

An Amorphous 2-Dimensional Barcode

Han Jin¹, Shi Jin^{2,*} and Junfeng Wu²

Abstract: Most existing 2-dimensional barcodes are designed with a fixed shape and clear area. Having a fixed shape and clear area makes the barcode difficult to lay out with other text and pictures. To solve this problem, an amorphous 2-dimensional barcode is presented in this paper. The barcode uses encoding graph units to encode data. There are two key points in a 2-dimensional barcode: One is the encoding graph unit, the other is its encoding rules. Because encoding graph units of a 2-dimensional barcode are surrounded by other graphics, the units should be self-positioned and distinguished from other units. This paper presents an encoding graph unit generation algorithm, which is based on genetic algorithms. Encoding rules of the barcode are also given in this paper. Those rules include data encoding rules and encoding graph unit sequence arrangement rules. Data encoding rules are used to encode data to an encoding graph unit sequence. Encoding graph unit sequence arrangement rules are used to embed the unit sequence in the target picture. In addition to those rules, it also discussed the steps to restore encoding graph unit sequence from a picture. In the experiments section of this paper, an example is provided to encode a string and embed it in a car ad picture by the barcode. According to encoding rules of the barcode, those encoding graphic units can be scattered and embedded in a picture with other graphics, so amorphous 2-dimensional barcode has no fixed shape. Take advantage of this, designer can present a more elegant design to lay out barcodes with other graphic elements.

Keywords: 2-Dimensional bar code, amorphous, encoding graph unit, genetic algorithm.

1 Introduction

A 2-dimensional barcode is a machine-readable optical label that contains information about the item to which it is attached. It is an encoding scheme such as Li et al. [Li, Jiang and Cheslyar (2018); Xia, Wang, Sun et al. (2014); Pan, Zhang and Kwong (2015)], which encodes data into a graphical image according to the predefined encoding rules. 2-dimensional barcodes can be printed on the surface of items made from different materials, and the data embedded inside can be recognized and read by machines within a certain distance in non-contact mode. With the advantages of large data capacity, reliable

¹ School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China.

² School of Information Management, Nanjing University, Nanjing, 210036, China.

* Corresponding Author: Jin Shi. Email: shijin@nju.edu.cn.

and low cost, the 2-dimensional barcode plays an important role in the IOT [Shi (2018); Zhang and Zhu (2011)], cloud computing [Xie, Yuan, Zhou et al. (2018); Ren, Shen, Wang et al. (2015); Fu, Sun, Liu et al. (2015)] and mobile computing.

Because it can read data directly from pictures, 2-dimensional barcodes are used more and more widely in advertising to store information such as contacts and URLs. Instead of manual recording, people can scan barcodes in advertising pictures and get embedded information immediately and correctly. But the wide application of two-dimensional code has also brought new problems. Advertisements are traditionally designed for human beings, while 2-dimensional barcodes are read by machines such as scanners and smart phones. For human beings, those barcodes are ugly and make no sense. When embedding a barcode in advertising pictures, it is difficult to lay out graphics of an ad picture alongside those barcodes. It will undermine the overall design of advertising. When the embedded information needs to be recognized and read at a longer scan distance, the problem becomes more serious because the size of the embedded barcode must be increased. It is often seen in the outdoor advertising as shown in the following Fig. 1.

The below figure shows a billboard in South Jakarta, Indonesia. With a large size of QR code in the ad, people can scan the QR code without leaving their car. The QR code occupies half of the area of the billboard, meaning that half of the area of the ad makes no sense to people.

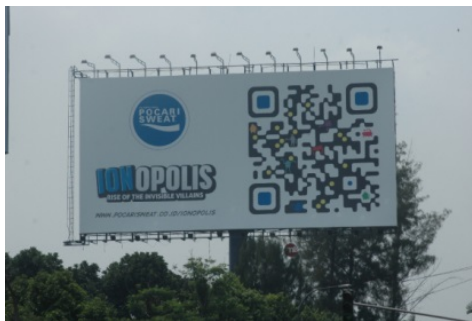


Figure 1: An ad billboard in Jakarta

There are two reasons for ad designers to lay out ad pictures like this. One reason is that almost all 2-dimensional barcodes encode data into graphs with a fixed shape, such as Maxi Code [Wang and Ye (2000)], QR Code [Lodge (2004)], Data Matrix Code [Mathus, Seguin and Torti (2002)], PDF417 [Marriott (1995)], and so on. The other reason is those 2-dimensional barcodes need to keep a certain interval (clean area) with other graphic of the picture in which it is embedded. Clear area is important to barcodes, making them more easily located and read by machines. So ad designers must keep an entire area to accommodate those barcodes.

