

Research on Recognition Method of Chinese-Sensible Code Under Clutter Background

Xinnan Fan, Ji Zhang, Bingbin Zheng, Qingyuan Zheng¹

Abstract The Chinese-Sensible Code is the first two-dimension code standard of China who totally owns proprietary intellectual property rights. It has been widely used in books, bank drafts and managements of armed police vehicles. However, due to the characteristics of application environment and interference factors, it's hard to ensure the accuracy and robustness of the recognition. In view of these, the paper proposed a recognition method of Chinese-Sensible Code under clutter background. Firstly, the barcode was positioned with two steps based on the general characteristics and scanning characteristics of barcode. Secondly, according to the center and angle inclination of the barcode, a rectification algorithm based on bilinear interpolation was used to restore image. At last, an edge projection segmentation algorithm and a double threshold analysis of projection results are proposed to recognize barcode. The result of the experiment showed that the proposed method can recognize the Chinese-Sensible Code under clutter background effectively and accurately, at the same time it has excellent ability of anti-interference to noise.

Keywords: Chinese-Sensible Code. Secondary positioning. Position detection pattern. Bilinear interpolation. Projection algorithm

1 Introduction

In the past few years, two-dimension code standards in China all came from foreign technology, such as America's PDF417 code, Japan's QR code etc. China didn't have its own proprietary intellectual property rights, and the strict patent protection led to expensive equipment and hidden danger in information security. Aug.23th, 2007, with complete proprietary intellectual property rights, China formally released a two-dimension code national standard "Chinese-Sensible Code"(CS Code), which was put into practice on Feb.1, 2008 [1]. The CS Code has the characteristics of high density, variable shape and large capacity for information. Moreover, the CS Code has strong ability of correcting errors, resisting fouling and distortion, expressing Chinese characters with high coding efficiency etc. [9]

¹ Xinnan Fan
College of internet of things engineering, Hohai University,
Changzhou, China

Ji Zhang (✉)
College of internet of things engineering, Hohai University,
Changzhou, China
e-mail: 497193712@qq.com

Bingbin Zheng
College of internet of things engineering, Hohai University,
Changzhou, China

Qingyuan Zheng
Production management department, Jiangsu xinyuan tobacco sheet LTD
Huaian, China

The CS Code structure diagram is shown in Fig.1. Compared with the previous two-dimension codes, the CS Code makes some changes in structure. For example, there is no continuous border in CS code; its four position detection graphs are not closed and arranged in different directions. These changes lead to difference recognition methods between the CS Code and traditional two-dimension codes.

For the CS Code recognition, it mainly includes three key steps which are positioning, inclination correction and single character identification. Only if the algorithm's efficiency of these three steps improved, the accuracy and speed of the final recognition can improve. In addition, it gives higher request to the method's adaptability and accuracy because of the change of the operating environment and the variety of image background. The clutter background usually includes words, forms, stains, scratches and so on.

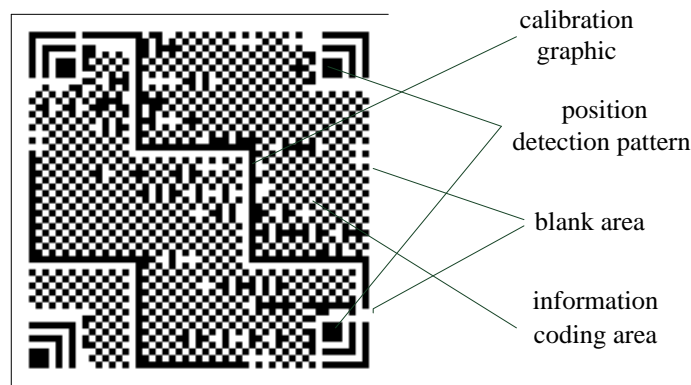


Fig. 1 CS Code structure diagram

Previously, the boundary positioning method was typically used to position the two-dimension code. References [5, 7, 13] get the code's left border by scanning from left to right, then get a straight line equation of boundary through Hough transform. Firstly, there is so much calculating work for Hough transform that this method won't process in real time. What's worse, CS Code is different from former stacked two-dimension codes because its boundary is not a straight-line continuum. Moreover, the complex background will affect the code positioning. Reference [4] uses the position detection to directly position the graphics features. However, when the code's proportion is very lower in the whole image, this method will need lots of time. And the proportion in the background can be as same as that of detection graphics, which will cause this method failed.

