BAR CODE READING FROM IMAGES CAPTURED BY CAMERA PHONES

Kongqiao Wang, Yanming Zou, and Hao Wang
Nokia Research Center, China

ABSTRACT

Bar codes are being widely used in many fields for applications of great commercial value. By encoding a series of characters or symbols, bar codes are able to both carry explicit information and a database key. Nowadays, The availability of imaging phones provides people a mobile platform for decoding bar code rather than the use of the conventional scanner which is lack of mobility.

However, the short-distance capture of bar codes using an imaging phone inevitably makes bar code images blurred, meanwhile, these images are contaminated heavily with noises. Hence, it is a challenge for automatic bar code reading by imaging phones in such applications.

In this paper, research effort on the algorithms of bar code reading by real NOKIA imaging phone products is proposed and EAN-13, a widely used 1-D bar code standard, is taken as an example to show the efficiency of the method. The method, of course, can be extended to other bar code standards without much effort. A wavelet-based bar code area location and knowledge-based bar code character segmentation scheme is applied to extract bar code characters under poor image quality of real conditions. Then the waveforms of the 12 marked divisions are input to the decoding engine, which is called statistical recognition block, and final decoding decision is made. Training of the statistical classifiers is based on the modified GLVQ (Generalised Learning Vector Quantization) method and the initial feature extraction is based on LDA (Linear Discriminant Analysis). Training samples are from the database contains over 1,100 bar code images taken by an imaging phone and the sample set is extended by manually shifting (distortion) of the original samples to cover more possibilities of occurrence.

Nearly 300 EAN-13 bar code images taken by imaging phone (NOKIA 3650) without micro-lens are tested to prove the effectiveness of the proposed method. The entire symbol recognition rate is 85.62%, which is desirable for the first kick-off of the attempt to implement bar code reading applications in the camera phone products. Bar code images taken with micro-lens or optical zoom functionality are also tested and the entire symbol recognition rate is nearly hundred percent.

Index Terms -Bar code reading, Camera/Imaging phone, Statistical recognition, GLVQ, LDA.

1. INTRODUCTION

The optics of an imaging phone is designed for normal scene capture in which case the photographic distance is not less than 50cm. However, for the purpose of bar code reading applications, it is inevitable to take the pictures of bar codes within a very short distance, normally less than 10cm. In such cases, the bar code images are often very blurred and damaged by noises, shadows, and geometrical distortion. Traditional decoding techniques based on 2^{nd} derivatives [1], peak locations [2], selective sampling [3] and EM algorithm [4] may not work well because they were designed mainly for the sensors such as scanners which have much higher resolution than the phone cameras. In other words, it is very difficult to robustly extract accurate features such as edges and peaks of the bars and spaces from the bar code images taken by a camera phone.

This paper proposed a statistical method which can analysis and learn the natures of the contaminated bar code symbols so that the trained classifiers are able to discriminate different code patterns by extracting their statistical features. In this way, it is not necessary to rely on the accurate extraction of the edge or peak locations of bar code waveforms, which is even hardly available.

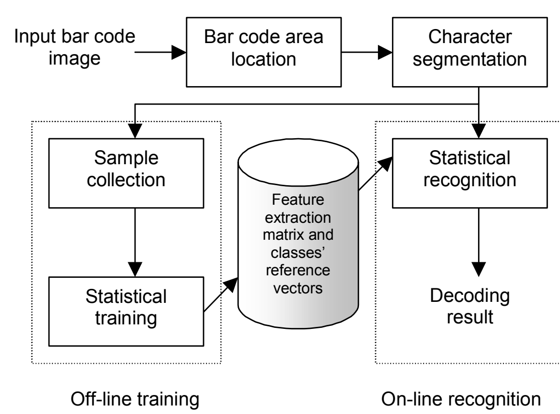


Figure 1: Diagram of the bar code decoding system.

