

Error Correcting Methods for Circle-Code using the Hamming Code and the Reed-Solomon Code

Taiji Wada*, Yuki Yamane[†] and Kiyotaka Kohno[‡]

* [†]Production System Engineering Course, Advanced Engineering Course, Yonago National College of Technology, 4448 Hikona-cho, Yonago, Tottori 683-8502, Japan

[‡]Department of Electronic Control Engineering, Yonago National College of Technology, 4448 Hikona-cho, Yonago, Tottori 683-8502, Japan

*s10719wt@gmail.com, [†]shiha48@gmail.com, [‡]kohno@yonago-k.ac.jp

Abstract—We originally devised a round bar code, which is named "Circle-Code". A book-type input interface and a card-type input interface using the Circle-Code were proposed in order to improve user interface for personal computer, and some applications have been developed using these input interfaces. Moreover, a recognition program of the Circle-Code for smart phone was developed in order to apply the Circle-Code to Augmented Reality (AR). However, error correcting function has not been implemented in the Circle-Code. In this paper, we propose error correcting methods for the Circle-Code based on the Hamming code and the Reed-Solomon code. We present evaluation results which is carried out using MATLAB for these error correcting methods.

I. INTRODUCTION

We originally devised a round bar code, which is named "Circle-Code" [1]. Fig. 1 (a), (b) and (c) show the Circle-Code in case of 0x0000, 0x55AA and 0xFFFF, respectively.

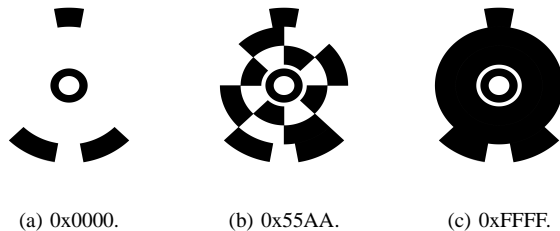


Fig. 1. The Circle-Code.

We proposed a book-type input interface and a card-type input interface using the Circle-Code in order to improve user interface for personal computer, and we have developed some applications using these input interfaces [2]. Moreover, we developed a recognition program of the Circle-Code for smart phone in order to apply the Circle-Code to Augmented Reality (AR). Fig. 2 (a), (b) and (c) show three applications using the book-type interface, Fig. 3 (a) and (b) show two applications using the card-type interface, and Fig. 4 shows a application for AR using smart phone, respectively.

However, error correcting function has not been implemented in the Circle-Code. Then we have some problems for using the Circle-Code in real environment where noises are occurred by dirt and shadow.

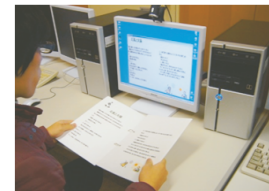


Fig. 2. Applications using the book-type interface.

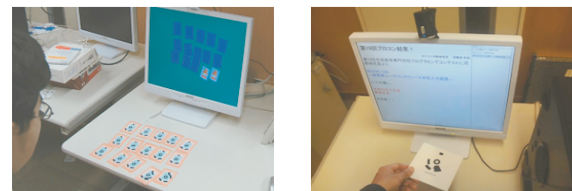


Fig. 3. Applications using the card-type interface.

In this paper, we propose error correcting methods for the Circle-Code based on the Hamming code and the Reed-Solomon code and we carry out the evaluation for these methods using MATLAB.

II. THE CIRCLE-CODE

Fig. 5 shows the structure of the 16-bit Circle-Code. The Circle-Code is composed of four concentric circles.

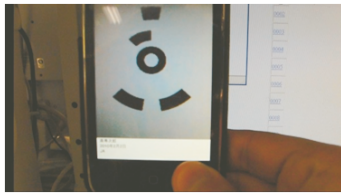


Fig. 4. Application for AR using smart phone.

An innermost concentric circle, which is named "coordinate recognition code" (Fig. 5 (b)), decides the two-dimensional coordinate and the size of the Circle-Code.

The most outside concentric circle (three circular arcs), which is named "rotation angle recognition code" (Fig. 5 (a)), recognizes the rotation angle, and decides the reading initiation position of the information bits of the Circle-Code.

Two concentric circles of the remainder, which are named "16-bit information code" (Fig. 5 (c)), are divided equally into eight parts (Fig. 5 (d)), and show the information of sixteen bits.

