{6}, \quad \text{for } d_r \text{ even},$$

$$m \geq \frac{d_r^3 + 5d_r}{6}, \quad \text{for } d_r \text{ odd},$$

where $m$ is the size of 3-D sphere [5]. This result is further extended for $M$-D arrays in [5]. However, this bound cannot be always to achieve for $M \geq 3$. For details we refer to [9, 5]. Here, we present an example of a 3-D basis interleaving array of size $2 \times 2 \times 2$ , which will be useful in our discussions to follow.

# 3 Successive Packing of Basis Interleaving Array

The initial idea of successive packing for interleaving was presented in [7, 8], where the authors focus on the interleaving of arrays of size $2^n \times 2^n$. Whether it can applied to $M$-D array with arbitrary size had not been investigated. Also, it is not clear if its optimal performance holds for rectangular 2-D arrays. In this section, we first present an $M$-D interleaving technique based on the successive packing of basis interleaving arrays. Then its performance for spreading error burst is analyzed. Subsequently, its optimality is discussed and proved.

## 3.1 Successive Packing

Now we discuss the proposed SP technique in $M$-D case.

**Procedure 3.1. ($M$-D interleaving using the successive packing)**
*Consider an $M$-D basis interleaving array of size $m_0 \times m_1 \times \cdots \times m_{M-2} \times m_{M-1}$. The interleaving array is the original basis interleaving array itself. When $m_0 = m_1 \cdots = m_{M-1} = 1$, it is*

$$S_1 = [s_0] \tag{3.6}$$

*where $s_0$ represents the element in the array, and $S_1$ the array. The subscript in $S_1$ represents the total number of elements in the interleaving array. Given interleaving array $S_N$ of size $N = m_1 \times m_2 \times \cdots \times m_{M-2} \times m_{M-1}$, the interleaving array $S_{N^2}$ can be generated by transferring each element $s_i$ in $S_N$ to a $M$-D array according to the operation $N \times S_N + i$ (this operation is described further in the following). This packing procedure is carried out successively to generate $S_{N^K}$ by transferring each element $s_i$ in $S_{N^K}$ to a $M$-D array according to the operation $N \times S_{N^{K-1}} + i$.*

In the above procedure, the operation $N \times S_N + i$ is the key point. Generalizing what we presented in [7], operation $N \times S_N + i$ generates a $M$-D array with the same dimensionality as $S_N$. Furthermore, each element in $N \times S_N + i$ is indexed in such a way that its subscript equals to the N times of that of the corresponding elements (i.e., elements occupying the same position in the $M$-D array) in $S_N$ plus $i$. A few examples are presented next.

*Example 3.1*: Given a 1-D basis array $S_3 = \{s_0, s_1, s_2\}$, the interleaving array is $S_9 = \{s_0,$ $s_3$, $s_6$, $s_1$, $s_4$, $s_7$, $s_2$, $s_5$, $s_8\}$.

*Example 3.2*: Given a $3 \times 3$ basis array as in Fig.3, the $9 \times 9$ interleaving array is generated as in Fig.6.

*Example 3.3*: Given a $2 \times 2 \times 2$ basis array as in Fig.5, the $4 \times 4 \times 4$ interleaving array is generated as in Fig.7, whereas the left hand side displays $2 \times 2 \times 2$ array obtained via the operation $8 \times S_8 + 5$.

To generate a interleaving array with arbitrary size, we use the successive packing method based on a combination of different basis interleaving arrays. For instance, given basis

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $S_0$ | $S_{27}$ | $S_{54}$ | $S_3$ | $S_{30}$ | $S_{57}$ | $S_6$ | $S_{33}$ | $S_{60}$ |
| $S_{63}$ | $S_9$ | $S_{36}$ | $S_{66}$ | $S_{12}$ | $S_{39}$ | $S_{69}$ | $S_{15}$ | $S_{42}$ |
| $S_{45}$ | $S_{72}$ | $S_{18}$ | $S_{48}$ | $S_{75}$ | $S_{21}$ | $S_{51}$ | $S_{78}$ | $S_{24}$ |
| $S_7$ | $S_{34}$ | $S_6$ | $S_1$ | $S_{28}$ | $S_{55}$ | $S_4$ | $S_{31}$ | $S_{58}$ |
| $S_{70}$ | $S_{16}$ | $S_4$ | $S_{64}$ | $S_{10}$ | $S_{37}$ | $S_{67}$ | $S_{13}$ | $S_{40}$ |
| $S_{52}$ | $S_{79}$ | $S_{25}$ | $S_{46}$ | $S_{73}$ | $S_{19}$ | $S_{49}$ | $S_{76}$ | $S_{22}$ |
| $S_5$ | $S_{32}$ | $S_{59}$ | $S_8$ | $S_{35}$ | $S_{62}$ | $S_2$ | $S_{29}$ | $S_{56}$ |
| $S_{68}$ | $S_{14}$ | $S_{41}$ | $S_{71}$ | $S_{17}$ | $S_{44}$ | $S_{65}$ | $S_{11}$ | $S_{38}$ |
| $S_{50}$ | $S_{77}$ | $S_{23}$ | $S_{53}$ | $S_{80}$ | $S_{26}$ | $S_{47}$ | $S_{74}$ | $S_{20}$ |