The proposed scheme is described in Figure 1. First, the bar code area in the image is located using a wavelet-based method. Then a rule-based segmentation algorithm is applied to extract bar code characters from the original waveform. The marked divisions, namely, the waveforms of the bar code characters, are input to the decoding block.

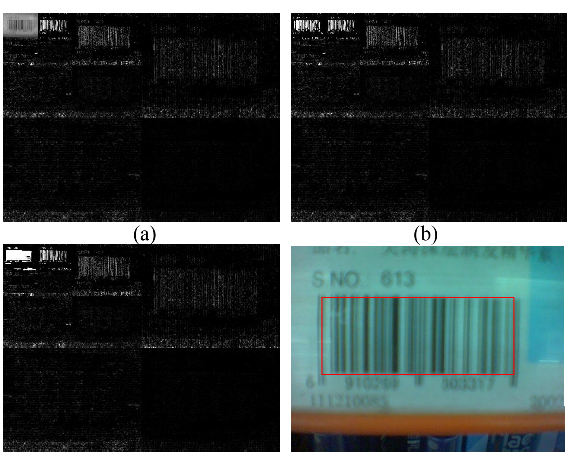
After post- processing, the final code is output. In the statistical training line, samples (segmented waveforms of characters) of each code pattern are input to the training block and supervised learning is performed. The output of the training block is a feature extraction matrix and classes' reference vectors, which are used in the on-line statistical recognition procedure. To keep generality, a widely used 1D bar code standard, EAN-13, is chosen for our experiment.

This paper is organized as follows. Section II introduces the methods of bar code area location and character segmentation. Statistical recognition and training algorithms are explained in section III. Section IV simply describes the method of data collection. Section V shows the experimental result and finally draws the conclusion.

2. BAR CODE LOCATION AND CHARACTER SEGMENTATION

2.1 Wavelet-Based Bar Code Area Location

A bar code is a pattern of parallel adjacent bars and spaces, which are aligned horizontally. Therefore the barcode area should be obviously dominated with vertical textures. In the different subbands of the wavelet image, the coefficients of *HL* subbands from the barcode area should be bigger than the ones of *LH* subbands or *HH* subbands spaced at the corresponding area. Hence, criteria defined based on the energy of a coefficient pyramid tree which emphasizes the vertical textures is applied for barcode area location.



(a) (b) (c) (d)

Figure 2: Wavelet-based bar code area location. The 3-level wavelet image is shown in (a). (b) gives the detected vertical textures in LL3 subband. The morphological and label process result is shown in (c). (d) is the location.

Figure 2b shows that most of coefficients in the barcode area meet the criteria. Regarding of the limitation in computation of mobile phones, Daubechies 5/3 symmetric filters [5] are selected for the image wavelet decomposition, which is simple and fast.

After marking the vertical texture region in the wavelet image, post processing is needed to determine the border of the bar code area. A morphological processing, namely, opening operation is applied to connect the vertical texture regions. Finally, an object clustering labeling process [6] is used to compute the size of all the object areas in the morphological processing result. Empirically, the maximum labeled area is the barcode area.

2.2 Knowledge-Based Bar Code Character Segmentation

Character segmentation is performed according to the structure of specific code standard, whereas the method can be generalized to most 1D bar codes. EAN-13 is one of the popular international barcode standards [7]. It includes 13 characters, in which the first character is the check digit, and the last character is the induced digit. Each character has seven modules, which is composed of two bars and two spaces. As shown in Figure 3, a typical EAN-13 barcode is composed of 95 modules:

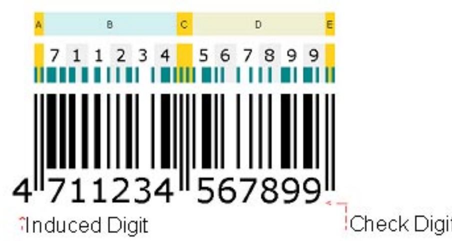


Figure 3: EAN-13 structure. The 13 digits are aligned from the right to the left.
A: Left-hand guard bar (3modules, the structure is bar-space-bar);
B: Left-hand six characters of code (7modules per character, the structure is space-bar-space-bar);
C: Center bar (5modules, the structure is space-bar-space-bar-space);
D: Right-hand six characters of code (similar to B);
E: Right-hand guard bar (similar to A).

