

## A Simple and Efficient Image Pre-processing for QR Decoder

Weibing Chen<sup>1,a</sup>, Gaobo Yang<sup>2,b</sup> and Ganglin Zhang<sup>1,c</sup>

<sup>1</sup>Dept. of Electrics & Communication Engineering, Changsha University, Changsha Hunan, 410003, P. R. China

<sup>2</sup>School of Information Science and Engineering, Hunan University, Changsha Hunan, 410082, P. R. China

<sup>a</sup>cwb680625@sina.com, <sup>b</sup>yanggaobo@hnu.edu.cn, <sup>c</sup>ccsuhfj@163.com

**Keywords:** 2D barcode; Quick Response; image preprocessing; localization; binarization

**Abstract.** In order to lower the consumed threshold, a practical image preprocessing method was proposed for Quick Response (QR) barcode recognition. It could increase the speed of recognition by this decoder so as to embed this algorithm into mobile terminals. Instead of using the traditional methods such as edge detection and line detection, the encoding characteristic of QR had been used, thus the influence by background noise and geometric distortion was minimized. Moreover, it used alignment patterns to adaptively sample the barcode in terms of regions, which greatly improved the recognition rate. Experimental results demonstrate that the proposed approach can overcome the influence in noise, inhomogeneous light and geometric distortion, what is more, it meets the requirement of decoding in real time.

### Introduction

QR code is a kind of matrix 2D (two-dimensional) barcode designed by Denso-Wave Corporation of Japan in 1994. Its benefits are high-speed recognition, strong error-correcting capability, effectual expression for Chinese characters and all-directional recognition.

Due to the influence of capturing environment and resolution, these 2D barcode images captured by scanner, digital camera, or camera head have some shortcomings, such as spherical distortion, geometric distortion, image blurring, and inhomogeneous light. Furthermore, 2D barcode is even partly defiled so that it is encoded in error or can't be recognized. In order to increase the reliability of recognition for 2D barcode, the captured images must be pre-processed [1]. In reference literatures about barcode recognition [2], most of them adopt such conventional method: The first step is resolving the angle of deflection by edge detection and Hough transform, then rotating the image to keep horizontal, and lastly implementing bilinear interpolation for the image. However, because this kind of method is excessively rely on edge detection and line detection based-on Hough transform, the recognition will often be failing in addressing when the QR code image has serious geometric distortion. In additional, Ming *et al.* [3] proposed a technique to eliminate the illegible effect by the binarization with self-adaptive threshold according to the features of defocused QR code image. The corner points are initially located by contour search algorithms, and a QR code image is rebuilt from a distorted one by using a double tangent method designed to approximate the right-bottom point with double inverse perspective transformations. Xiao *et al.* [4] held the point that the bilinear transform is referred for the rectification of geometry distortion of QR Code images. Then with the research in the characters of QR code symbols, a simple, effective and high-speed method of localization and orientation is brought up. The common disadvantage of these methods is that they need so much floating point calculation as to effect recognition speed. In recent years, 2D barcode services based on mobile terminal have been widely used. However, the most of algorithms are too complex to be used in embedding mode.

In this paper, for the sake of lowering the threshold of QR code, a low-complexity and practical image pre-processing algorithm is researched for QR barcode recognition according to its design feature. This algorithm is easy to be embedded into mobile terminals.

## Structure and Decoder Procedure of QR Code

As shown in Fig. 1, a QR code symbol is a square array which consists of some square modules. It comprises an encoding region and a function region (includes finder patterns, separator symbols, timing patterns and alignment patterns). Function region shall not be used for the encoding data. The surrounding region of a QR code symbol is blank space. The finder patterns (includes three probing patterns) located at three corners of the symbol intend to assist in easy location of its position, size and inclination. There are separator symbols width as a module between each probing pattern and the encoding region. All of them consist of light modules. The functions of timing patterns are determining the density and version of QR Code symbol, and providing the reference position which can decide coordinates of modules. The specific encoding standard of QR code is detailed in Ref.[5].