A lot of references have presented methods on two-dimension code's inclination correction [8, 10], but these methods are not suitable for CS Code due to the different structures of position detection patterns. It also needs an approximate treatment with an interpolation method in this process. Although the calculation is simple, the nearest interpolation will result in jagged edge, overlap and hole. But these won't appear on the bilinear interpolation [11].

In single character identification, the projection algorithm can increase square SNR by H times, thus it can reduce the influence of the noise during the identification [6].

Combining closely with the structural features and the application background, this paper proposed a recognition method of CS Code under clutter background. It aims at several key steps including positioning, inclination correction and single character identification during recognition. First, the secondary positioning is raised to position the barcode. Then, a new method is proposed for the inclination correction and an approximate processing by bilinear interpolation. Finally, this paper segments the code

with edge projection algorithm and identifies single character with double threshold analysis of projection results.

2 CS Code Positioning

The proposed method gains code's position by using secondary positioning, which consists rough positioning based on subdomains feature and accurate positioning based on position detection patterns feature.

2.1 Rough Positioning based on Subdomain Feature

CS Code's gray image is divided into $m \times n$ subdomains and each subdomain is marked as $A_{i,j}$ ($i = 0, 1, \dots, m-1$; $j = 0, 1, \dots, n-1$). The subdomains containing CS Code will be picked up according to the subdomains' contrast feature and linear scale feature. Finally, rough positioning will be done when the subdomains containing CS Code are merged.

2.1.1 Subdomain Feature

A) Contrast Feature

The CS Code is constructed by black and white squares. Therefore, if the subdomain $A_{i,j}$ contains a part of CS Code, it will obviously have the contrast feature of gray value, while the background has no the similar contrast feature in general case. According to this distinction, the parts with uniform gray value will be filtered out.

In the subdomain $A_{i,j}$, gray value k of each pixel point is calculated, and then a suitable threshold value T is chosen to class the pixel points into two classifications. Pixels with values greater than T will be labeled as bright and others will be labeled as dark, and the sums of pixels in the two classifications are equal. Define " n_k " as the sum of the pixels whose gray value is k in the sub-domain, and then the gray averages of dark and bright pixels are defined as follows:

$$G_{dark} = \frac{\sum_{k=0}^{T-1} k \times n_k}{\sum_{k=0}^{T-1} n_k}, \quad G_{bright} = \frac{\sum_{k=T}^{255} k \times n_k}{\sum_{k=T}^{255} n_k} \quad (1)$$

Define $Con(A_{i,j})$ as the contrast between dark and bright pixels' gray averages.

$$Con(A_{i,j}) = G_{bright} - G_{dark} \quad (2)$$

B) Linear Scale Feature

Due to the fact that CS Code is a matrix code, the subdomains containing parts of CS Code should include many line segments after edge extraction, and the quantity in one certain direction is far larger than that in other directions. After edge detection, the subdomain can be considered as a part of CS Code if the line segments' quantity is up to a certain number. On the other hand, subdomains, which contain lines or forms in other directions, or less numbers of line segments in the certain direction, will be filtered out.

Detect the edges of image by using Canny operator to get a binary image of these edges. After that, the Hough transform is used to detect all the line segments whose

length is larger than a certain threshold value in the binary image. Then, the line segments' quantity of every angle is got by statistical analysis and an angle θ_{\max} with maximum number of line segments is chosen as Target direction.

Finally, the line segments' quantity $Line(A_{i,j})$ of each subdomain in the target direction θ_{\max} is counted and define $Line(A_{i,j})$ as the linear scale feature of the subdomain.

2.1.2 Choosing and Merging the Subdomains

Contrast $Con(A_{i,j})$ and linear scale $Line(A_{i,j})$ are two features of the subdomain. In order to determine whether the subdomain $A_{i,j}$ is a part of CS Code, two rules are defined as follows:

$$\begin{cases} Con(A_{i,j}) > T_c \\ Line(A_{i,j}) > T_l \end{cases} \quad (3)$$

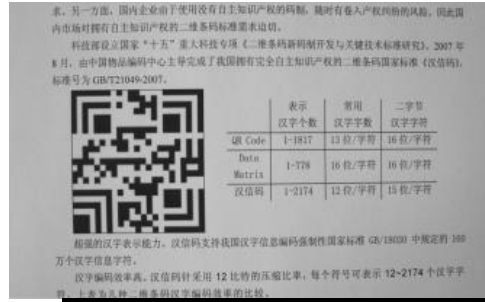
If both two features satisfy the formula, the subdomain $A_{i,j}$ is set as part of CS Code.