This problem has also attracted ad designers' attention with much efforts being made to help those barcodes to be more easily accepted by people. That effort is called Barcode Design. The following pictures are some work examples of Barcode Design.



Figure 2: QR Code design examples

As shown in examples, QR barcode designers adopt different colours and distort the blocks to make barcodes more artistic. It has a certain effect, but it also brings other problems. According to the results of scan testing on those barcodes, blocks distorting causes a certain impact on the correct rate of data reading.

In order to solve the problem, we present an amorphous 2-dimensional barcode in this paper. Unlike other 2-dimensional barcodes, the barcode encodes data by independent encoding graph units which consist of small black blocks and white blocks. According to the amorphous 2-dimensional barcode encoding rules, each of the encoding graph units are designed with self-position and differ from each other. Without a clean area, those units can also be recognized and read by machines. Because it can be split into encoding graph units, amorphous 2-dimensional barcode has no fixed shape. And its encoding graph units can mix up with other text and images without interfering with the machine to read data from it. Taking advantage of this, ad designers can produce a more elegant design to lay out barcodes with other ad graphic elements such as text and pictures.

There are two key points of amorphous 2-dimensional barcode. One is the encoding graph units. Because those encoding graph units will be mixed up with other graphics without a clear area, it needs to be carefully designed to ensure the correct read rate of amorphous 2-dimensional barcode. In the second section of this paper, it presents an algorithm to generate those units based on genetic algorithm. The other is the encoding rules of the amorphous 2-dimensional barcode. Those rules can be divided into two rule groups. One is data map rules which map data into an encoding graph unit list. And the other is embed rules which embed encode graph unit list into target pictures. We will represent the rules in the third section. The remaining parts of the paper cover experiments and discussions.

2 The encoding graph unit generation algorithm

The encode graph units of an amorphous 2-dimensional barcode will be scattered and placed anywhere in the picture in which it is embedded. So the barcode has no fixed shape. In order to be recognized and allow other machines to read data from it, those encoding graph units need to be carefully designed. In general, the units should meet the following requirements:

1. Encoding graph unit should be distinguished easily from the other texts and pictures around it. So it can be recognized and read reliably by machines.

2. Each encoding graph unit should be as different as possible from other units. That is to say, those units should be designed with maximum code distance between each other.
3. Encoding graph unit can be positioned by itself without any other position marks or position patterns. That means it is self-positioned. This is an important feature which allows units to be scanned and read from any direction by barcode scanner.

In order to meet the first requirement, it is adopted that an encoding graph unit is composed of black square blocks and white square blocks. Because square blocks have right angles, it can be relatively easy to be identified by image processing algorithms. It is defined in the amorphous 2-dimensional barcode that a white block represents bit (0) and a black block represents bit (1).

Supposing that each encoding graph unit has N blocks, there are 2^n encoding graph units in the complete encode table of amorphous 2-dimensional barcode. And the minimum code distance of this encode table is one in the complete encode table. This means there are many encoding graph units which are different from other units with only one block, which doesn't satisfy the second requirement mentioned above. When one block of those encode graphic units has damaged, the unit would hardly be able to distinguish the two encoding graph units from each other. In order to enhance the interference resistance capacity of our barcode, we introduce redundant encoding. Encoding graph units are selected from the encoding table to form an encoding graph unit set. The minimum encoding distance of the unit set is the minimum encoding distance between two units of the set. According to the second requirement above, the unit set should be selected with the longest minimum encoding distance among optional unit sets from the complete encode table.

From a practical point of view, 2-dimensional barcode images are susceptible interference by image noise, scratches, stains and so on. Those interferences obstruct machines from reading data from damaged barcode correctly. Because the size of encoding graph units are much smaller than the size of the picture in which it is embedded, the interferences mentioned above can be simplified as shown in the following figure (in which the encoding graph unit consists of 3×3 blocks, for example).