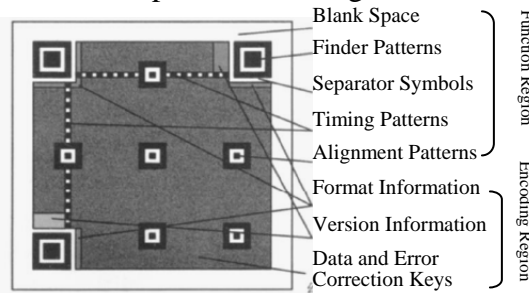


Figure 1. The structure of QR code symbol

The structure features of QR code symbol must be adequately used in its decoder procedure. The flow chart of QR code recognition is shown as Fig. 2. It mainly includes follow steps:

- Binarization;
- Obtain the approximate region of QR code, and implement coarse positioning for QR image according to the finder patterns;
- Implement accurate positioning according to the alignment patterns;
- Calculate the angle of inclination to rotate QR image, and implement rectification processing;
- Obtain version number and implement self-adaptive sampling;
- Decode based on corrected image and input a standard 2D matrix.

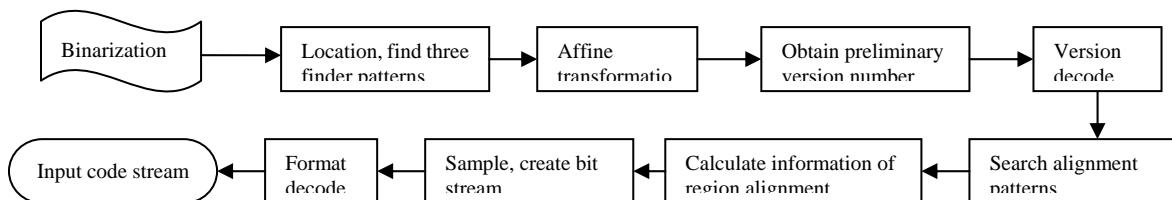


Figure 2. The decoder flow chart of QR code

## Image Pre-processing Algorithm of QR Code

**Binarization.** Many algorithms may be used for image binarization [6]. Binarization is the base of QR code decoding and directly influences the recognition rate of QR code. However, it must keep balance between real-time and effects. In this paper, dynamic iterations method is used to implement dimensional histogram statistic for inputted grey-scale image data and obtain the optimal threshold value. Then, this threshold value can be used to implement binarization processing for the image. Supposing  $T(i)$  as a new separator threshold value,  $V_1(i)$  as the mean gray value of pixels which gray values are less than  $T(i)$ , and  $V_2(i)$  as the mean gray value of pixels which gray values are more than  $T(i)$ , its iterations processing includes following steps:

- $T(0)=128$ ;
- $T(i+1)=[V_1[T(i)]+V_2[T(i)]]/2$ ;
- Return prior step. If  $T(i+1)=[V_1[T(i+1)]+V_2[T(i+1)]]/2$ , then exit this iterations processing, where  $T(i)$  is the calculated optimal threshold value for binarization.

**Localization of QR Code.** The localization of QR code gets help from finder patterns of its most marked feature. According to the international standard of QR code encoding, Finder patterns may be obtained from the relationship which the ratio between black blocks and white blocks is 1:1:3:1:1. Therefore, we can scan whole image to obtain the center positions of three finder patterns. Steps as follows:

- Scan the pixels of each row (in order to increase recognition rate, it can be scanned once interval several rows), and record the length of black blocks and white blocks, then judge whether black : white : black : white : black equals to 1:1:3:1:1 in each row.
- If a row is the same with this ratio, move the reading image pointer on to the row center position of this middle black block, and measure the length of a white block and a black block by extending respectively toward up and down, and then judge whether the ratio between five black blocks and white blocks is also 1:1:3:1:1.
- If its column is also the same with this ratio, it is similar with finder pattern. In order to try more verifies for this point, adjust this cross-point of row and column scan line on to the center position of black block, and then play next step.
- Considering that single judgment condition isn't suitable for different distortion images, thus, it would be three judgment conditions are used in turn by controlling with variable pImgCounter : (1) case 0: the length ratio between back blocks and white blocks is 1:1:3:1:1 in transverse and vertical directions, and is also 1:1:3:1:1 in diagonal directions; (2) case 1: the length ratio between back blocks and white blocks is 1:1:3:1:1 in transverse and vertical directions, and the ratio between left diagonal line and right diagonal line is 1:1; (3) the ratio between transverse line and vertical line is 1:1, and the length ratio between back blocks and white blocks is 1:1:3:1:1 in left and right diagonal directions.