According to the variability of threshold, choosing threshold values can be carried out by using a fixed threshold or a floating threshold. Since the floating threshold is relatively more adaptable than the fixed threshold, OTSU algorithm is used to select the threshold value in this paper.

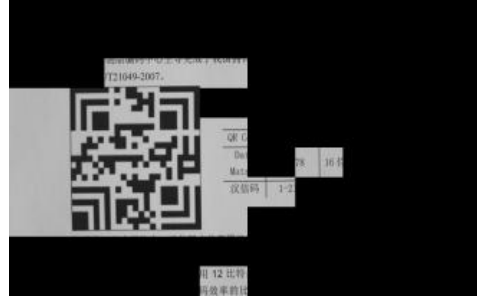
Depending on the connectivity between subdomains, cluster all the candidate subdomains into several connected domains which are labeled as S_1, S_2, \dots, S_N . Count the sum of subdomains in every connected region $S_i (i = 1, 2, \dots, N)$, and define the sum $\|S_i\|$ as the base of S_i . Remove the connected region with small base according to formula (4).

$$\|S_i\| < T_{area} \quad (4)$$

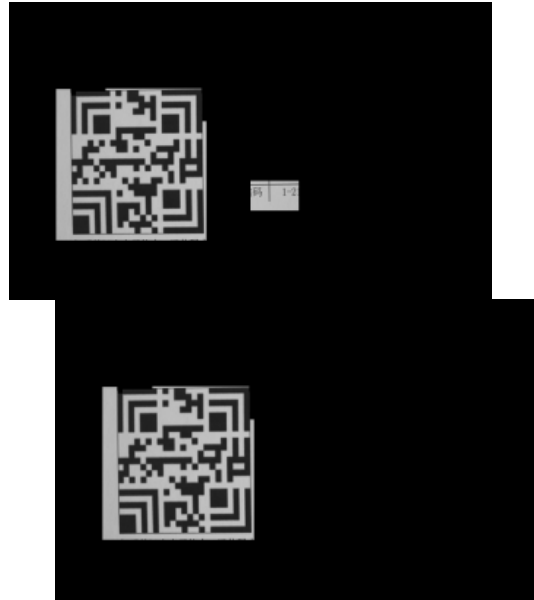
T_{area} is an empirical threshold value that the CS Code will be filtered out if it's too large and nothing will be filtered out if it's too small. Fig.2 shows the result of rough positioning, which is worked on the gray image of CS Code.



(a)



(b)



(c)



(d)

Fig. 2 Rough positioning: (a) Figure shows the gray image of CS Code (b) Figure shows the positioning result based on single gray value feature (c) Figure shows the location result based on double feature and (d) Figure shows the positioning result of screening and merging on subdomains

2.2 Accurate Positioning based on Position Detection Pattern Feature

After the rough positioning has been performed, the barcode will be located accurately on the basis of the position of image-finder graphics. The template matching method can be used to position the target, while it costs an expensive calculation and does not work when the size of the code changes [12]. This paper uses the scanning feature of position detection patterns to position the barcode. The image-finder graphic of CS Code is made up of four position detection patterns each of which is a 7×7 module

which is shown in Fig.3 (a). Its structure can be seen as a 7×7 dark module, a 6×6 light module, a 5×5 dark module, a 4×4 light module and a 3×3 dark module overlapped in turn. The center of the 3×3 dark module is the center of this position detection pattern. When a horizontal or vertical line goes through the center of this position detection pattern, the scanning characteristic is a dark and bright alternation with the ratio of 1:1:1:1:3 or 3:1:1:1:1 as shown in Fig.3 (b). This characteristic will not change regardless of the size or rotation, and it's unlikely to appear in other locations in the image. So the recognition of four position detection patterns can position the specific position of the code quickly and accurately.

