RESEARCH ON OPTIMIZATION OF ENCODING ALGORITHM OF PDF417 BARCODES

Ming Sun[1,*] ,Longsheng Fu[1] ,Shuqing Han[1]
[1] *College of Information and Electrical Engineering, China Agricultural University, Beijing, China, 100083*
[*] *Corresponding author, Address: P. O. Box 63, 17 Tsinghua East Road, Haidian District, Beijing, 100083, P. R. China, Tel:+86-10-62737591, Email: drmingsun@163.com*

Abstract: The purpose of this research is to develop software to optimize the data compression of a PDF417 barcode using VC++6.0. According to the different compression mode and the particularities of Chinese, the relevant approaches which optimize the encoding algorithm of data compression such as spillage and the Chinese characters encoding are proposed, a simple approach to compute complex polynomial is introduced. After the whole data compression is finished, the number of the codeword is reduced and then the encoding algorithm is optimized. The developed encoding system of PDF 417 barcodes will be applied in the logistics management of fruits, therefore also will promote the fast development of the two-dimensional bar codes.

Keywords: Two-dimensional barcode, PDF417, encoding algorithm, visual C++6.0

1. INTRODUCTION

In recent years, the consumption conceptions of the agricultural products such as fruits have changed significantly. People’s focus is transferring from the amount to the quality. However, it is difficult to distinguish a high-quality product from a counterfeit or an inferior product from its appearance alone. The immediate concerns are to ensure the quality of fruit products from the market to the consumers and to guarantee consumer benefits; simultaneously, to protect manufactures' economic benefits and to encourage manufacturers to plant healthy fruit products without harmful effects. Currently, however, inferior fruits pass for genuine ones on the market;

moreover, the difficulty of identifying the manufacturer responsible is compounded by the lack of product marking.

The two-dimensional (2D) barcode, considered a new kind of practical technique in the field of automatic recognition and information carrier, is understood by more and more people, and can be used for marking fruit products. The primary advantage of 2D code is the ability to encode a large quantity of information in a small space. Since PDF417 code is a type of wide application 2D code, we store the basic information of fruits and their manufacturer in a PDF417 code, which is glued or printed on the fruit box. The consumer can request salespeople to identify PDF417 code with the aid of a decoding system and print out the basic information of fruits, their manufacturer, and also the web address to an online database containing further information such as: the time of harvest, fertilizer information, national national attestation with a prize certificate, etc. Alternatively, the consumer can also make use of a digital camera or a web camera to take a PDF417 code picture and read its information using a decoding software. Since 2D barcodes permit faster and more accurate recording of information, work in process can move quickly and be tracked precisely. Quite a bit of time can be spent tracking down the location or the status of projects, folders, instruments, materials, or anything else that moves within an organization. A 2D barcode can help you keep better track of them so you can save time and respond more quickly to inquiries and changes. Thus, 2D barcodes can improve operational efficiency and management level of manufacturers, enhance market order of agricultural products and foods including the fruits, and realize the modern management of agricultural products and foods in China.

2. BRIEF INTRODUCTION OF PDF417

PDF417 is a 2D barcode which can store up to about 1,800 printable ASCII characters or 1,100 binary characters per symbol, as shown in Figure 1. The symbol is rectangular; the shape of the symbol can be adjusted to some extent by setting the width and allowing the height to grow with the data. It is also possible to break large amounts of data into several PDF417 symbols which are logically linked. There is no theoretical limit on the amount of data that can be stored in a group of PDF417 symbols.

Each PDF417 2D symbol is composed of 4 bars and 4 spaces, as shown in Figure 2 (GB/T 17172-1997, 1997). Each bar or space includes 1 to 6 mold piece and for a total mold number of 17, their structure accounts for their name “417 barcode”.

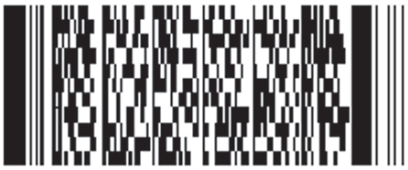
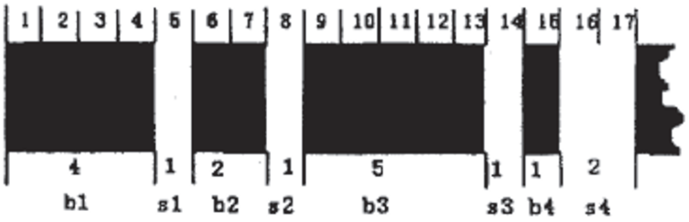


