УДК 621.391

METHOD FOR GENERATING TWO-DIMENSIONAL DEPENDENT ERRORS

REN XUNHUAN, V.K. KONOPELKO, V.Yu. TSVIATKOU

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus

Submitted 12 March 2021

Abstract. The method of forming patterns of dependent two-dimensional errors is considered. It is shown that the algorithm works much faster than the known algorithms for generating dependent two-dimensional errors.

Keywords: dependent error, error pattern set, error pattern generation.

Introduction

Currently, error correcting code (ECC) is widely used in many fields, such as in information processing systems, in computer memory devices, in telecommunications systems, in missile control systems, in information compression systems, etc. Decoding by the syndrome is used to correct random errors of small multiplicity, the length of the verification characters will be increased with the multiplying errors. Obviously, the dependent errors can't fix decoding by the syndrome. In the theory of two-dimensional coding, the high efficiency of using the error correction pattern library is noted.

In [1, 3] proposed a method generate dependent errors patterns which is based on the process of calculating the pattern of the $t \times t$ type library and applying the identification parameters to recognise the errors. The method classified all of the formating error patterns in typical and non-typical, but with the increasing of *t*, the computational complexity increases rapidly. Based on this defect, a new method of generate the two-dimentional dependent errors patterns is proposed.

Method for generating two-dimensional dependent errors

The error, caused by the unknown interference in the channel, appeared in the receive device can be divided into random and dependent. Since the random errors are independently happened, the multiplicity of errors t over the length of the message are usually used to evaluate this type of error. Dependent errors can be further divided into batch errors and modular errors according to the degree of the generality. In fact, the modular errors are the special case of the batch errors. Batch errors may cause by the events such as the aging in communication cable， interference from periodic noise, dust particles on magnetic tape, etc. The modular errors are phased burst errors, which can be observed in memory systems built on multi-bit LSI memory. A good example is in a 16-bit memory system implemented on microcircuits with four-bit outputs, modular errors of length due to failures of individual memory LSIS [4]. The length of the modular errors is denoted by *b* in this paper.

In addition, batch errors and modular errors can be either single or multiple. Supposing there are some codes whose *n* is 16 and *b* is 4. Then supposing there are two error pattern which we detected which are $E = (0000\ 0000\ 1101\ 0000)$ and $E = (0000\ 1111\ 0000\ 1100)$. It is easy to find that the former error pattern has one error segment, in which there are four bits. Whereas, in the latter one, there are two error segments [4].

If two error patterns can become the same pattern by conducting a special rules set which given later to filter the similar error patterns, we will categorize them into one type. Based on our method, the following 8 error patterns – $E = \{(11\ 00), (00\ 11), (10\ 10), (01\ 01), (10\ 01), (01\ 10), (11\ 10), (11\ 11)\}$, which are the all-possible potential error patterns $2 \le t \le 4$, will reduced to five patterns. The procedure of the reducing presented in fig. 1.

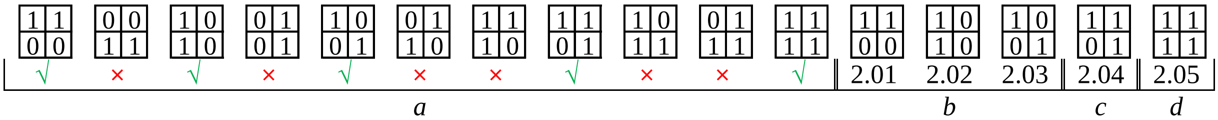

Fig. 1. procedure of the reducing size 2×2: *a* – original error patterns; *b* – dependent error patterns when $t=2$; *c* – two dependent error patterns when $t=3$; *d* – two dependent error patterns when $t=4$

Since special rules set eliminate the redundant error patterns, the leftover patterns of error can be regarded as the basis set, which will use to extend the higher order error patterns. This procedure will first extend the row and column of each element in basis set and add "1" (error) to each site where the value is "0". After that, we again applied our rule set to those extended error pattern to acquire the higher order basis set. The whole procedure can be viewed in the fig. 2.

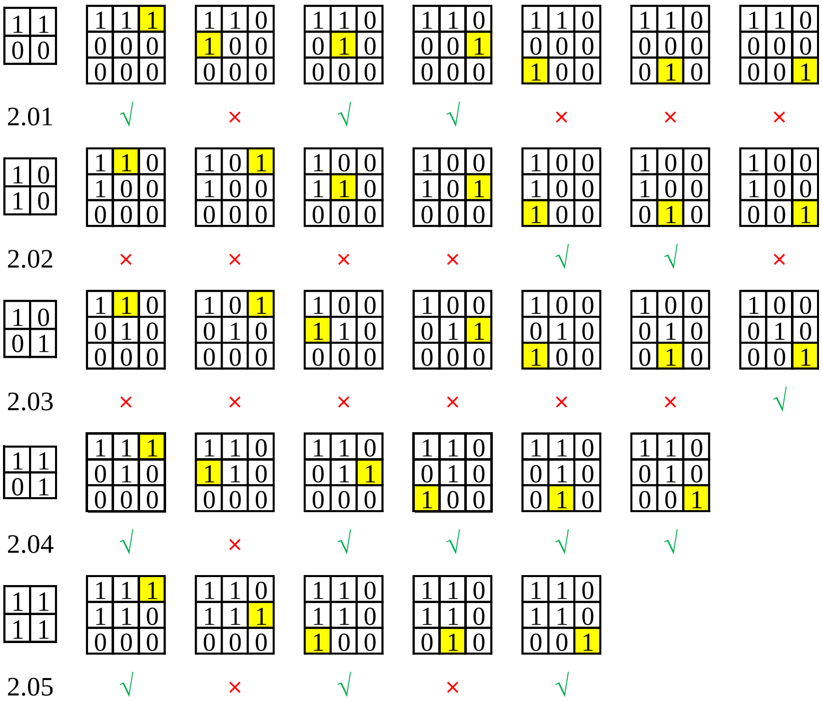

Fig. 2. Procedure of extending basic set of size 2×2 to pattern errors of all size 3×3 and the procedure of reducing error patterns of order 3 to basis set of order $t=3:5$