Because it is assumed that the Circle-Code is printed at an arbitrary size, the size of the Circle-Code is shown in the ratio and the angle (Fig. 5 (e)).

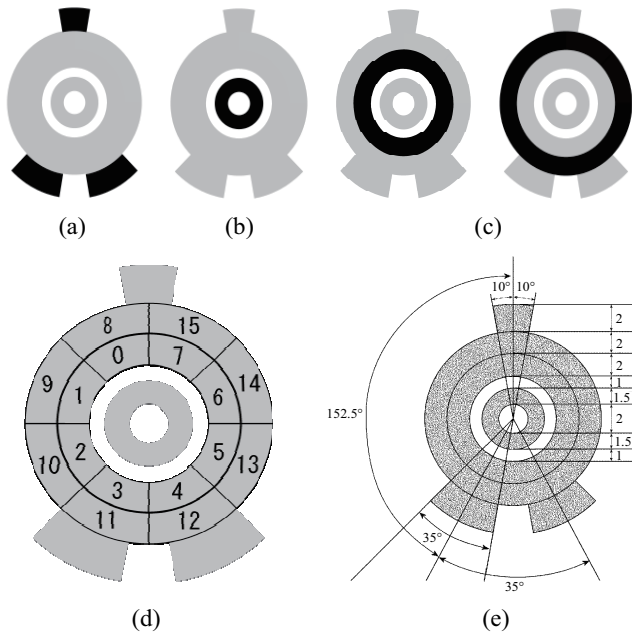


Fig. 5. Structure of the 16-bit Circle-Code: (a) the rotation angle recognition code, (b) the coordinate recognition code, (c) the 16-bit information code, (d) the information of 16 bits, and (e) the size of the Circle-Code.

Fig. 6 shows the comparison with traditional bar codes which are the JAN code and QR code. The Circle-Code has some advantages in comparison with traditional bar codes as follows:

1) The quantity of information of the Circle-Code is less than

traditional bar codes. However,

2) The type of information of the Circle-Code is more than traditional bar codes. It is possible that the two-dimensional coordinate, the rotation angle and the magnification ratio are acquired. Moreover,

3) It is easy to carry out the recognition (Fig. 7).

4) It is possible to carry out the recognition from left place (about two meters).

5) It is possible to carry out the recognition even if the Circle-Code has about ± 70 degrees of tilting angle.

6) It is possible that two or more codes are recognized at the same time.




	Quantity of information	Type of information	Easiness of recognition
	○ (91 bits)	numeral	△
	◎ (441 ~ 31329 bits)	<ul style="list-style-type: none"> Alphanumeric character Kanji character 	○
	△ (16 bits)	<ul style="list-style-type: none"> 16 bits data Two-dimensional coordinate Rotation angle Magnification ratio 	◎

Fig. 6. Comparison with traditional bar codes.

Fig. 7 shows the procedure of the recognition of the Circle-Code. The procedure of the recognition of the Circle-Code is as follows:

Step 1) An image is captured from a moving image.

Step 2) The captured image is translated to a binary format image.

Step 3) The labelling is carried out for the binary format image.

Step 4) The coordinate recognition code is recognized.

All ring figures in the image are detected, and four positions on each ring figure are specified. The four positions are shown using mark '□' in Fig. 7. Therefore, it is possible that two or more Circle-Codes are recognized at the same time.

Step 5) The rotation angle recognition code is recognized.

Three positions on the rotation angle recognition code are specified using the ratio and the angle which are showed the size of the Circle-Code (See Fig. 5 (e)). The three positions are shown using mark '△' in Fig. 7.

Step 6) The 16-bit information code is recognized.

Sixteen positions on the the 16-bit information code are specified using the ratio and the angle which are showed the size of the Circle-Code (See Fig. 5 (e)). Then, a value of each position on the 16-bit information code is judged in 0 or 1. The sixteen positions are shown using mark '○' in Fig. 7.

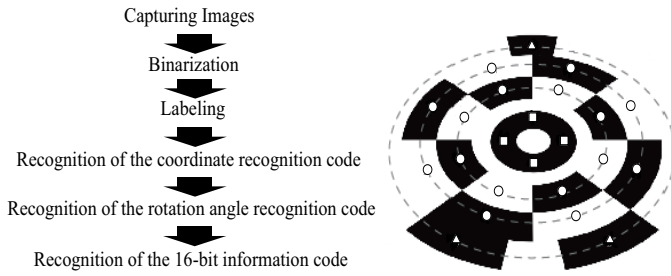


Fig. 7. Recognition procedure of the Circle-Code.