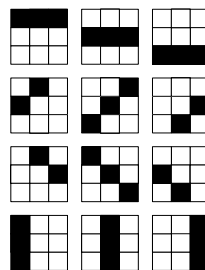


Figure 3: Simplified interferences

According to the error correction coding principle [Barron, Rissanen and Yu (1998)], supposing the encoding graph unit consist of $N \times N$ blocks and it can correct up to n -bits of errors, then the minimum code distance between units is at least $2 \times N + 1$. So in the last example, the minimum encoding distance of the encoding graph unit set should be

$3 \times 2 + 1 = 7$ to ensure the damaged encoding graph unit can be identified by machines when one of the interferences occurs.

According to the third requirement mentioned above, units of the encoding graph unit set are self-positioned. So the third requirements can be converted into the following conditions that the encoding graph units set of amorphous 2-dimensional barcode should be met to:

1. The minimum encoding distance of the encoding graph set is at least $2 \times N + 1$, when units of the set consist of $N \times N$ blocks.
2. Each unit of the set is distinguishable from other units even if it has rotated by 90, 180, 270, 360 degrees.
3. The encoding graph unit set should have the longest minimum encoding distance among the sets which meet condition 1.

Supposing an encoding graph unit consists of $N \times N$ blocks, there is $2^{(N \times N)}$ optional encoding graph units, and the max length of encoding graph units set is $2^{(N \times N - 2 \times N - 1)}$. If N is 4, we need select $2^{(4 \times 4 - 2 \times 4 - 1)} = 2^7 = 128$ units from $2^{16} = 65536$ optional units. This is a complicated task.

There are some methods for solving complex problems such as the Genetic Algorithm, SVM [Bin, Sheng, Wang et al. (2015)], AI, and so on. If it takes the binary sequences which are represented by each encoding graph units as a gene, the Genetic Algorithm can be used to generate the preferred encoding graph unit set as mentioned above. In this paper, an encoding graph unit generation algorithm based on the Genetic Algorithm [Harik, Lobo and Goldberg (1999)] is presented.

First of all, it needs to define the fitness function for the Genetic Algorithm to generate the encode graphic unit set. In our algorithm, it defines a code distance measure function first. Then the fitness function of our GA algorithm is defined based on the function. The code distance function which measures the distance of two encoding graph units is defined as follows:

$$\text{dis}(U_1, U_2) = \begin{cases} 0 & \text{row}(U_1, i) = \text{row}(U_2, i) \\ 0 & \text{col}(U_1, i) = \text{col}(U_2, i) \\ 0 & \text{dig}(U_1, i) = \text{dig}(U_2, i) \\ 0 & \text{rotate}(U_1) = U_2 \\ \text{diff}(U_1, U_2) & U_1(i, j) \neq U_2(i, j) \end{cases} \quad (1)$$

In formula (1), $\text{row}(U_1, i)$, $\text{col}(U_1, j)$ and $\text{dig}(U_1, k)$ get the remaining part of an encoding graph unit, which has been removed blocks of the i th row, blocks of j th column and blocks of k th diagonal of the unit respectively. If the remaining parts of U_1 , U_2 are same, the distance between U_1 and U_2 is zero. These definitions are designed to improve the capacity of resisting disturbance of encode graph units generated by the algorithm. The $\text{rotate}(U_1)$ gets unit U_1 rotated by 90, 180, 270, 360 degrees, if the rotated U_1 the are same as U_2 , the distance between U_1 and U_2 is also zero. The definition allows the generated encode graph unit to be distinguished with other units by barcode scan machine from any direction. The function $\text{diff}(c_1, c_2)$ gets the number of blocks which are different blocks with same location in U_1 and U_2 , as general code distance definition.

According to the definition of the distance function, fitness function of a given encoding graph unit Set S is defined as the following formula (2):

$$P(S) = \sum_{i,j=1}^n \text{dis}(U_i, U_j) \quad (2)$$

Then encoding graph unit generation algorithm based on genetic algorithm for the amorphous 2-dimensional barcode is shown as the following:

Encoding Graph Units Generation Algorithm

Count<-0; Sbest, S1, S2<-{};

While (count< MaxRoundTime){

if (Count=0) s1, s2<- generate two encoding graph unit set randomly;

S'<-encoding units generated by gene mutation of units in s1, s2;

S''<-encoding units generated by gene exchange of units in s1, s2;

S<-puts all units of s1, s2, s', s'' together in a set;

S'best<- select the best encode graph units set of S;

Compare P(Sbest) and P(S'best)

if S'best has higher fitness value than Sbest, Sbest<-S'best;

S1<-S'best, S2<-generate an encoding graph unit set randomly;

Count++;

}

end.