Fig.1: Structure of PDF417



. *Fig.2:* Module distribution of PDF417

3. ENCODING OF PDF417

PDF417 2D barcodes are first put forward by the USA, so application outside China has already been very widespread. A barcode Symbology defines the technical details of the barcode and the ISO standard defines many concrete encoding rules, but the particularities of Chinese characters have not been taken into account in the encoding of PDF417 barcode (Qi, 2003). Though China has already established a national standard of PDF417, the encoding rules suitable for Chinese characters still have not been put forward. This paper focuses the encoding of Chinese characters, and effectively realizes the encoding of PDF417 barcode of Chinese characters in the computer.

In fact, the computer encoding process corresponds to conversion of input information into codewords. The concrete process is as follows:

Firstly, encode the input information with compression according to the certain mode structure, and obtain data codewords. Secondly, calculate error-correction codewords according to the data codewords. Then, according to the number of the data codewords and the error-correction codewords, make the arrangement of codewords matrix, and obtain the finder of both sides of each row. Finally, search the code character list number, i.e. character set, obtain the sequence consisting of bars and spaces corresponding to the codewords, and then draw a PDF417 image.

3.1 Data encoding

The PDF417 has three kinds of data compression mode structure: text compression mode, byte compression mode, numeral compression mode. These modes can be conversed each other (Chen and Liu, 2006).

The text compression mode includes four sub-modes: Alpha, Lower Case, Mixed, and Punctuation.

Sub-modes can make representation of data more effective. The character set, which appears more frequently, is chosen in every sub-mode. Every codeword can be represented with a pair of characters and is calculated by the following formula:

$$\text{Codewords} = 30 * H + L \qquad (1)$$

Where: H and L are high-bit and low-bit character value in the pair of characters, respectively.

The numeral compression mode converts the radix of number information according to the numeral total. By the conversion of the radix from 10 to 900, the numeral sequence is converted to a codeword sequence. We can represent three numeral bits with one codeword in numeral compression mode. The numeral sequence is divided into a group every 44 bits. A pre-bit '1' is added in the front of every numeral sequence group. Then by the conversion of the radix from 10 to 900, 15 codewords are obtained.

The byte compression mode converts the radix of the number information according to byte total. By the conversion of the radix from 256 to 900, the byte sequence is converted to codeword sequence. The byte compression mode contains two locked modes (901/924). When byte total is multiples of six, the 924 mode is taken. By the conversion of radix from 256 to 900, every 6 bytes from left to right in the sequence can be represented by 5 codewords. When byte total is not times of 6, the 901 mode is taken. The conversion of first 6 bytes is the same as 901 mode. Every one in the left bytes corresponds to a codeword, and can be represented directly by codeword. 913 mode is used for temporary conversion from text compression mode (TC) to byte compression mode (BC), which is represented directly by its codeword. The conversion only works on the first codeword, the succeeding codewords return to the current sub-mode of text compression mode (TC).

When encoding is done in this mode, we need to judge whether the number of data streams is times of 6 at first, and choose the right mode to perform the radix conversion. If it is times of 6, 924 mode is chosen; if not, 901 mode is chosen.

3.2 Optimization of byte compression encoding algorithm

Inasmuch as this system aims at the application of marking fruits, its input information contains fruit name, habitat, internal quality, appearance and other details such as: the address, telephones, and faxes of manufacturer and clients, as well as the production date. The system is mainly used in China, so only Chinese character encoding is employed in the system. A Chinese character is stored as machine code in two bytes. Machine code is used to compression encoding for Chinese characters. Byte compression mode is adopted for other numbers and alpha. Since English characters and numbers are represented in one byte in the computer, to conform to Chinese characters, they are converted to representation in two bytes. Then we can implement byte compression encoding for all the input information. The concrete process is as follows.

(1) Judgment of the input information.
(2) Conversion from the acquired GB Code to QW code.
(3) Compression encoding of radix conversion for QW code.

3.3 Error-correction codewords

Since PDF417 is oriented to users, we can choose an error-correction rank for it as required and obtain the related error-correction codeword Ci. When error-correction rank S is chosen, the number of error-correction codewords is k = 2^(S+1). The error-correction rank can be determined automatically by the system as shown in Table 1, or designated by users. The rules, by which the system determines an error-correction rank, are showed in table 1. However, users designate an error-correction rank by themselves, which cannot be lower than the recommended rank. Otherwise, there will appear an error prompt. For a given codeword set, an error-correction codeword is calculated by the Reed–Solomon error-correction algorithm. However, the VC++ program can not run in the equation with the unknown. Therefore, Ci has to be calculated indirectly as follows:

Table 1. Recommended error-correction rank

Number of codewords	Error-correction ranks
1 ~40	2
41 ~160	3
160~320	4
321~ 863	5

(1) Establishment of a polynomial.