III. METHODS OF ERROR CORRECTING

A. The Hamming code

The Hamming code is a error correcting code devised by Bell Laboratory's Richard humming in 1950. It is possible to correct one bit error per block by using the Hamming code.

Let m be a number of redundancy bit, k be a number of information bit and n be a number of total bit. Then, relation of m , k and n is expressed as follows [3], [4]:

$$n = 2^T - 1, \quad (1)$$

$$k = n - m. \quad (2)$$

Fig. 8 shows the number of information bit k and the number of total bit n by changing the number of redundancy bit m from 2 to 6. It can be seen from Fig. 8 that the 16-bit information code of the Circle-Code should be extended from 16 to 32 or more than 32 if the number of information bit k is ensured over 16.

Error bit	Redundancy bit m	Infomation bit k	Total bit n
1	2	1	3
	3	4	7
	4	11	15
	5	26	31
	6	57	63

Fig. 8. The number of bit for the Hamming code.

B. The Reed-Solomon code

The Reed-Solomon code is a error correcting code devised by Irving S. Reed and Gustave Solomon in 1960. It is possible to correct consecutive error of two or more bits by using the Reed-Solomon code.

The Reed-Solomon code has the concept of the Galois field (limited field) to generate a code. An error detection and

correction of the Reed-Solomon code is carried out in each symbol. Let r be a number of bit per symbol for the Galois extension field $GF(2^r)$, t be a number of error symbol, k be a number of information symbol and n be a number of total symbol. Then, relation of r , t , k and n is expressed as follows [3], [4]:

$$t = \frac{n - k}{2}, \quad (3)$$

$$n < 2^r. \quad (4)$$

Fig. 9 shows the number of information symbol k and the number of total symbol n by changing the number of error symbol t from 1 to 4 when the number of bit per symbol is 4 ($r = 4$). It can be seen from Fig. 9 that the 16-bit information code of the Circle-Code should be extended from 16 to 32 or 48 if the number of information symbol k is ensured over 4 (16 bits).

bit per symbol r	Error symbol t	Redundancy symbol $n - k$	Information symbol k	Total symbol n
4	1	2(8)	2(8)	4(16)
			6(24)	8(32)
	2	4(16)	4(16)	8(32)
			8(32)	12(48)
	3	6(24)	2(8)	8(32)
			6(24)	12(48)
	4	8(32)	4(16)	12(48)

(): The number of bit.

Fig. 9. The number of symbol and bit for the Reed-Solomon code.

C. Extension of the 16-bit Information Code of the Circle-Code

The number of information bit of the Circle-Code can be extended by subdividing the concentric circle of the 16-bit information code and by adding some concentric circles to the information code.

Fig. 10 (a) shows the 32-bit Circle-Code which is extended by dividing equally into 16 parts of two concentric circle of the information code. Fig. 10 (b) shows the 48-bit Circle-Code which is extended by adding a concentric circle of the information code for the foregoing 32-bit Circle-Code.

IV. EVALUATION OF THE ERROR CORRECTING METHODS

The 32-bit Circle-Code shown in Fig. 10 (a) was used to implement the error correcting methods.

The recognition program of the 32-bit Circle-Code, the generation program of the 32-bit Circle-Code with the Hamming code and the Reed-Solomon code and the error correcting program were developed on MATLAB in order to evaluate the error correcting methods using MATLAB.

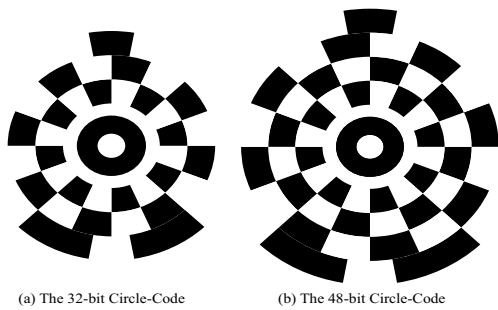


Fig. 10. Extension of the information code of the Circle-Code.

For the method based on the Hamming code, the number of information bit was chosen to be 26 ($k = 26$), the number of the redundancy bit was chosen to be 5 ($m = 5$), and one parity bit was added. Then, the number of total bit becomes 32 ($n = 32$) (See Fig. 8).

