

Research and Application of the EAN-13 Barcode Recognition on Iphone

YeMin Li

College of Computer Science and Technology
Sichuan University
Chengdu, Sichuan Province, P.R.CHINA
liyemin88@126.com

Li Zeng

College of Computer Science and Technology
Sichuan University
Chengdu, Sichuan Province, P.R.CHINA
zl801014hj@gmail.com

Abstract - Barcode is worldwide used, its popularity can be attested by its use in all parts of our life. Though the laser barcode scanner is very popular, yet it is lack of mobility. So we want a portable scanner as a substitute, without a doubt, iphone is a good choice. In this paper we present an algorithm for the EAN-13 barcode recognition on iphone. The algorithm can be divided into four parts: image pre-processing, edge detection, image segmentation and image decoding. Tests showing that in most light condition, this algorithm can extract the barcode with very high accuracy.

Index Terms - barcode recognition; EAN-13 barcode; iphone; image processing

I. INTRODUCTION

With the rapid development of the barcode technology in recent years, almost every consumer product has a barcode label. It is being widely used in many fields for applications of great commercial value. We know that a laser barcode scanner can obtain the information about a product such as price and description by the product's barcode quickly. But the laser barcode scanner has one deficiency, it is lack of mobility.^[1] Nowadays, more and more people use iphone as their smart phones, no doubt the use of iphone is becoming increasingly widespread, and its features also are rapidly growing. Iphone 3GS can take high-quality photographs easily because it has a three million pixel camera. So we can use iphone as a mobile platform for decoding barcode rather than the use of the conventional scanner which is lack of mobility, after that we can use the decoded barcode as a key to search and update the goods information by the shopping guide platform, this includes not only product description and price but also product review, price comparison, location of retailers, etc. Of course, the shopping guide platform must be established firstly.

In this paper, we present an image processing algorithm which rely on knowledge about structure and appearance of 1D barcodes. It is capable of interpreting a particular type of barcode, called EAN-13, take by the camera phone. Our contribution is an algorithm which is both fast and robust

This paper is organized as follows: a brief introduction of the EAN-13 barcode is given in section two; barcode recognition algorithm is presented in section three;

experimental results can be found in section four. Finally, section five gives a conclusion of this paper.

II. THE EAN-13 BARCODE

EAN-13 barcode is used worldwide for marking retail goods. The numbers encoded in EAN-13 barcode are product identification numbers. EAN-13 barcode is a (7,2) code, that's means each character has total width of 7 modules, composed of two bars and two empty bars alternately, and the width between each bar and empty is no more than 4 modules. EAN-13 barcode is formed by the left blank area, start character, left data character, intermediate separator, right data character, check character, terminator character and right blank area. An example of the EAN-13 barcode is shown in Figure 1.

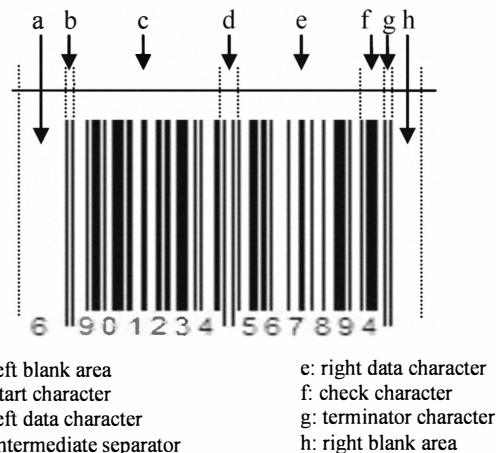


Figure 1: EAN-13 barcode

EAN-13 barcode has 13 numeric characters, also known as digits. The first two or three characters are the country code. Depending on the length of the country code, it followed by either nine or ten characters for manufacturer code and product code. The 13th character of the EAN-13 barcode is the checksum digit.^[2]

The EAN-13 barcode is a graphical rendering of a 13-digit code in the form of line pattern. One can consider such line pattern as being constructed using dark bars and light spaces of

various thickness, but for better clarity we simply treat the pattern as alternating black and white bars of various widths.

III. BARCODE RECONGNITION ALGORITHM

In the whole process of barcode recognition, we assume that each photograph taken by the iphone should contain the whole barcode, because this will very well reduce the computation.

The barcode proposed algorithm, as depicted in Figure 2, is divided into four parts: image pre-processing, edge detection, image segmentation and image decoding.