Figure 6: Successive Packing generated $9 \times 9$ interleaving array.

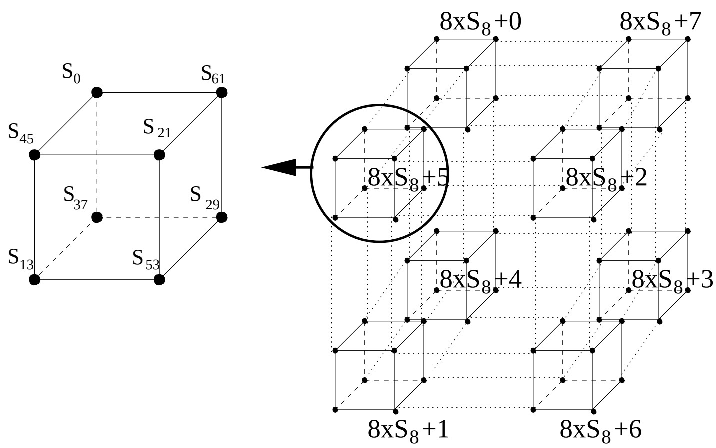


Figure 7: Successive Packing generated $4 \times 4 \times 4$ interleaving array.

interleaving array $S_N$ and $S_2$, we can generate the interleaving array $S_{2N}$ by $\{S_N \times 2 + 0, S_N \times 2 + 1\}$.

*Example 3.4*: Given $2 \times 2$ basis array as in Fig.2 and $3 \times 3$ basis array as in Fig.3, the $6 \times 6$ interleaving array is generated as in Fig.8.

## 3.2 Performance Analysis

Before embarking on performance analysis of our SP based $M$-D interleaving technique, we introduce the following definition.

**Definition 3.1.** *Consider two bursts $B_1$ and $B_2$ having the same size and shape in an interleaving $M$-D array. If each element in a burst (e.g., $B_1$) is either an element of another burst (e.g., $B_2$), or K-equivalent of an element of another burst ($B_2$), then we say that bursts $B_1$ and $B_2$ are K-equivalent bursts.*

In the remainder of the paper, when discussing error burst correction, we may consider

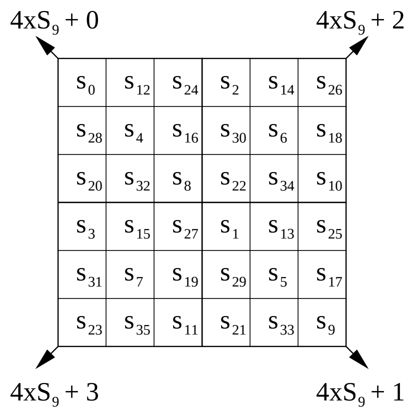


Figure 8: Successive Packing generated $6 \times 6$ interleaving array.

each set of equivalent elements defined in Definition 2.5 as a $M$-D codeword. This implies that a codeword consists of a set of consecutive code symbols. This is necessary since we need to discriminate code symbols within a codeword in our ensuing discussion of the SP technique for $M$-D interleaving. An error burst (in the interleaved array) is said to be spread, and can be corrected with one-random-error-correction codes, if each element in the burst is spread in distinct codewords of the de-interleaved array. From this point of view, it is easy to see that given two equivalent bursts, if one is interleaved then the other burst must have also been interleaved.

We may now state the following results.

**Lemma 3.1.** *Let $A$ be a 2-D array of size $m_0^n \times m_1^n$ obtained by using successive packing of a basis interleaving array of size $m_0 \times m_1$. Then all bursts of size $m_0^k \times m_1^k$ in $A$ with $k < n$ are $K_1$-equivalent, where $K_1 = (m_0 \times m_1)^{n-k}$.*

The proof of this lemma is ommitted for brevity. Now we are in a position to present the following theorem.