Polynomial is established as follows:

$$\begin{aligned} d(x) &= d(n-1)\,x\hat{}\,(n-1) + \\ &\quad d(n-2)\,x\hat{}\,(n-2) + \ldots + \\ &\quad d(1)\,x + d(0) \end{aligned} \tag{2}$$

Where: coefficients in the polynomial are composed of data codewords. The first codeword is the coefficient of the highest-order term. The last one corresponds to the lowest-order term.

(2) Establishment of generated polynomials of error-correction codewords.

K generated polynomials of error-correction codewords are shown as follows:

$$\begin{aligned} g(x) &= (x - 3)(x\ \ 3\hat{}2)\ldots(x\ \ 3\hat{}\,k) \\ &= x\hat{}\,k + g(k-1)\,x\hat{}\,(k-1) + \ldots + g(1)\,x + g(0) \end{aligned} \tag{3}$$

(3) Calculation of error-correction codewords.

To a set of given codewords and a chosen error-correction rank, an error-correction codeword is the complement of the coefficient of the remainder obtained by polynomial d (x) multiplied by x^k and divided by generated polynomial g(x). The coefficient of the highest-order term in the remainder is the first error-correction codeword. The coefficient of the highest-order term corresponds to the last codeword, which is also the last valid codeword in the module. If c(i)>-929, the minus number in GF(929) is equal to the complement of itself. If c (i)<=-929, the minus number is equal to the complement of the remainder (c(i) /929).

(4) Optimization of error-correction algorithm

He and Kang (2002) gave a method, which several coefficients in generated polynomials of error-correction codewords were determined by computer. In this method, every level of powers of three was calculated many times directly. For high levels of powers, overflow would occur because of large data and error would happen. It also was difficult to perform the storage (Dai,2004; Dai and Wu, 2006). Therefore, we adopted to calculate the mod of 929 for several coefficients and store them in a two-dimensional array, called coefficient[9][512]. The value of the two-dimensional array can directly be employed in calculation. The concrete procedure is as follows.

```
for(j=0;j<=8;j++)     // error-correction rank
{
int n=(int)pow(2, (j+1));// the number of error-correction codeword
const int mask=929;
int coefficient[9][512];// store the coefficient of g(x) after divided by 929
int i, k, p;
coefficient[j][0]=1;p=1;// initialization
for(i=1;i<=n;i++)
{
p=p*3%mask;// 3^i(mod929)
    coefficient[j][i]=0;// error-correction rank is j, the coefficient of g(x) x^i
for(k=i;k>=1;k--)// begin iteration
    {
        coefficient[j][k]= (coefficient[j][k-1]-p* coefficient[j][k])%mask
    }
coefficient[j][0]=-coefficient[j][0]*p%mask
}
for (i=0;i<=n;i++)
    coefficient[j][i]= (coefficient[j][i]+mask)%mask
}
```

3.4 Optimization of the arrangement of codeword matrix

The arrangement of codeword matrix should comply with the following rules:

3 <= the number of rows <= 90

1 <= the number of column <= 30

Filling-code number (Fillnum) should be as small as possible

Filling-code number (Finalnum) should be smaller than 2700

In the application of PDF417 bar codes, the reasonable utilization of space should be taken into sufficient consideration. The barcode matrix should economize valid space and also be user-friendly, so it is necessary to arrange the barcode matrix in optimal form (Sun, 2005; Zheng and Liu, 2006). The ratio of row to column can be user-defined, but in order to save space, the filling-codeword number should be ensured to be the smallest. The number of column (column_temp) can be determined temporarily by the row-column ratio (lwratio) and total number of codewords (totalnum), which includes data codeword and the error-correction codeword. The rule is as follows:

column_temp=(int)sqrt(totalnum/lwratio)+1.