On the other hand, for the method based on the Reed-Solomon code, the number of bit per symbol was chosen to be 4 ($r = 4$), the number of error symbol was chosen to be 2 ($t = 2$), the number of information symbol was chosen to be 4 ($k = 4$, 16 bits), and the number of redundancy symbol was chosen to be 4 ($n - k = 4$, 16 bits). Then, the number of total symbol becomes 8 ($n = 8$, 32 bits) (See Fig. 9).

Fig. 11 shows the 32-bit Circle-Code with the Hamming code. The information code is 0x00055AA (26 bits). The redundancy bit and one parity bit for this information code are 111001 (5+1 bits).

Fig. 12 shows the 32-bit Circle-Code with the Reed-Solomon code. The information symbol is 0x55AA (4 symbols, 16 bits). The redundancy symbol for this information symbol is 0xECA7 (4 symbols, 16 bits).

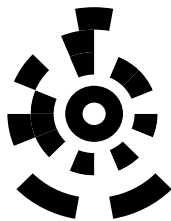


Fig. 11. The 32-bit Circle-Code with the Hamming code.

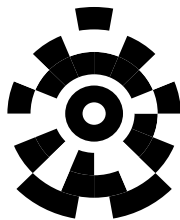


Fig. 12. The 32-bit Circle-Code with the Reed-Solomon code.

The artificial fouling was gained to the 32-bit Circle-Code

shown in Fig. 11 and Fig. 12. For example, Fig. 13 shows a sample image of a 32-bit Circle-Code with the Reed-Solomon code added some noises. Some images included noises were input to MATLAB, and the error correction was carried out to the images on MATLAB.

It can be seen through the evaluation on MATLAB that one bit error was corrected for the 32-bit Circle-Code with the Hamming code and two symbol errors were corrected for the 32-bit Circle-Code with the Reed-Solomon code.

Therefore, proposed error correcting methods are useful for real environment where noises are occurred by dirt and shadow.

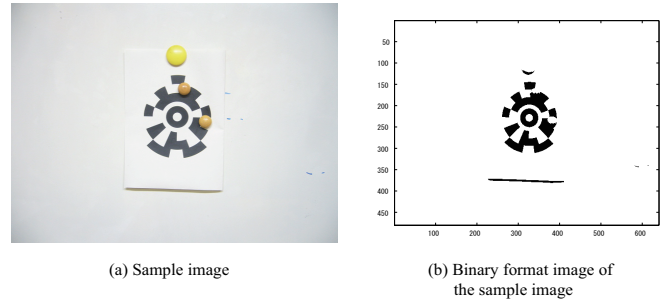


Fig. 13. Sample image of a 32-bit Circle-Code with the Reed-Solomon code added some noises.

V. CONCLUSIONS

We proposed error correcting methods for the Circle-Code based on the Hamming code and the Reed-Solomon code. We extended the number of information bit of the Circle-Code from 16 to 32 in order to implement these error correcting methods.

The recognition program of the 32-bit Circle-Code, the generation program of the 32-bit Circle-Code with the Hamming code and the Reed-Solomon code and the error correcting program were developed on MATLAB. Then, the evaluation for these error correcting methods was carried out using MATLAB. It has been shown through the evaluation using MATLAB that proposed error correcting methods is useful for real environment where noises are occurred by dirt and shadow.

The important remaining issue is the implementation the error correcting method based on the Reed-Solomon code for the 32-bit Circle-Code to a personal computer and a smart phone.

REFERENCES

- [1] D. Murata, N. Ito, R. Watanabe and K. Kohno, "Circle-Code and Its Application to Book-Type Input System," in *Proc. the 17th Annual Conference of the SICE Chugoku Chapter*, Hiroshima, Japan, Nov. 15, 2008, pp. 36-37 (in Japanese).
- [2] D. Murata and K. Kohno, "Input Interface Using the Circle-Code and Its Application," in *Proc. the 18th Annual Conference of the SICE Chugoku Chapter*, Tottori, Japan, Nov. 28-29, 2009, pp. 108-109 (in Japanese).
- [3] J. Justesen and T. Hoholdt, *A Course In Error-Correcting Codes*. Tokyo, Japan: European Mathematical Society and Morikita Publishing Co., Ltd., 2004.
- [4] Y. Eto and T. Kaneko, *Error Correcting Codes and Its Application* Tokyo, Japan: Ohmsha, Ltd., 1996 (in Japanese).