**Theorem 3.1.** *Consider a 2-D array $A$ of size $m_0^n \times m_1^n$. Then any burst of size $m_0^m \times m_1^m$ with $m \leq n$ in the interleaving array $A$ obtained by using the successive packing is spread in the de-interleaved array so that each element of the burst falls into a distinct block of size $m_0^{n-m} \times m_1^{n-m}$.*

Theorem 3.1 indicates that, if a distinct code symbol is assigned to each element in a block (cf. to Definition 2.5), and all the code symbols associated with an individual $K$-equivalent class form a distinct codeword, then burst error correction with a one-random-error-correction code is guaranteed, provided the code is available. Furthermore, the interleaving degree equals the size of the burst error, hence minimizing the number of codewords required in an interleaving scheme. In other words, with the successive packing technique, the interleaving degree attains the lower bound (the interleaving gain)[1]. In this sense, the successive packing

[1]see [8] for definition of interleaving degree and interleaving gain.

interleaving technique is optimal. Note that discussions in [8] can be considered to be a special case of Theorem 3.1 with $m_0 = 2, m_1 = 2$. We conjecture that following result hold.

**Conjecture 3.1.** *Consider an $M$-D array $A$ of size $m_0^n \times m_1^n \times \cdots \times m_{M-1}^n$. Then any burst of size $m_0^m \times m_1^m \times \cdots \times m_{M-1}^m$ with $m \leq n$ in the interleaving array $A$ obtained by using the successive packing is spread in the de-interleaved array so that each element of the burst falls into a distinct block of size $m_0^{n-m} \times m_1^{n-m} \times \cdots \times m_{M-1}^{n-m}$.*

In Section 3.1, we proposed to generate arbitrary size interleaved array by combining different basis interleaving arrays. Here, we first show how to generate a square 2-D array of size $2m^2 \times 2m^2$. We then prove its optimal performance. The procedure can be described as follows.

**Procedure 3.2.**

- *Generate the $m \times m$ interleaving array according to our SP technique.*
- *Generate the $2m \times 2m$ interleaving array as follows*

$$S_{4m^2} = \begin{bmatrix} 4 \times S_{m^2} + 0 & 4 \times S_{m^2} + 2 \\ 4 \times S_{m^2} + 3 & 4 \times S_{m^2} + 1 \end{bmatrix} \tag{3.7}$$

- *Let $l_{i,j}$ denote the subscript of the corresponding element in the 2-D interleaving of array $S_{4m^2}$ of size $2m \times 2m$. The $2m^2 \times 2m^2$ interleaving array is generated as*

$$S_{4m^4} = \begin{bmatrix} m^2 \times S_{4m^2} + l_{0,0} & \cdots & m^2 \times S_{4m^2} + l_{0,m-1} \\ \vdots & & \vdots \\ m^2 \times S_{4m^2} + l_{m-1,0} & \cdots & m^2 \times S_{4m^2} + l_{m-1,m-1} \end{bmatrix} \tag{3.8}$$

We first need the following lemma.

**Lemma 3.2.** *Let $C$ be a cluster of size $2m$ in a 2-D array of size $m_1 \times m_1$, where $2m < m_1$. Then there must exist a rectangular block $R_1$ of size $2m \times m$, and/or a rectangular block $R_2$ of size $m \times 2m$ such that $C$ is entirely contained in either $R_1$ or $R_2$, or in both.*

*Proof.* For the purpose of establishing a contradiction, let us hypothesize that there do not exist blocks $R_1$ and $R_2$ as in the statement of the Lemma entirely containing $C$. Then, $C$ would be outside of $R_1$ either in the $X$, or in the $Y$ direction. Since the length of $R_1$ in $Y$ direction is $2m$, which is equal to the size of $C$, it is only possible for $C$ to be outside of $R_1$ in $X$ direction. Hence we have $C_X > m$, where the $C_X$ is the dimension of $C$ along $X$ direction. Since $C$ is not entirely contained in $R_2$, based on the same reasoning above, we would have $C_Y > m$, where $C_Y$ is the dimension of $C$ along $Y$ direction. Our hypothesis then would imply that the size Size($C$) of cluster $C$, satisfy the following:

$$\text{Size}(C) \geq C_X + C_Y - 1 \geq 2 \times (m+1) - 1 = 2m + 1,$$

which contradicts that $C$ is of size $2m$. The lemma is hence proved. ∎

**Theorem 3.2.** *The* 2*-D interleaving array generated by Procedure 3.2 is optimal in the sense that it can spread arbitrary burst of size* $2m$ *to distinct codeword of size* $2m^2$*.*