According with the judgment condition, the pattern can be regarded as a finder pattern. The specific finding processing is shown as Fig. 3.

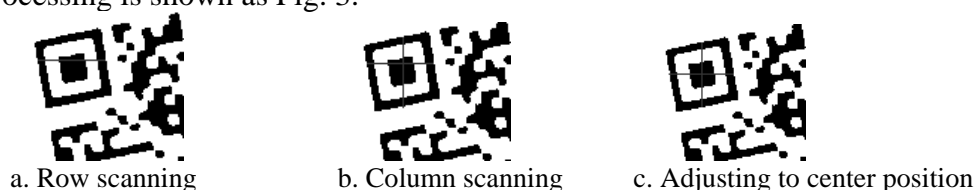


Figure 3. The search path of finder patterns

**Geometric Rectification.** Geometric rectification refers to follow processing: when an image occurs geometric distortion, for example, QR code occurs gradient by an unsuitable shooting angle, the image is corrected to a square. Due to knowing image's three peaks by prior location of finder patterns, we can use the affine transformation formula with six parameters to implement geometric rectification for image, and obtain new image data.

The effect of geometric rectification based on affine transformation is shown as Fig. 4. It is remarkable that the QR code after geometric rectification possibly is not a true square and still has a few of distortions. Especially in the case of trapezoidal distortion, the distortion in top left corner could be minimum, and the distortion in lower right corner is maximum. It needs more treatments in following sample processing.



Figure 4. Affine transformation

**Localization of Alignment Patterns.** After the localization of finder patterns, it is easily to get the height and width of each finder pattern. Then divided by 42 after add them together. The result would be average width of modules. Lastly the version number also can be obtained. The QR codes which version number is greater than 2 have alignment patterns. Shown as Fig. 1, so-called alignment pattern refers to a square region which ratio between blacks and whites is 1: 1: 1: 1: 1. Then, versions from 2 to 6 will be taken example for illustrating the search and localization of alignment patterns. Steps as follows:

- Estimate the approximate position of alignment patterns by version number.
- Move image pointer near to this point, and search the center coordinate of a corrected image. If this center can be found, then return to this point; if it can not be found within the specific region, then take estimation value as standard.
- Calculate the center coordinates of three finder patterns, and the widths of vertical module and transverse module of the quadrangle comprised of alignment patterns.
- The center point is respectively moved the width of 3 transverse modules toward left and the width of 3 vertical modules. Then the fourth point can be obtained.
- According to the quadrangle comprised of the center coordinates of three finder patterns and the fourth point, the distortion degree from left to right and from up to down can be estimated.

The method of accurate localization for alignment patterns as follows:

- First judge whether the estimated point is black. If it is black, then judge whether there are over 7 white points in 8 adjacent modules around it. However, if this condition is satisfied, then the estimated point can be regarded as located within alignment patterns. On next step, directly adjust this pointer on to center position and return.
- If above method is failure, then the estimated point can be regarded as located outside the center module of alignment patterns. On next step, search in 3 adjacent left and right modules. There, the judgment conditions of alignment pattern are follows: (1) whether this point is black; (2) whether there are over 7 white points in 8 adjacent modules around this point; (3) whether there are over 14 black points in 16 adjacent modules around this point.

In Fig. 5 (a), the sampling region and alignment patterns' numbers of version 14 are shown. With region alignment information, subsequent sampling grids can be equably distributed in a region, and sampling can be implemented in the region between alignment patterns.