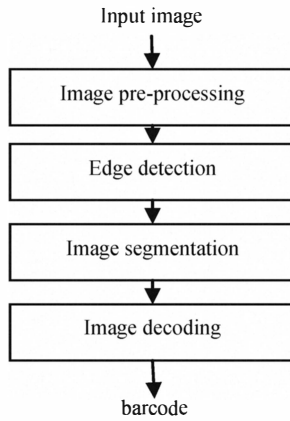


Figure 2: the proposed algorithm

A. Image Pre-processing

- 1) **Gray-scale processing:** It provides the preconditions for the threshold of the digital image transformation. Gray describes any shade between black and white collectively, white, black and the range of grays between them are known as achromatic colors or neutral colors.^[3] We use the relationship between gray and RGB, $Y = 0.299R + 0.587G + 0.114B$, to change 256 color bitmap into gray scale image. The result is shown in Figure 3(b).
- 2) **Image sharpening:** In order to make the image become clearer, we should do some sharpening process to the blurred barcode image. The essence of the blurred image is the attenuating of the high-frequency component. So we can apply laplacian sharpening to sharpen the gray scale image. Laplacian operator is one of the commonly used image sharpening operators, it can be described by the following formula

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} = f(i-1, j) + f(i+1, j) + f(i, j-1) + f(i, j+1) - 4f(i, j) \quad (1)$$

Where $f(i, j)$ represents the pixel value of a gray scale image.

According to (1), the Laplacian operator reflects the variety of one pixel relative to its surrounding four pixels. When it is used to get the sharpened image, the sharpened pixel value is

$$F(i, j) = f(i, j) - k\nabla^2 f(i, j) \quad (2)$$

where k is parameter concerning with diffusion effect, and it should be chosen reasonably according to the practice situations, since it affects the sharpening effect directly. The result is shown in Figure 3(c)

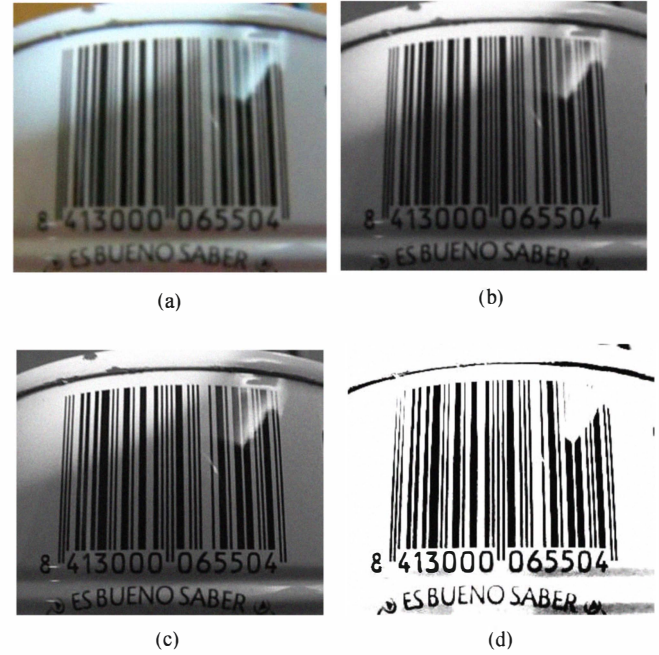


Figure 3: Image pre-processing. The original image is shown in (a). Gray-scale processing result is shown in (b). Image sharpening result is shown in (c). Image binarization result is shown in (d).

- 3) **Image binarization:** it's means that we should convert the gray-scale image into binary image through the threshold transformation theory. For the binarization a dynamic threshold is required^[4], also, it is very important. We are most concerned about the barcode in the whole image, so we get an image of 80*80 pixels in the center, after that we get the gray histogram, finally, we obtain the threshold T through the two-peak method. The threshold T can effectively separate bars and empty bars of the barcode. In accordance with the following formula we can obtain the binary image $g(x, y)$

$$g(x, y) = \begin{cases} 255, & F(x, y) \geq T \\ 0, & F(x, y) < T \end{cases} \quad (3)$$

The result is shown in Figure 3(d).