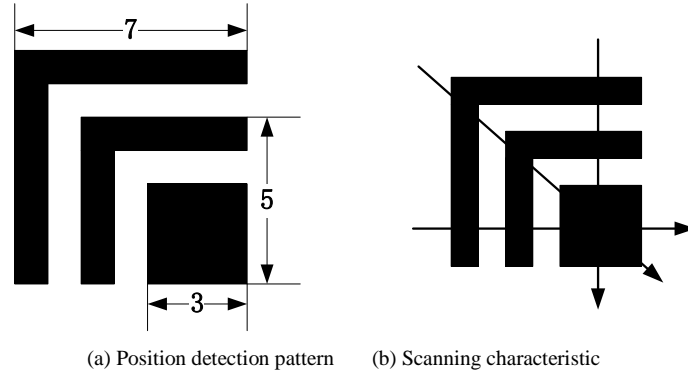


Fig. 3 Structure of position detection pattern

3 CS Code Inclination Correction

Combining the specific characteristics of CS Code, this paper presents a method for the inclination correction. Four centers of the position detection patterns are connected as shown in Fig.4 (a). According to the position and direction of four the position detection patterns, an equilateral right triangle is made up by three lines (a, b and e) as shown in Fig.4 (b). The vertex B is the center of the top right corner of the position detection pattern, and the intersection of two diagonals O is the code figure's center. The angle of inclination θ is worked out according to the equation of the straight line AB $x \cos \theta + y \sin \theta = \rho$. Then rotate the barcode figure around the center O.

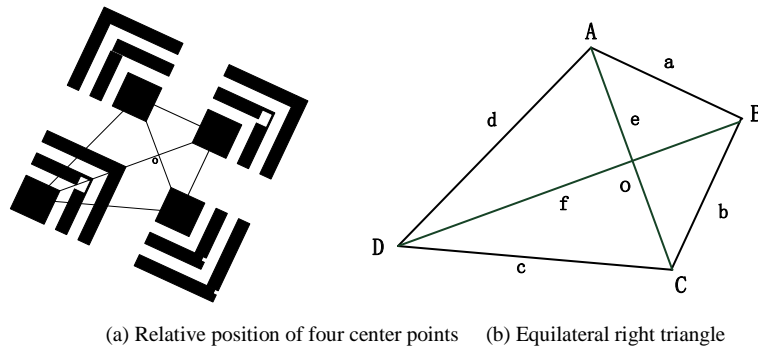


Fig. 4 Straight line joining the four center points

If all the points rotate anticlockwise around the origin, the transformation function is as follows:

$$x' = x \cos \theta + y \sin \theta \quad y' = -x \sin \theta + y \cos \theta \quad (5)$$

where (x, y) is original coordinate, (x', y') is rotated coordinate. The inversion function is:

$$x = x' \cos \theta - y' \sin \theta \quad y = x' \sin \theta + y' \cos \theta \quad (6)$$

With the matrix form said:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (7)$$

For a digital image, the coordinate is an integer; the pixels in rotated image may not find the corresponding pixels in the original image, so there must be an approximate treatment with an interpolation method.

This paper used the bilinear interpolation for the approximate treatment. The bilinear interpolation is also named as first-order interpolation. It is to count the distance between target coordinates and four adjacent points, then make use of distance ratio and four adjacent pixels' grey value for gray interpolation[2]. For a target pixel, after transformation the floating point coordinate is $(i + p, j + q)$, where i and j are nonnegative integers, p and q are floating numbers in the region of $[0,1]$. The pixel's value is determined by four adjacent pixels' gray values in the original image whose coordinates are (i, j) , $(i, j + 1)$, $(i + 1, j)$ and $(i + 1, j + 1)$:

$$f(i + p, j + q) = (1 - p)(1 - q)f(i, j) + (1 - p)qf(i, j + 1) + p(1 - q)f(i + 1, j) + pqf(i + 1, j + 1) \quad (8)$$

4 Single Character Identification

4.1 Code Segmentation

The cores of the code segmentation are edge detection and projection algorithm [3]. Based on the CS code's specific edge features, this paper detects the edge of the image with a Sobel level operator, and then conducts the projection of the horizontal direction [6]:

$$p(y) = \sum_x \Delta_x f(x, y) \quad (9)$$

Here Δ_x is the Sobel level operator:

$$\Delta_x f(x, y) = [f(x - 1, y + 1) + 2f(x, y + 1) + f(x + 1, y + 1)] - [f(x - 1, y - 1) + 2f(x, y - 1) + f(x + 1, y - 1)] \quad (10)$$

Because the peak value of $p(y)$ corresponds to the line boundary of code, the procedure of position $p(y)$ is as follows:

- Construct one-dimensional Gaussian template:

$$g(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right) \quad (11)$$

- Execute a convolution to $p(y)$ with Gauss function:

$$\phi(y) = p(y) * g(y) \quad (12)$$

- Compute difference calculation to get the first-order difference $\phi'(y)$ and the second-order difference $\phi''(y)$, the set of peak values is:

$$\{y_i \mid \phi'(y) = 0, \phi''(y) < 0, p(y) > k\} \quad (13)$$

where k is an empirical value.