Algorithm 1: Encoding graph unit generation algorithm

At the beginning of the algorithm, two encoding graph unit sets S1 and S2 are generated randomly. Each set contains the same number of encoding graph units according to the first condition mentioned above. Then two new sets S' and S'' are produced by gene mutation and gene exchange of units of S1 and S2. Each unit of S1 and S2 can occur mutate and exchange gene segment with other units by predefined gene mutate and gene exchange percentages.

In the next step, it puts all units of four sets up together in unit set S. And the algorithm gets all optional encode graph unit set from set S. Then the fitness value of each optional unit set is calculated by the algorithm according to the fitness function. The unit set which has the maximum fitness value among those optional unit sets is called the best encoding graph unit set and is denoted as S'best. If S'best has a higher fitness value than the current best encoding graph unit set Sbest, Sbest will be replaced by S'best. Then S1 is replaced by S'best. A new set is produced by units which are randomly selected from the mixed encoding graph unit group S, and then replaces S2.

The algorithm repeats the above steps round by round until the predefined round time is reached. When the last round is completed, it outputs the best encoding graph unit set Sbest as the result set.

3 The encoding rules of the amorphous 2-dimensional barcode

The encoding rules of the amorphous 2-dimensional barcode are mainly divided into two groups. The first rule group is data map rules, it is used to transform data into a sequence of encode graph units. Considering that barcodes are easily damaged by interference, the redundancy check codes are added to the binary byte sequence of the originating data by those rules. The second rule group is arrangement rules for encoding the graph unit sequence. Those rules are used to arrange encoding graph units in positions in the target picture according to the ad design. Because the units are scattered and placed in any possible position of the picture, the embedded unit sequence should be able to read from any direction for the same reason as encoding graph unit. Further, the order of each unit in the unit sequence should be restored correctly. In order to meet those requirements, it arranges the unit sequence in a spiral shape by our barcode embed rules, as the following Fig. 3 shows.

As can be seen from Fig. 3, all of the embedded encoding graph units can be divided into three types. The encoding graph units whose name starts with char ‘C’ are the units which carry the origin mapped data. The encoding graph unit whose name is ‘ ϕ ’ is the start and end encoding graph unit of the embedded unit sequence.

In order to read data from an amorphous 2-dimensional barcode, it is important to restore the encoding graph units sequence correctly from encoding graph units which are detected by barcode scan machine from an ad picture. The Steps to get the units sequence are listed as follows:

1. First of all, get all detected units from the target picture, the set of detected units noted by S;
2. For each unit of S, judge whether or not the unit is at the outermost edge of a spiral shape. If it is, put it in set S1;
3. Remove the S’ from S. Return to Step one and repeat the above steps until S is empty.

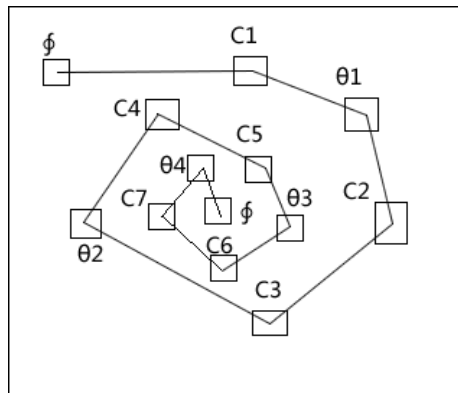


Figure 4: The embedded encoding graph unit sequence spiral arranged in spiral shape

In Step 2 of the list mentioned above, we need to judge whether or not an encoding graph unit is at the outermost edge of a spiral shape. It adopts the following steps to judge it:

1. First connects all detected units by lines to form a complete graph;

2. For each unit, take it as a starting point, draw lines from it with any direction from 0 to 360;
3. If any of the lines is always crossing an edge of the complete graph, the unit is inside of the complete graph, so it isn't at the outermost edge of the spiral shape, else it is.

In Step 3, we judge the place of a unit in a spiral shape based on the construct of the complete graph which consists of detected graph units. According to the spiral shape structure, if a unit isn't on the outermost edge of the spiral, the unit is inside of the complete graph. So the half-line which is drawn from the unit with any slope will always be crossed with the edges of the complete graph.