- 4) **median filtering:** There are many noise in the binary image because the brightness of the original image are

not symmetrical and the background images are not the same, not only in the area of the barcode, but also in the outside of the region. In order to get the right barcode, we should do some filter processing. Since the characteristic of the EAN-13 barcode, median filtering is a wonderful method. First of all, we get another four pixels around the pixel which will be processed, two in the above and two in the following; then, we sort the five pixels; at last, we use the median instead of the current pixel. In this way, salt and pepper noise is removed, and background area of the noise can be inhibited. The module is shown in Figure 4.

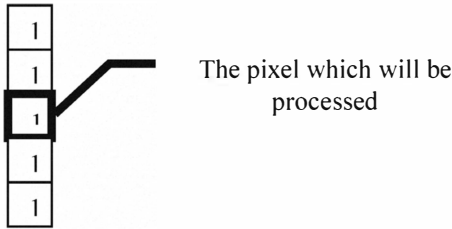


Figure 4: median filtering module

B. Edge Detection

- Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames as a precursor step to feature extraction and object segmentation.^[5] Examples of edge detection are Prewitt, Roberts and Sobel operators. Since the characteristic of the EAN-13 barcode, we use vertical-based edge detection operator, the module can be seen in Figure 5.

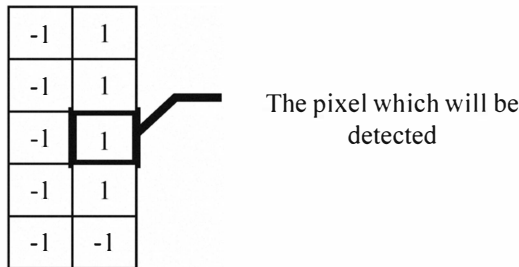


Figure 5: edge detection module

- The pixel which will be detected is determined by ten neighborhood pixels. The value of $S(x, y)$ is absolute of the sum of ten modules. Since the edge detection image is an binary image, either white (255) or black (0), we set the threshold $t = 255 \times 4 = 1020$, so if the pixel is called an edge, there are must at least four black and white mutations in this neighborhood. In accordance with the following formula we can obtain the binary image $h(x, y)$

$$h(x, y) = \begin{cases} 255, & S(x, y) \geq t \\ 0, & S(x, y) < t \end{cases} \quad (4)$$

C. Image Extraction

Image extraction is to separate out the real barcode, preparing for the decoding.

- Upper and lower division:** We know that EAN-13 barcode has 59 black and white areas ($3+6*4+5+6*4+3$), so the edge num is 60. Then we design this method to remove the blank area of the upper and lower part of the barcode. In the first, we get the binary image S1 after edge detection; then, we scan image from the middle line, go above line by line until the number of edge less than 60, now, we record the coordinate as y1; using the same method, go down line by line until the number of edge less than 60, at this time, we record the coordinate as y2.
- Left and right division:** Left and right division means divide the barcode in row direction, we design this division algorithm based on the following two features. First, the start character and the terminator character of the EAN-13 codes are 101. Second, left blank area has 11 modules, right blank area has 7 modules. In the first, we get the binary image S2 after upper and lower division; second, scan the left part of this image from left hand, if there are eleven blank modules in the left of 101, then, record the coordinate as x1; finally, scan the right part of this image from right hand, if there are seven blank modules in the right of 101, then, record the coordinate as x2.

Through the two methods above, we can divide out the real barcode by x1, x2, y1, y2. As shown in Figure 6.



Figure 6: Image after division

D. Decoding

Sometimes, the image may be deformed or distorted. In order to obtain the correct results, we should calculate the value by using the average data in the following steps.

- Calculate the width of one module:** we know that the start character is 101 (black white black), and in its left are 0 of eleven modules. So we check from left to right, and record three distance of consecutive edges as $d1$, $d2$, $d3$, that means we get 101 of the start character. Obviously, the width of one module can be record as $w = d1 + d2 + d3 / 3$. At the same time, we can get how much pixels in one module, record as $pNum$.

- 2) **Record the value in one row:** Check from left to right. We record a value after each distance on w , if the average pixel value is similar to 0 (there are $pNum$ pixels in w , use the average pixel value), we record $info[i]$ as 1; if the average pixel value is similar to 255, we record $info[i]$ as 0. Finally, the sequence of black bar and white bar is recorded in $info[index]$.
- 3) **Sequence division:** $info[index]$ contains the information of start character, left data character, intermediate separator, right data character and terminator character. We should separate $info[index]$ for each part.
- 4) **Decode the barcode:** After sequence division, we can decode the barcode by looking the EAN-13 coding table, details of the EAN-13 coding table can be found in [6].
- 5) **Checkout:**

IV. EXPERIMENTAL RESULTS

In this section, the experimental results are described based on the previous implementations. To perform experiments on the go, we implement our algorithm in the platform of XCode 3.1.3 on Mac OS by using the language of Objective-C, then import the program to iPhone 3GS.