When totalnum/lwratio is the square of some number r or the difference between totalnum/lwratio and r is not significant, we can use r as the column number of the barcode data codewords. Then, we add filling-codeword to the column as the preliminary column number. Because the image to be produced is saved in bmp format, it is requested that the byte number of each row must be times of 4. The rule is as follows:

```
if(column_temp%2==1)
{
 column=column_temp;
}
else
{
 column=column_temp+1;
}
```

By the row-to-column ratio, we can obtain the number of rows as follows:

$$row = lwratio * column \tag{4}$$

If neither the number of row nor column comply with the rule, in the other word, it is not in the range which the rule defines, a new row-to-column ratio is needed to be set unless the rule is complied with. Data codewords, filling codewords and error-correction codewords are put together in the matrix. According to the following formulas, the value of finders of left and right row can be obtained:

When imod3=0, Li=30Xi+Y, Ri=30Xi+V
When imod3=1, Li=30Xi+Z, Ri=30Xi+Y
When imod3=2, Li=30Xi+V, Ri=30Xi+Z
Where Xi=INT (Layers No./3), i=0, 1, ……89
Y=INT[（Layer No.-1）/3]
Z=error-correction rank*3+[(Layer No.-1) mod3]
V=number of columns every layer -1

Thus, the arrangement of the codeword matrix is accomplished.

3.5 Drawing according to the character set

The PDF417 barcode character set is composed of three clusters. Every cluster contains all the 929 PDF417 codewords represented in different bar-space form. Every symbol character corresponds to only one codeword in every cluster. The three cluster character sets are stored in two-dimensional array symbolset[3][929]. The row 0, 1 and 2 correspond to 0, 3 and 6 cluster in the character set, and the column 0 to 928 correspond to the sequence of bar-space corresponding to codewords 0 to 928, respectively. The sequence of bar-space can be obtained from codewords by searching the two-dimensional array symbolset[3][929], when a PDF417 image is drawn. It can be realized by the following program:

```
 // code_matrix[i][j]  matrix of codewords bar-space sequence
 // [code_temp[i][j]]  codewords matrix ////
 for(j=1;j<=column+2;j++)
 {
 for(i=0;i<row;i++)
 {
  if(i%3==0) // barcode symbol adopts the first cluster of character set
  {
     code_matrix[i][j] = symbolset[0][code_temp[i][j]];
  }
  if(i%3==1)// barcode symbol adopts the third cluster of character set
  {
     code_matrix[i][j] = symbolset[1][code_temp[i][j]];
  }
  if(i%3==2)// barcode symbol adopts the sixth cluster of character set
  {
     code_matrix[i][j] = symbolset[2][code_temp[i][j]];
  }
 }
}
```

Thus, the sequence of bar-space is obtained; thereupon, drawing can be done directly according to the rule.

4. CONCLUSION

The proposed improved algorithm of codeword byte compression was proven to be an effective algorithm for making PDF417 barcodes used for the Chinese characters, which realized encoding optimization for the same data information with less codewords, and improved the encoding efficiency of Chinese characters greatly. The coefficients of generated polynomial of the error-correction codewords were obtained by iterative calculation, and saved in a two-dimensional array. Since the value can be read directly from the array, the encoding process is greatly accelerated. Because the arrangement of barcode matrix was optimized, the barcode space was economized on condition that user's requirement was satisfied. Under the VC++6.0 programming environment, the encoding algorithm of PDF417 was realized by programming. For example, Figure 3 is a PDF417 barcode image produced by the developed software.

Fig. 3: A example of PDF417

REFERENCES

D. Chen, H. Liu. Two-dimensional Barcode Technology & Application, Chemic Industry Press, 2006

D. Zheng, E. Liu: Optimization of PDF417 Bar Code's generation. Packaging Engineering, 2006, 27(1): 84-86

GB/T 17172-1997. National Standard of the People's Republic of China: 417 Bar Code, China State Bureau of Quality and Technical Supervision, 1997

J Qi. Research on the Generation and Recognition of Two-dimensional Barcode, Graduate School of Harbin Engineering University, 2003

J. He, J. Kang. Computer Coding and Decoding of Bar Code, Computer Measurement & Control, 2002, 10(4): 263-266

J. Sun. Coding technology of PDF417 two-dimensional bar code and its implementation in Visual Basic. Journal of Xi'an Shiyou University(Natural Sciences Edition), 2005, 20(1): 77-80

S. Dai, X. Wu. PDF417 Error Correcting Code and its Implementation, Journal of PLA University of Science and Technology, 2006, 7(2): 137-140

Y. Dai. Research on the Application and Principle of Two-dimensional barcode'encoding and decoding , Nanjing University of Aeronautics and Astronautics, 2004