The knowledge-based segmentation scheme is implemented in the following steps:

(1) The 2^{nd} derivative of the bar code waveform is calculated while the zero-crosses (ZC2s) are marked;

(2) The left-hand guard bar A and the right-hand guard bar E are located based on the bar-space-bar structure thus the start and end borders of the characters are determined and the average module width is calculated as reference;

(3) The center bar is segmented, where combinations of left/right guard bars and center bar are permitted;

(4) An iterative process is performed to segment characters of B and D respectively. The detail criteria and implementation can be referred to [9].

Figure 4 shows an example of bar code character segmentation result, which might keep some uncertainties that will be finally verified in the following recognition stage.


(a) (b)

Figure 4: Two examples of bar code character segmentation with only one candidate in (a) and two candidates in (b). The recognition stage will make decision which candidate is the correct one by the checking rule.

3. STATISTICAL RECOGNITION AND TRAINING ALGORITHM

3.1 Statistical Recognition

Figure 5 gives out the structure of the statistical recognition block. As mentioned before, the output of the character segmentation contains 12 one-dimension vectors from a one-dimension waveform. They are just the 12 input vectors of the statistical recognition block, corresponding to the characters $a_i, i = 1,2,\cdots,12$. Each input vector goes through the three steps: preprocessing for statistical recognition, feature extraction, and nearest neighbor classification. Combining all the outputs from the three classifiers, the post processing part will gives out the final decoding result.

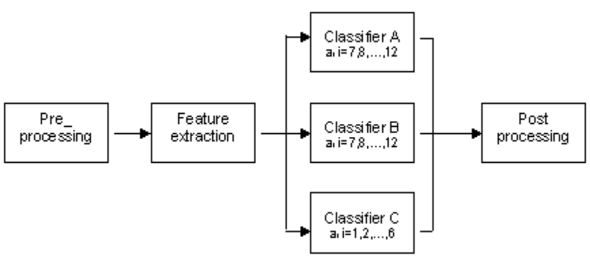


Figure 5: Structure of statistical recognition engine.

3.2 Preprocessing

The input patterns cannot be used for feature extraction and recognition directly, since the length of vectors is not constant, and the values of the vector components are too random for recognition. Therefore normalization is needed.

The following three normalization tasks are done in this preprocessing stage:

1) The length of each basic input vector, i.e. each marked part in the waveform, is normalized to 49 (7 points/module). In order to perform further processing, the extended 7 points nearest to the basic vector on both sides are remained. Thus the total length of the normalized vector is 63.

2) The sum of the components is normalized to 0.

3) The sum of the squared components is normalized to 1.

3.3 Feature Extraction and Nearest Neighbor Classification

Here, the final features are extracted from an input vector $\mathbf{y}$ with an matrix Ψ, which can be expressed as:

$$\mathbf{x} = \Psi^T \mathbf{y}. \qquad (1)$$

Although there are three different pattern sets, only one extractor (i.e. one feature extraction matrix) is used. This makes it sensible to sum the distances coming from different classifiers. The length of final feature vector is 16 in the current algorithm.

The distance between a normalized input vector and a reference vector is defined as:

$$d_k^j(\mathbf{x}) = d\left(\mathbf{x}, \mathbf{r}_k^j\right) = \left(\mathbf{x} - \mathbf{r}_k^j\right)^T \left(\mathbf{x} - \mathbf{r}_k^j\right), \qquad (2)$$

where $\mathbf{r}_k^j$ is the kth reference vector of the jth class.