Obviously, it will get a set sequence (S1, S2, ..., Sn) after the steps above are finished. The start point unit ϕ of the whole encoding graph units sequence must be one of the units which are in S1, and the end point unit ϕ must be one of the units which are in Sn. The units of Si+1 are following the units of Si in the origin encoding graph unit sequence. Though by the above steps the algorithm is able to get the encoding graph unit slices with correct order of an encoding graph unit sequence, it is still required to restore orders of the units in each slice. We use redundancy check code to denote the order of each graph unit in a slice Si. As is shown in Fig. 3, there are some encoding graph units whose name start with char ' θ '. Those units not only carry redundancy check codes, but also connect adjacent graph units in an encoding graph unit sequence by redundancy checking codes. The redundancy check codes of the amorphous 2-dimensional barcode are generated by formula 3.

$$\theta_i = C(i-1) \times 2 \oplus [L(C2 \times i - 1)], H(C2i)], C0 = \phi \tag{3}$$

In formula 3, $[L(C2 \times i - 1)], H(C2i)]$ is a function to get a new byte by combining the low half of $(C2 \times i - 1)$ and the high half of $C2i$ together. Then θ_i is the XOR operation result of the new byte and $C(i-1) \times 2$.

Denote the length of origin data encoding graph unit sequence C^* is L, according to the formula 3, when $L \% 2 = 0$, there isn't redundancy check code for the low half bits of the last unit CL, and when $L \% 2 = 1$, it lacks a half bits of encoding graph unit to generate check code. In order to solve the problem, we append one unit ϕ after CL when $L \% 2 = 0$, and appended two ϕ units after CL when $L \% 2 = 1$, as the following Fig. 4 shows. Then the last redundancy check code of an encoding graph unit sequence is:

$$\theta = CL \oplus [L(CL-1)], H(\phi)] \quad \text{if } L \% 2 = 1 \tag{4}$$

$$\theta = \phi \oplus [L(CL)], H(\phi)] \quad \text{if } L \% 2 = 0 \tag{5}$$

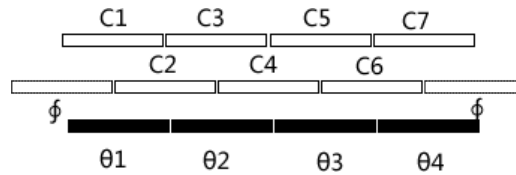


Figure 5: The fragment of encode graph unit sequence

The steps for transforming origin data to encoding graph units sequence are listed as follows:

1. Transform origin data into binary byte sequence;

2. According to the length of encoding graph unit set, split the binary byte sequence into binary byte fragments (B1, B2, ..., Bn). Supposing the number of units of encoding graph unit set is 2^N (this does not include the unit ϕ), and the length of binary byte sequence of origin data is L, then n is L/N ;
3. Get encoding graph unit sequence (C1, C2, ..., Cn) by mapping each Bi of the binary byte fragment sequence to encoding graph unit Ci according to the encoding graph unit set;
4. Insert a unit ϕ at the begin of the unit sequence (C1, C2, ..., Cn), and if n is odd, append one ϕ unit to the end of the sequence, if n is even, append two ϕ units to the end of the sequence;
5. Insert θ i units after C_{2i+1} unit of the sequence.

There is a chance that L may not be divisible by n in Step 2. The solution for this is to append the binary byte sequence of origin data by magic chars. For example, if the origin data was a string, the magic char can be '\0'.

4 Experiments

This paper realizes the generation algorithm of encoding graph unit set of the amorphous 2-dimensional barcode. In the experiment, it defines an encoding graph unit which consists of 4×4 square blocks. According to the algorithm, the length of bit fragment to be encoded by each encoding graph unit is $4 \times 4 - 2 \times 4 - 1 = 7$. In order to improve the anti-interference capability of encoding graph units, it is set to 6 in the experiment, so we can get an encoding graph unit set with longer code distance. In addition to this, considering the encode rules of the amorphous two-dimensional barcode, it needs an encode graph unit ϕ to denote the start unit and end unit of encode graph unit sequence. Thus in our experiment, it has selected 65 units from 65536 optional units by the algorithm.