We take 400 images by using the iPhone 3GS in different places and lighting conditions, 100 of them are taken indoors during the daytime, 100 of them are taken outdoors during the day, 100 of them are taken indoors at night, others are taken outdoors at night.

Among the 400 images, 26 of them can't be decoded correctly, with an average recognition rate of 93.5%. The results are shown in Table 1:

Table 1: the results of barcode recognition

Data items	Indoors during the daytime	Outdoors during the daytime	Indoors at night	Outdoors at night	total
Total images	100	100	100	100	400
Correct decoding	98	92	95	89	374
Recognition rate	98%	92%	95%	89%	93.5%

The errors come from two factors:

One is from the image brightness, because of the recognition accuracy of the boundary between the black and white parts, the lighting conditions are very important.

The other is the form of the image, if the image is a severely distorted one, it will get the wrong result after edge detection and image Segmentation, not to mention the correct barcode.

Figure 7 shows an example of experimental results using our algorithm.

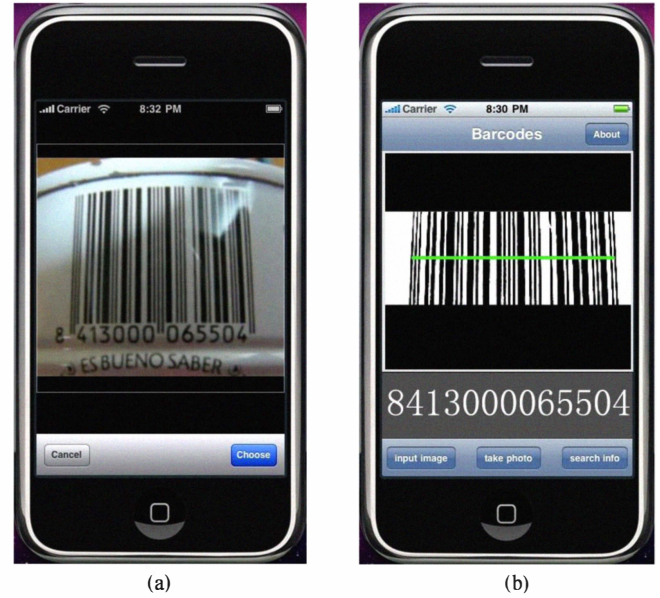


Figure 7: example of experimental results. The original image is shown in (a). The result is shown in (b).

V. CONCLUSION AND FUTURE WORK

In this paper we presented an algorithm for barcode recognition on iPhone. As shown in experimental results, our algorithm has a high recognition rate. In the future, we will focus on two things to make the barcode recognition more accurate and useful: one is improving the algorithm to shorten the decoding time; the other is establishing a shopping guide platform by using the barcode as a key to allow users to easily search and update the goods information.

ACKNOWLEDGMENT

We would like to thank all the members of our laboratory for their technical supports.

REFERENCES

- [1] K.Wang, Y.Zou and H.Wang, "bar code reading from images captured by camera phones". Mobile technology, application and systems, 2005 2nd international conference on 15-17 Nov. 2005. 6 pp. -6.
- [2] D.Chai and F.Hock, "locating and decoding EAN-13 barcodes from images captured by digital cameras". Int. conf. on Information, Communications and Signal Processing, 1595-1599, 2005.
- [3] S.Wachenfeld, S.Terlunen and X.Jiang, "robust recognition of 1-D barcodes using camera phones". Pattern recognition. ICPR 2008.
- [4] Blayvas.I, Bruckstein.A and Kimmel.R. "efficient computation of adaptive threshold surfaces for image binarization". Computer vision and pattern recognition, proceedings of the 2001 IEEE Computer Society Conference. 737-742, 2001.
- [5] H.S.Neoh and A.Hazanchuk, "adaptive edge detection for real-time video processing using FPGAs".
- [6] <http://en.wikipedia.org/wiki/EAN13>.