Therefore the distance between an input vector and a class can be defined as the smallest distance among all its references, which can be expressed as:

$$d^j(\mathbf{x}) = \min\left\{d_k^j \middle| k = 1,\cdots,K\right\}. \qquad (3)$$

The output of a classifier is a candidate list with the distances between them and the input vector. The list contains all the patterns in the pattern set corresponded to it and the order depends on their distances, from the smallest to the largest.

3.4 Postprocessing

This part calculates the summed distance of all possible combinations of the character values, which satisfied the two conditions:

1) The constraint in EAN-13

2) There are at least 11 characters taking their first choice in the combination.

The combination with the smallest summed distance is given out as the recognition result.

3.5 Training Algorithms

The aim of this training part is to get the best feature extraction matrix and classes' reference vectors. Actually, the training part implements a modified generalized learning vector quantization (GLVQ) method, which can optimize the feature extraction matrix and reference vectors simultaneously. Here, the linear discriminant analysis (LDA) is used to get the initial feature extraction matrix, and K-means clustering method is used to get the initial reference vectors for the optimization method.

3.6 Modified GLVQ Method

LVQ is a supervised learning method to get the optimized reference vectors from amount of training samples. GLVQ in the reference provides a general method to control the learning process of the original method [8]. The modified version proposed in this paper further generalized the GLVQ method, making it possible to deal with the cases where one class can have several reference vectors and to optimize the feature extraction matrix in the optimization process of the reference vectors.

The proximity of an input vector $\mathbf{x}$ to its own class, can be defined as

$$\mu(\mathbf{x}) = \frac{d^m(\mathbf{x}) - d^j(\mathbf{x})}{d^m(\mathbf{x}) + d^j(\mathbf{x})}, \tag{4}$$

where d^m is the distance between an input vector $\mathbf{x}$ and the nearest reference vector $\mathbf{r}_i^m$ of the class to which $\mathbf{x}$ belongs, and d^j is the distance between $\mathbf{x}$ and the nearest reference vector $\mathbf{r}_k^j$ of the classes to which $\mathbf{x}$ does not belong. It is obvious that the smaller μ is, the higher confidence $\mathbf{x}$ belongs to class *m* we have.

So our modified GLVQ method can be formalized as a minimization problem of an evaluation function Q:

$$Q = \sum_{i=0}^{S} f(\mu_i) \cdot \tag{5}$$

Here S is the number of training samples and $f(\cdot)$ is monotonously increasing function. The real definition of $f(\cdot)$ is not needed; only its derivative is given out:

$$\frac{\partial f}{\partial \mu} = F(\mu,t)(1 - F(\mu,t)), \tag{6}$$

$$F(\mu,t) = \frac{1}{1 + e^{-\mu(\mathbf{x})t}} \cdot \tag{7}$$

Thus, one step of the modified GLVQ algorithm for an input vector can be described as

$$\mathbf{r}_{i,t+1}^m = \mathbf{r}_{i,t}^m + \alpha \cdot \frac{\partial f}{\partial \mu} \frac{d^j}{\left(d^m + d^j\right)^2}\left(\mathbf{x} - \mathbf{r}_{i,t}^m\right), \tag{8}$$

$$\mathbf{r}_{k,t+1}^j = \mathbf{r}_{k,t}^j - \alpha \cdot \frac{\partial f}{\partial \mu} \frac{d^m}{\left(d^m + d^j\right)^2}\left(\mathbf{x} - \mathbf{r}_{k,t}^j\right), \tag{9}$$

$$\mathbf{\Psi}_{t+1} = \mathbf{\Psi}_t - \beta \cdot \mathbf{y}^T \cdot \frac{\partial f}{\partial \mu} \frac{d^j \cdot \left(\mathbf{x} - \mathbf{r}_{i,t}^m\right) - d^m \cdot \left(\mathbf{x} - \mathbf{r}_{k,t}^j\right)}{\left(d^m + d^j\right)^2} \cdot \tag{10}$$

3.7 Feature Extraction Matrix and Reference Vector Initialization

Experiments show that the initial state is very important to the performance of the final training results of our modified GLVQ. Therefore, lots of efforts are paid to the process of generating the initial state of the optimization.