*Proof.* According to Lemma 3.1 and Theorem 3.1, it is easy to see that any two $2m \times 2m$ bursts within the generated $2m^2 \times 2m^2$ array are $m^2$-equivalent, and any two $m \times 2m$ or $2m \times m$ bursts within the generated $2m^2 \times 2m^2$ array are $2m^2$-equivalent. Thus, any burst with size $m \times 2m$ or $2m \times m$ can be spread into distinct blocks with size $2m^2$. However, according to Lemma 3.2, an arbitrary burst of size $2m$ is necessarily contained in a burst of size either $m \times 2m$ or $2m \times m$. Theorem 3.2 is, thus, proved. □

Theorem 3.2 indicates that if a distinct code symbol is assigned to each element in blocks of size $2m^2$ (refer to Definition 2.5) and all the code symbols associated with a block form a distinct codeword, then the SP technique can correct arbitrarily-shaped error burst of size $2m$ with one-random-error-correction code, provided the code is available. That is, the SP technique achieves the same performance as that achieved by the technique in [5].

*Comment*: Needless to say that there are certain constraints with the SP technique. Namely, it is not guaranteed that the upper bound of the minimum distance can be achieved for the larger 2-D array (multiple basis interleaving array) with size $(2m+1) \times (2m+1)$. However, since the SP based interleaving method is optimal for a large set of bursts, it provides a versatile tool for busrt error correction.

In summary, the SP approach does provide an effective way for $M$-D interleaving. For a given 2-D array of size $m_0^n \times m_1^n$, it can be applied once, and is optimal for a set of error bursts having different sizes defined in Theorem 3.1. In addition, for the case of arbitrarily-shaped error bursts having a size of $2m$, to which both the SP technique and the technique in [5] can be applied, the SP approach can also spread and correct arbitrarily-shaped error bursts with the same lower bound obtained by using the approach in [5]. For the basis interleaving array, we proposed a method which is proved to be optimal in 2-D case. For $M > D$ case, this optimality cannot guaranteed [9, 5].

# 4 Summary

In this paper, we focus on how to realize effective $M$-D interleaving. We first present a novel concept, basis interleaving array. Based on this, a new interleaving method, called successive packing (SP), is proposed to combat $M$-D error bursts. We have proved that the proposed method can spread any error burst of $m_0^k \times m_1^k$ (with $1 \leq k \leq n-1$) to different code blocks in the array of $m_0^n \times m_1^n$. Thus the simple error correction code which is optimal for independent channel can be used to correct this kinds of error bursts. It needs to be implemented only once for a given $M$-D array, and is thereafter optimal for the set of error bursts having different sizes.

# References


[1] S. B. Wicker, Error Control System for Digital Communication and Storage, *Prentice-Hall, Inc.*, Englewood Cliffs, NJ, 1995.

[2] C. de Almeida and R. Palazzo Jr., "Two-dimensional interleaving using the set partitioning technique," *Proceedings of IEEE Int. Symp. Information Theory*, p. 505, 1994.

[3] "Maxicode briefing document," *UPS Inc, http://www.maxicide.com*, 1996.

[4] K. A. Abdel-Ghaffar, "Achieving the Reiger bound for burst errors using two-dimensional interleaving schemes," *Proceedings of IEEE Int. Symp. Information Theory*, p. 425, 1997.

[5] M. Blaum, J. Bruck, and A. Vardy, "Interleaving schemes for multidimensional cluster errors," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 730–743, 1998.

[6] G. F. Elmasry, Detection and Robustness of Digital Image Watermarking Signals: A Communication Theory Approach, *Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ,* Ph.D Dissertation, 1999.

[7] Y. Q. Shi and X. M. Zhang, "Two-dimensionalinterleaving by using successive packing," *Proceedings of Thirty-Eighth Annual Allerton Conference on Communication, Control, and Computing*, pp. 945–954, 2000.

[8] Y. Q. Shi and X. M. Zhang, "A new two-dimensional interleaving technique using successive packing," *IEEE Trans. Circuit and Systems: Part I*, to appear in June 2002.

[9] S. Golomb and L. Welch, "Perfect codes in the Lee metric and the packing of polyominnoes," *SIAM J.Appl. Math.*, vol. 18, no. 2, pp. 302–317, 1970.