Figure 5. (a) The sampling region and alignment patterns' numbers of version 14; (b) The self-adaptive sampling for the QR code with alignment patterns

**Image Sampling.** Image sampling refers to follow processing: according to located QR code, obtain the version number to determine the size of QR code, and obtain that the bit represented by each grid is 0 or 1. Shown as Fig. 5 (b), how to divide an image into grids is the key point. Considering the QR code image after geometric rectification still has some distortion, thus self-adaptive sampling is implemented in conjunction with alignment patterns. According to the version number of QR code, sampling is implemented in follow 4 cases:

- Version 1: without alignment pattern, sample in one region;
- Version 2-6: with only one alignment pattern, sample in one region;
- Version 7-13: with 6 alignment patterns, respectively sample in 4 regions;
- Version 14: with 13 alignment patterns, respectively sample in 9 regions.

Fig. 5 (b) is the self-adaptive sampling diagram for the QR code with alignment patterns. It is perceptible that the pixels of each grid aren't completely same. How to judge the bit represented by each grid is 0 or 1 has two methods: (1) the ratio between all black pixels and all white pixels in a grid; (2) drawing a "+" symbol on the center of grid, then judge the ratio between black pixels and white pixels on the cross symbol. In this paper, the second method was applied. Its effect is shown as Fig. 5(a). Experimental results demonstrate that the recognition rate of the second one is a little higher than the first one's.

## Experimental Results and Discussion

To test the effect of this paper' algorithm, we realized the encoder and decoder of QR code with Visual C++ 6.0. The experiment was performed under the following environment:

- An ordinary camera *i.e.* AC650 with USB interface is used. It has 24 bits true color, and its resolution factor may be adjusted from 160×120 to 640×480;
- Multiple illumination intensities, such as ordinary light, dim light, floodlight, side light;
- 5 kind of module widths are applied to simulate the density of QR barcode;
- To simulate geometric distortion, the distance between camera shot and barcode varies from 8 to 20 cm, and the separation angle between camera plane and barcode plane varies from 45°-90°, and shooting angle is omnibearing.
- To simulate a defiled barcode, draw lines on a barcode, and knead the barcode;
- The error correcting level varies among H (30%), Q (25%), M (15%), and L (7%);
- The version number varies from 1 to 14.

200 selected barcodes were tested under above conditions. Experimental results demonstrate that the right recognition rate is over 95%, *i.e.* this paper' algorithm can overcome the influence by noise, inhomogeneous light and geometric distortion. The recognition rates under different conditions have small difference. Except encoding content, the main influencing factors are the resolution of camera and error correcting level. General speaking, the recognition time for QR image with 320×240 or 352×288 is about 5 ms. It completely meets the requirement of decoding in real time. Compared with the experimental results reported by Ref. [3] and [4] (recognition rate as about 90%, recognition time as about 80 ms), the algorithm mentioned in this paper earns more benefits.

## Conclusions

The image pre-processing algorithm is the key to extend the applied range of 2D barcode and lower used threshold and increase degree in practical. In this paper, a practical image preprocessing method was proposed for QR barcode recognition. It didn't utilize the traditional methods such as edge detection and line detection, thus the influence by background noise and inhomogeneous light and geometric distortion was minimized. In additional, it used alignment patterns to adaptively sample the barcode in terms of regions, which greatly improved the recognition rate. Because the number of calculation of this algorithm is reduced, its recognition speed is higher. But this algorithm was only simulated on PC and not truly tested in embedding system, so further work is the optimization of program codes, and embedding it into mobile terminals, and developing correlative research.

## Acknowledgment

This work is supported in part by the Science and Technology Program of Hunan China under Grant 2011FJ3116.

## References

- [1] H. Kato, K.T. Tan: IEEE Pervasive Computing, vol.16(2007), p.76-85.
- [2] D.W. Liu: Ocean University of China, Qindao (2008).(in Chinese)
- [3] A.L. Ming, H.D. Ma, Q.P. Zhao: Journal of Computer-Aided Design & Computer Graphics, vol.19(2007), p.1080-1084. (in Chinese)
- [4] X.Xiao, X.M. Lau, Y.K. Wang: Computer Science, vol.34(2007), p.217-219.(in Chinese)
- [5] Y. Liu, M.J. Liu: Proc. of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), p.783-788 (2006).
- [6] O.D.Trier: IEEE PAMI, vol.17(1995). p.1191-1201.