LDA is a method to get the dimension compression matrix Φ' for feature extraction. Here the matrix Φ' is defined as the first *n*' columns of eigenvector matrix Φ, which satisfies the equation

$$S_b \Phi = S_\omega \Phi \Lambda, \tag{11}$$

where Λ is the eigenvalue matrix having eigenvalues $\lambda_i \ (\lambda_i \geq \lambda_i \geq \ldots \geq \lambda_n)$ as its diagonal elements, S_b and S_ω are the between-class covariance matrix and within-class covariance matrix respectively.

Normally, the columns in the matrix Φ' are not orthonormal to each other. To make a sensible distance definition, the orthonormal basis for the range of Φ' is gotten as the matrix Φ''.

With Φ'', the within-class covariance matrix in the compressed feature space becomes

$$S''_\omega = \Phi''^T S_\omega \Phi'' \,. \tag{12}$$

The weighted Euclidian distance is used in this method. Thus the initial feature extraction matrix is

$$\Psi = \Phi'' \cdot \Phi''' \cdot \left(\sqrt{\Lambda'''}\right)^{-1}. \tag{13}$$

where Λ''' and Φ''' are the eigenvalue and eigenvector matrix of S''_ω respectively.

And the distance between the input pattern and a reference vector is just as the formula (1) and (2).

Here, a clustering is used to generate the initial reference vectors for the modified GLVQ method. Assumed N reference vectors are used for each class, the initial references for clustering $\left\{\mathbf{r}_{n,0}^m \middle| n = 1, \cdots, N\right\}$ are selected with the min-max method, which can be describe as follows:

$\mathbf{r}_{1,0}^m = \mathbf{x}_i$, for *n*=1; *i* is selected randomly;

$\mathbf{r}_{n,0}^m = \arg \max_{\mathbf{x}_j, j=1,\cdots,N-1} \left(\min_{k=1,\cdots,n-1} d\left(\mathbf{x}_j, \mathbf{r}_{k,0}^m\right)\right)$, for *n*=2:N;

Here $\mathbf{x}_i$ is selected randomly from the training samples of the *m*th class.

The iteration process is the standard K-means clustering, which means:

$$\mathbf{r}_{n,t+1}^m = \frac{1}{\left\|C_{n,t}\right\|} \sum_{\mathbf{x}_j \in C_{n,t}} \mathbf{x}_j, \tag{14}$$

where $C_{n,t} = \left\{ \mathbf{x}_j \middle| \mathbf{r}_{n,t}^m = \arg \min_{k=1,\cdots,N-1} d\left(\mathbf{x}_j, \mathbf{r}_{k,t}^m\right)\right\} \cdot \quad (15)$

4. DATA SAMPLES

The waveform of a bar code is obtained by sampling along 8 parallel horizontal scan lines and a medium-averaging method is applied to reduce random noises. After character segmentation, the original waveform is divided into 12 segments (so called vectors) corresponding to the first 12 characters of the bar code. In order to cover the statistical possibilities of existence, more than 1,000 bar code images taken by NOKIA 3650 model are collected as our training set. A semi-automatic data collection framework is used to extract training samples. EAN-13 defines 30 code patterns, and in the original database, there are more than 300 samples collected for each pattern.

However, the samples in the original database still seem insufficient. Our way to solve this problem is to generate some new training samples by distorting the directly collected ones.

From analysis, it is found that our training sample set does not give out enough coverage for the possibilities that the boundaries of input vectors generated by the character segmentation may be correct but to some extent inaccurate. Thus the first kind of distortion is to shift each division point randomly in the waveform. Here, the shift distance is restricted to 1 or 2 pixels, which is about the same level of the position accuracy of the real image processing part. One distorted sample is generated for each sample we collected.

In the training process, the original and distorted samples are mixed together.