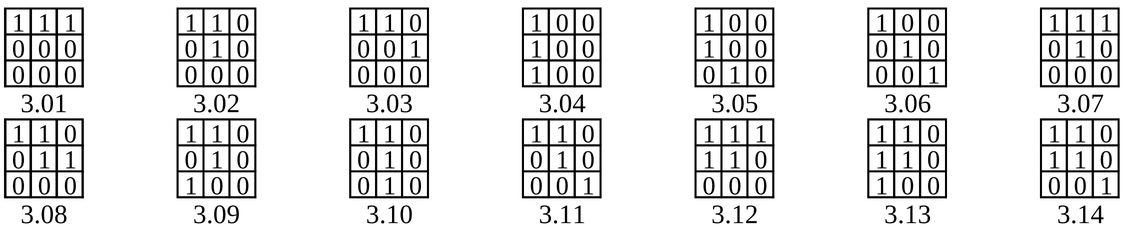

Fig. 3. From the basic set of All the basis set of order $t\times t=2\times 2$ chose the library of $t\times t=3\times 3$

From the discussion above, it is intuitive to noticed that the basis set of order 2×2 can derived 14 distinct error patterns of order 3×3, which have 5 maximum errors.

Special rules set for two-dimensional dependent errors

The rules set are worked according to the parameters of the error pattern. If two error patterns have the same parameters, one of them will be removed.

These parameters can be calculated as follows:

1. Calculate the total number of units for each row (*R*) and column (*C*).
2. Counting the number of point of intersections (*PI*) by rows and columns.

3. Calculation of the coordinate of the point of intersection (*CPI*).

4. Calculation of sum (*S*) and the difference (*D*) of the point of intersection (*CPI*).

For example, in table 1 shows the values of the parameters correspond to the above rules for fig. 3.

Table 1 shows that all of the error patterns for the size of $t \times t = 3 \times 3$ have the distinct identification parameters which can represent the other error patterns in the error space of $t \times t$. Therefore, based on these special error patterns, the higher error order of *t* can be acquired by extending the row and column.

Table 1. **Identification parameters for two-dimensional dependent error patterns when $t \times t = 3 \times 3$**

№	*R*	*C*	*PI*	*CPI*	(*S*, *D*)
3.01	(300)	(111)	0	–	–
3.02	(210)	(210)	1	(2,1)	(3,1)
3.03	(210)	(111)	0	–	–
3.04	(111)	(300)	0	–	–
3.05	(111)	(210)	0	–	–
3.06	(111)	(111)	0	–	–
3.07	(310)	(121)	1	(3,2)	(5, 1)
3.08	(220)	(121)	2	(2,2), (2,2)	(4,0), (4,0)
3.09	(211)	(220)	2	(2,2), (2,2)	(4,0), (4,0)
3.10	(211)	(130)	1	(2,3)	(5,–1)
3.11	(211)	(121)	1	(2,2)	(4,0)
3.12	(320)	(221)	4	(3,2), (3,2), (2,2), (2,2)	(5,1), (5,1), (4,0), (4,0)
3.13	(221)	(320)	4	(2,3), (2,3), (2,2), (2,2)	(5,–1), (5,–1), (4,0), (4,0)
3.14	(221)	(221)	4	(2,2), (2,2), (2,2), (2,2)	(4,0), (4,0), (4,0), (4,0)

Evaluation of the efficiency of the algorithm for generating two-dimensional dependent error patterns

Table 2. shows the values of the average time for calculating two-dimensional dependent errors when conducting the experiment, a quad-core platform was used and the program was developed in the environment of the windows 10 operating system, and Matlab was used for the program.

Table 2. **The average generating time of two-dimensional dependent error patterns for $t = 2:6$**

Methods	Average time for generating the library of dependent error patterns				
	2	3	4	5	6
Method [3]	0,008 c	0,03 c	0,1 c	4 c	800 c
Proposed	–	0,02 c	0,02 c	0,5 c	2 c

The analysis of the table data shows that the time of generating libraries of dependent errors under the proposed method is many times faster than the convoluted method.

Conclusion

We have proposed a novel fast error pattern generating algorithm based on the idea of gradually extending the lower order error pattern to higher order error pattern. The experiments have proved that the efficiency of the proposed algorithm is much higher than that of the algorithm in [3]. The reason for this result is that our method adopts the method of extending the $t \times t$ to $(t+1)\times(t+1)$ rather than the brute force search used in [3].

References

1. Конопелько В.К., Смолякова О.Г. // Докл. БГУИР. 2008. С. 19–28.
2. Конопелько В.К., Липницкий В.А., Смолякова О.Г. [и др.] // Докл. БГУИР. 2010. № 5. С. 40–47.
3. Конопелько В.К., Макейчик Е.Г., Смолякова О.Г. // Докл. БГУИР. 2009. № 5. С. 57–64.
4. Конопелько В.К., Липницкий В.А., Дворников В.Д [и др.] Теория прикладного кодирования. БГУИР. 2004.