In our experiments, the total round time of the algorithm is defined to 2000, and gene mutation percentage and gene exchange percentage are all set to 0.3. The result encoding graph unit set produced by the algorithm is shown as the following in Fig. 5. The 64th encode graph unit is the unit ϕ . The bits represented by the unit are '1011010011011001'.

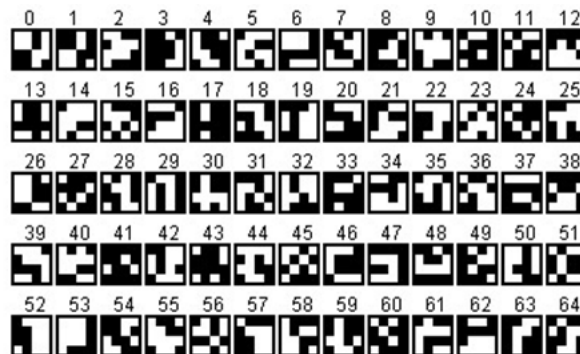


Figure 6: The result encoding graph unit set

In our experiments, we select a car ad as a target picture, and choose to embed a string ‘Hello World!’ in the data embedding experiments. According to encode rules mentioned above, after appended units ϕ and redundancy check code, the final bit fragments that are to be embedded are shown as the following bit arrays:

```
[101101, 001101, 100101,
001000, 011001, 010110,
110001, 101100, 011011,
110111, 011101, 010111,
011011, 110111, 001001,
101100, 011001, 000010 ]
```

Then the encode graph unit sequence is obtained by mapping those bit fragments according to the encoding graph unit set shown in Fig. 6.



Figure 7: Embeds string “hello world!” in a car ad picture

As it can be seen in Fig. 6, according to the encode rules, the final encoding graph unit sequence is arranged in spiral shape and forms a character ‘W’. In the ad picture, the sequence replaces the character ‘W’ of word ‘We’. Although it looks not good for the point of professional design view, it is more easily accept by human being. Obviously our barcode can give more space for ad designers to utilize their expertise.

We also carried out the anti-interference ability experiment of the amorphous two-dimensional barcode. 100 pieces of 64bit binary data were randomly generated in the experiment. The binary data were encoded with QR code and the amorphous two-dimensional barcode in this paper, respectively, and converted into three sets of sample pictures. These three sets of sample images superimpose the same crease, noise and defect image interference respectively. The recognition rate of QR code and the amorphous two-dimensional barcode in this paper is compared by using their respective recognition algorithms. In this paper, the recognition rate of amorphous two-dimensional code reached 95.4% when the crease coverage was 33.2%, noise coverage was 31%, and defect coverage was 25.7%. The QR code had a recognition rate of 83.2% under the same interference. Therefore, the code in this paper is better than QR code in anti-interference ability.

5 Conclusions

In this paper, we present an amorphous two-dimensional barcode as a solution to arrange barcodes and other graphics together harmonically. Unlike other barcodes, this type of barcode encodes data by different encoding graph units which consist of square black blocks and white blocks. The units of an amorphous two-dimensional barcode can be

scattered and placed anywhere within a picture. Therefore, an amorphous two-dimensional barcode has no fixed shape. In this paper, it also presents an encoding graph unit generation algorithm based on the Genetic Algorithm. In the algorithm, it defines a code distance measure function first, and by the function, the graph encode units generated by the algorithm keep a certain distance from each other. This increases the anti-interference ability of our barcode.

Long code distance ensures those units can be recognized by a machine reliably when those units are embedded in a picture and surrounded with other graphics or even a part of blocks of those units has been damaged by interference.

Because the amorphous two-dimensional barcode encodes data in a graph encode unit sequence, it is important to restore the encoding graph unit sequence correctly from a scanned picture. According to the amorphous two-dimensional barcode encoding, the units of the sequence are embedded separately in ads. To restore the embedded sequence, the first step is to get the correct order of each graph encode unit in the encode sequence. In our barcode, it arranges the encoding unit sequence in a spiral shape. Taking advantage of the feature of spiral shape, the embedded amorphous two-dimensional barcode not only can be scanned directly, but also can restore the correct order of unit sequence slices in the encode graph unit sequence, as the resolution is given in Section 3 of this paper.