5. EXPERIMENTAL RESULT AND CONCLUSION

The current testing database contains 292 bar code images taken by NOKIA 3650. Among the 292 images, 24 of them are manually made bar codes using the bar code creation function in the data collection framework. The remaining 268 images are all from natural bar codes printed on the surfaces of products. Situations of picture taking vary from normal laboratory environment with good illumination to the real supermarket environment with poor lightening and noises. Since the sizes of printed bar codes are not fixed, the degrees of blurring of the images are also variant.

The current bar code reading result is shown in Table 1. The errors come from two factors: one is from the code segmentation stage as the characters might not be extracted correctly in some cases; the other type of error comes from the recognition engine, namely, the statistical classifier.

Table 1 –Bar code decoding result. All the testing images are taken by NOKIA camera phone without micro-lens or optical zooming.

Total number of testing samples	Number of correction	Number of wrong	Correct *SYMBOL/STRING* recognition rate
292	250	42	85.62%
	Sementation correction	Sementation error	Segmentation correction rate
	275	17	94.18%

All the testing images are taken without micro-lens, which intends to imitate the real application situation for bar code reading in the scenario environment. The string recognition rate is 85.62%, which means there are over four fifths of the testing bar code symbols that can be correctly recognized (for each symbol, the 13 digits are all correct).

Supposing that the symbol recognition procedures are independent, it is deduced that if two images are taken for one bar code symbol, then the probability that at least one image can be correctly recognized is:

$1-(1-p)^2= p(2-p)$=97.93%, where p stands for the string recognition rate. It means that in real application cases, the printed bar code is scanned by the camera and consequently a series of continuous frames of the bar code are input to the recognition engine. Thus among the frames, the probability of at least one frame can be correctly recognized is much higher than the isolated string recognition rate.

We also tested the bar code images taken with micro-lens or optical zoom functions. An automatic judgement of out-of-focus is performed during the bar code character segmentation stage and only the qualified images with enough clear segment marks are input to the statistical recognition engine. Therefore, the string recognition rate is nearly hundred percent which is very desirable for practical usage.

Further research will focus on two factors to improve the performance of bar code reading. The first one is to combine the character segmentation (image processing part) and the recognition part more efficiently, which means the two stages should not be independently proceeded as they might have effects to each other. Another point is to obtain more appropriate deformable model of the very-short-distance performance of the phone camera, which will be very helpful for the sample distortion and feature restoration in statistical training process.

ACKNOWLEDGMENT

We would like to thank Mr. Yrjanainen Jukka and Mr. Vehvilainen Markku, who initialized the project and provided helpful materials and supports.

REFERENCES

1. Juha Uusikyla, 2000, “Barcode recognition in digital image”, *Master of science thesis*, TUT.
2. E. Joseph and T. Pavlidis, 1998, “Bar code waveform recognition using peak locations”, *IEEE Trans. on PAMI*, v16, n6, pp. 630-640.
3. Stephen J. Shellhammer, et al., 1999, “Novel signal-processing techniques in barcode scanning”, *IEEE Robotics & Automation Magazine*, pp. 57-65.
4. W. Turin and R. A. Boie, 1998, “Bar code recovery via EM algorithm”, *IEEE Trans. on PAMI*, v46, n2, pp. 354-363.
5. I. Daubechies, 1988, “Orthonormal bases of compactly supported wavelets”,. *Commun., Pure Appl., Math*. Vol.41, pp909-996.
6. M. W. Mak, 1994, “A lip-tracking system based on morphological processing and block matching techniques”, *Signal Processing: Image Communication*, Vol.6, pp. 335-348.
7. http://www.export911.com/e911/coding/ean13.htm#xEAN13Sym
8. T. Fukumoto, T. Wakabayashi, F. Kimura and Y. Miyake, 2000, “Accuracy improvement of handwritten character recognition by GLVQ*”, IWFHR VII*, pp.271-280.
9. K. Q. Wang, 2003, “Barcode character segmentation for mobile cameras”, *technical report of NOKIA*.