After these above steps, the boundary of CS code and single character's width H are worked out, and the code is divided into single-line codes.

4.2 Single Character Identification based on Projection Analysis

According to the width H of character, this proposed method projects the single-line CS code in the direction of vertical and segments the projection result. The number of light color pixels point on column k of row i and column j (i.e. height of light projection) is defined as $h_{i,j}^k$. The threshold value is set as h_{th} . Calculate the number of $h_{i,j}^k$ which is greater than h_{th} in every single-line image and name it as $lines(i, j)$. So the ratio of $lines(i, j)$ to single character's width is:

$$l_{i,j} = \frac{lines(i, j)}{H} \quad (14)$$

In the end, a suitable threshold value l_{th} ($0 < l_{th} < 1$) is selected and compared with $l_{i,j}$. The value of each single CS code $S_{i,j}$ is determined according to the formula (15).

$$\begin{cases} S_{i,j} = 1 & l_{i,j} > l_{th} \\ S_{i,j} = 0 & l_{i,j} \leq l_{th} \end{cases} \quad (15)$$

In the experiment, the threshold value are set as $h_{th} = 1/2H$ and $l_{th} = 1/2$.

5 Experiment Comparison

5.1 Segmentation Experiment

The segmentation result of the projection image is shown in Fig.5, the numbers beneath correspond to the serial number of a single characters image. The segmentation is processed with the width of character H and the segmentation object is the projection result rather than the barcode image. In the figure, it's easy to see that the segmentation result is clear and conducive to next experiments.

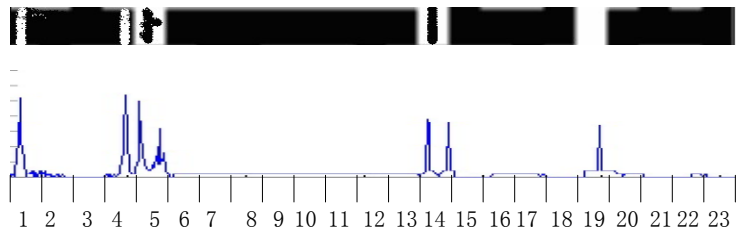


Fig. 5 Segmentation result of projection image

5.2 Single Character Identification based on Projection Analysis

A detailed analysis is made on the fourth polluted barcode character in Fig.5 after segmentation and amplification. As shown in Fig.6, Fig.6 (a) is the fourth barcode character and Fig.6 (b) is the result of projection, it's a little blurry due to the enlargement effect. Here the threshold value are set as $h_{th} = 1/2H$ and $l_{th} = 1/2$. It is apparent that $l_{i,j} < 1/2$ so the value of this character is 0 ($S_{i,j} = 0$), which means dark module.

Fig.7 is the result of Fig.5 after double-threshold analysis of projection result. It can be seen that discriminations are all correct even when the fouling is more than half of column number and no more than half of width. To a large extent, this method strengthens anti-noise and anti-fouling ability of the system and improves the robustness.

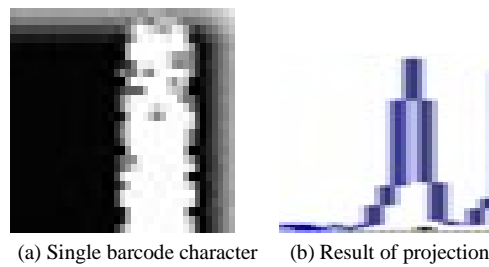


Fig. 6 Analysis on single character projection result

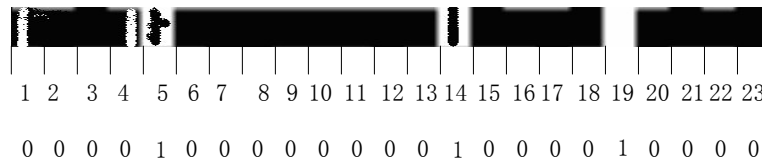


Fig. 7 Example of single character identification

5.3 Contrast Experiment between New and Old Algorithm

This paper does a contrast experiment between new algorithms raised in this paper and old algorithms already raised by former scholar on 100 shot CS Code images under clutter background. Because no method existed is suitable for the CS Code on the inclination correction, the experiment is divided into three separate contrasts.