To further improve the reliability of our barcode, it also defines unit ϕ to mark the start point and end point of unit sequence. Obviously, if there was an encode graph unit which has been damaged by interference, it will destroy the spiral shape of the whole unit sequence and prevent the unit sequence from being restored. In order to solve the problem, it introduces redundancy checking units in the unit sequence. Redundancy checking units are inserted in the position among three adjacent units in the graph encode unit sequence. Those units carry the result of a XOR operation of three adjacent units in the encode graph unit sequence. Although redundancy checking units reduce about a quarter of the capacity of our barcode, they increase the anti-interference ability of our barcode significantly by redundancy checking code. In our barcode, redundancy checking units are not only used to carry redundancy checking code, but also to denote the orders of three adjacent units in the unit sequence. So redundancy checking units are important to our barcode for they can be used to restore the order of units of an encode graph unit sequence correctly.

In summary, redundancy is introduced twice in an amorphous two-dimensional barcode. One is introduced in the encoding graph units generator algorithm and the other is in the encoding rules. It allows our barcode to be read reliably by scanning machines.

Because almost all of existing 2D barcodes require an entire and fixed shape to encode data, there isn't a similar type of 2D barcode to make a comparison with our barcode. But as it is shown by the experiments of this paper, the encode graph unit sequence of our barcode can be arranged as a character or a symbol. This means our barcode can not only carry encoded data for machines, but also it can bring more information to people and make the barcodes more easily accepted by human beings. Lastly, our barcode provides more space for ad designers to use their ingenuity.

Acknowledgement: This work was funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

- Barron, A.; Rissanen, J.; Yu, B.** (1998): The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743-2760.
- Bin, G.; Sheng, V. S.; Wang, Z. J.; Ho, D.; Osman, S. et al.** (2015): Incremental learning for v-support vector regression. *Neural Networks*, vol. 67, pp. 140-150.
- Fu, Z. J.; Sun, X. M.; Liu, Q.; Zhou, L.; Shu, J. G.** (2015): Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. *IEICE Transactions on Communications*, vol. E98-B, no. 1, pp. 190-200.
- Harik, G. R.; Lobo, F. G.; Goldberg, D. E.** (1999): The compact genetic algorithm. *Evolutionary Computation, IEEE Transactions on Information Theory*, vol. 3, no. 4, pp. 287-297.
- Li, C.; Jiang, Y.; Cheslyar, M.** (2018): Embedding image through generated intermediate medium using deep convolutional generative adversarial network. *Computers, Materials & Continua*, vol. 56, no. 2, pp. 313-324.
- Lodge, D.** (2004): Glass vials with data matrix codes and method of making the same. *U.S. Patent Application*, no. 6.
- Marriott, M.** (1995): PDF417 portable data files a new dimension in barcodes. *Sensor Review*, vol. 15, no. 1, pp. 33-35.
- Mathus, G.; Seguin, D. J.; Torti, V. A.** (2002): Test tube with data matrix code markings. *U.S. Patent*, no. 6.
- Pan, Z. Q.; Zhang, Y.; Kwong, S.** (2015): Efficient motion and disparity estimation optimization for low complexity multiview video coding. *IEEE Transactions on Broadcasting*, vol. 61, no. 2, pp. 166-176.
- Ren, Y. J.; Shen, J.; Wang, J.; Han, J.; Lee, S. Y.** (2015): Mutual verifiable provable data auditing in public cloud storage. *Journal of Internet Technology*, vol. 16, no. 2, pp. 317-323.
- Shi, C.** (2018): A novel ensemble learning algorithm based on DS evidence theory for IoT security. *Computers, Materials & Continua*, vol. 57, no. 3, pp. 635-652.
- Wang, Y. P.; Ye, A. Q.** (2000): Maxicode data extraction using spatial domain features. *U.S. Patent*, no. 6.
- Xia, Z. H.; Wang, X. H.; Sun, X. M.; Wang, B. W.** (2014): Steganalysis of least significant bit matching using multi-order differences. *Security and Communication Networks*, vol. 7, no. 8, pp. 1283-1291.
- Xie, X.; Yuan, T.; Zhou, X.; Cheng, X.** (2018): Research on trust model in container-based cloud service. *Computers, Materials & Continua*, vol. 56, no. 2, pp. 273-283.
- Zhang, H.; Zhu, L.** (2011): Internet of things: key technology, architecture and challenging problems. *IEEE International Conference on Computer Science & Automation Engineering. IEEE*, vol. 4, pp. 507-512.