Table 1 shows that only three images cannot be positioned due to the vague border while old algorithm can only position 94%. The 97 successfully positioned images can all make inclination correction done. During the step of single character identification, it counted the ratio of characters identified right to all the characters in 97 images. And the accuracy this paper raised is up to 98.17% while the old is just 91.36%.

Table 1 Comparison of different algorithm

	New algorithm	Old algorithm
Positioning	97%	94%
Inclination correction	100%	—
single character identification	98.17%	91.36%

On the premise of no data-level decoding, the correct rate is up to 95.8%. Combination the error correcting code of data-level, the algorithm of this paper can recognize CS Code under more complex background and with higher pollution degree.

6 Conclusions

This paper proposed a recognition method of the CS Code under complex background focusing on three big problems of the CS Code recognition that are positioning, inclination correction and single character identification,. The proposed method gets a standard barcode through the secondary positioning and a new inclination correction method with the bilinear interpolation, and then identifies single character with the double threshold analysis of projection results. After simulation experimental comparison with old algorithm, the result shows that the algorithms proposed in this paper can position and identify the CS code accurately and efficiently, and it has a strong anti-interference ability for noise and violation, so this method is suitable for the CS Code recognition under complex background.

Acknowledgments This work was supported by the National Natural Science Foundation of China (No.61273170), and Jiangsu Key Laboratory of Power Transmission and Distribution Equipment Technology Construction Project (No.BM2009704). This project was also sponsored by Qing Lan Project for Middle-aged and Young Science Leaders of Jiangsu Province, China.

References

1. SAC. GB/T21049-2007 Chinese-Sensible Code. China Zhijian Publishing House, 2008.
2. Dianguo Cao, Haojie Chen, Peng Li. Application of Bilinear Interpolation Algorithm in Image Rotation Based on Matlab. China Printing and Packaging Study, 2010, v.204: 74-78.
3. Zheyi Fan, Fan Jiang, Zhiwen Liu. Recognition of PDF417 Barcode Based on Captured Images. Transactions of Beijing Institute of Technology, 2008, 28(12): 1088-1092.
4. Huijuan Liu. Omnidirectional Recognition of Quick Response Code Image. Chinese Journal of Scientific Instrument, 2006, 27(4): 376-379.
5. Ningzhong Liu, Jingyu Yang. Recognition of Two-dimension Barcode Based on Projection Algorithm. Computer Engineering, 2002, 28(9): 32-34.
6. Ningzhong Liu. Research on Recognition Technology and Encoding Theory of High-dimensional Bar Code. Nanjing University of Science and Technology, 2003.
7. Wei Liu. Research and Application of Chinese-Sensible Code Encoding and Decoding. Nanjing University of Aeronautics and Astronautics, 2010.
8. Xin Liu. Design and Research of PDF417 Barcode Recognition Based on Embedded Platform. Hunan Normal University, 2012.
9. Xiaochun Ma. Chinese-Sensible Code open a new era for China's two-dimension barcode development. International Business Daily, 2011-04-20013.
10. Ming Sun, Fu Long-sheng. Image Analysis Method for QR Code's Automatic Recognition. Journal of University of Electronic Science and Technology of China, 2009, 39(6): 1017-1020.
11. Xiaohui Xu, Bingzhong Wang, LIU Zhi. A method of 2D barcode skew correction using least square methods. Journal of Zhejiang University of Technology, 2008, No.13303: 250-253.
12. Jiayao Yang. Research on Embedded 1-D Barcodes Recognition System Based on Image Processing. Changsha University of Science&Technology, 2012.
13. Peng Zhang, Shuguang Dai, Ping'an Mu. Research on information recognition method of image for PDF417 two-dimension bar code. Proceeding of the 7th World Congress on Intelligent Control and Automation, June 2008: 5835-5837.