

Kongqiao Wang, Yanming Zou, Hao Wang, 1D bar code reading on camera phones, International Journal of Image and Graphics, vol. 7, no. 3, pp. 529-550, 2007, World Scientific Publishing, ISSN 0219-4678.

© 2007 World Scientific Publishing Company

Reprinted with permission.

<http://www.worldscinet.com/ijig/ijig.shtml>

1D BAR CODE READING ON CAMERA PHONES

KONGQIAO WANG*, YANMING ZOU[†] and HAO WANG[‡]

*Visual Interaction Group, Nokia System Research Center Beijing
Nokia House 1, No.11, Hepingli Dongjie, 100013, Beijing, China*

**kongqiao.wang@nokia.com*

†yanming.zou@nokia.com

‡hao.wi.wang@nokia.com

Received 5 October 2006

Revised 25 January 2007

Accepted 15 February 2007

The availability of camera phones provides people with a mobile platform for decoding bar codes, whereas conventional scanners lack mobility. However, using a normal camera phone in such applications is challenging due to the out-of-focus problem. In this paper, we present the research effort on the bar code reading algorithms using a VGA camera phone, NOKIA 7650. EAN-13, a widely used 1D bar code standard, is taken as an example to show the efficiency of the method. A wavelet-based bar code region location and knowledge-based bar code segmentation scheme is applied to extract bar code characters from poor-quality images. All the segmented bar code characters are input to the recognition engine, and based on the recognition distance, the bar code character string with the smallest total distance is output as the final recognition result of the bar code. In order to train an efficient recognition engine, the modified Generalized Learning Vector Quantization (GLVQ) method is designed for optimizing a feature extraction matrix and the class reference vectors. 19 584 samples segmented from more than 1000 bar code images captured by NOKIA 7650 are involved in the training process. Testing on 292 bar code images taken by the same phone, the correct recognition rate of the entire bar code set reaches 85.62%. We are confident that auto focus or macro modes on camera phones will bring the presented method into real world mobile use.

Keywords: Bar code reading; camera phone; wavelet transform; morphological processing; GLVQ; LDA.

1. Introduction

A conventional 1D bar code is composed of a serial number coded in black and white bars, which encodes the identifying data for the corresponding goods being produced and distributed around us daily. Now there are quite a lot of international 1D bar code standards, e.g. EAN-13, ISBNCode, UPC, Code128, etc. used in department stores, factories, mail distribution centers, etc.

After laser scanning technology came into practical use, many efforts^{1–9} were paid to bar code reading through laser scanners. Till now, commercial laser bar code readers are broadly used in many places, e.g. supermarkets. However, the biggest

disadvantage of the laser bar code readers is their lack of mobility, because most of the time, they can only be used together with the corresponding POS machines.

With the availability of camera devices, attention is being paid to developing bar code reading technology on these devices.^{10,11} With a bar code reader integrated in the camera phone, end-user will not only benefit from bar code information read by the phone, but the camera phone will also provide the full mobility. However, bar code images normally are prone to deterioration due to geometric distortion, noise, blurring, and so on. Even though geometric distortion can be controlled, and noise can be efficiently reduced through image preprocessing, image blurring is sometimes a factor influencing the performance of a bar code recognition system. It is proven that image blurring is usually inevitable in a camera-based picturing system, especially in the case of the camera that does not have auto focus or macro mode.

Although some high-end camera phones which integrate a high resolution and auto focus/macro mode have been launched into markets, the low-end camera phone user segment is still huge, especially, in emerging markets. Hence, in this paper, we aim to provide a general 1D bar code reading solution for camera devices ranging from low-end to high-end. As described in Fig. 1, the presented system is mainly composed of three parts, i.e. wavelet-based bar code localization, knowledge-based bar code segmentation, and statistical bar code recognition.

At the bar code localization stage, the bar code is localized by using the strong directional selection competence of the wavelet subbands spaced in high frequencies. Due to the blurring and geometric distortion of bar code image, the zero crossings of second derivative on bar code waveform cannot always be detected. Even if detected, not all the zero crossings can be localized accurately. The presented bar code segmentation scheme is able to robustly segment guard bars and characters of a bar code even when some of the zero crossings on the bar code waveform cannot

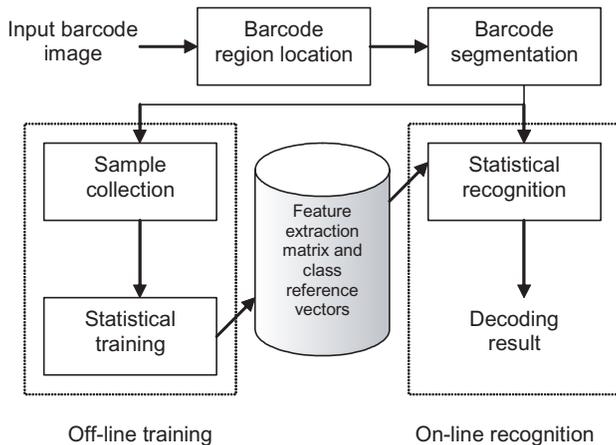


Fig. 1. Diagram of the bar code decoding system.

be detected. The border segmentation of some bar code characters might not be very accurate. However, this will not usually prevent the correct recognition of the characters, because, in the statistical recognition part, all the training character samples are extracted with the segmentation scheme, and extended one code module into both sides respectively before normalization. A modified GLVQ method is designed for optimizing a feature extraction matrix and the class reference vectors, which are used in the on-line bar code recognition procedure. The bar code segmentation and recognition is performed according to the structure of EAN-13 specific code standard, and the method can be easily generalized to almost all the other existing 1D bar code standards, e.g. Code 128, etc. .

The rest of the paper is organized as follows: The methods of bar code region location and sampling are given in Secs. 2 and 3, respectively. Section 4 introduces the knowledge-based bar code segmentation scheme, and the statistical recognition and training algorithms are described in Sec. 5. Finally, the experimental results and discussion are given in Sec. 6.

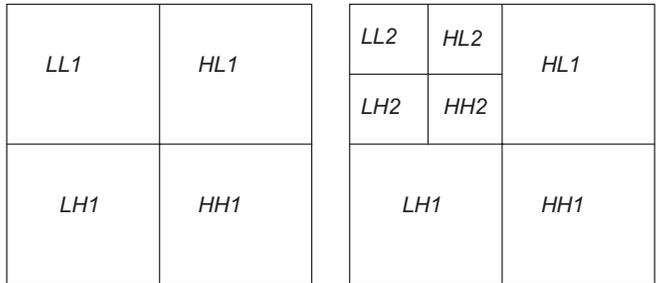
2. Bar Code Location

Any 1D bar code is composed of parallel and adjacent bars and spaces, which are aligned horizontally. Therefore the bar code region should be obviously dominated with vertical textures. Based on the knowledge, the wavelet-based method is presented in this section to locate the bar code region from a bar code image. However, even though a bar code has not been captured horizontally, the method can still work if the bar code direction can be correctly detected and rotated until it is horizontal.

2.1. Wavelet transform

Wavelet transform can provide a compact multiresolution representation of the image, and the wavelet subbands spaced in high frequencies have strong directional selection competence.¹² With the wavelet transform, an image is divided into four subbands (one low-frequency subband $LL1$ and three high-frequency subbands $LH1$, $HL1$ and $HH1$ as shown in Fig. 2(a)). To obtain the next coarser scale of wavelet coefficients, the subband $LL1$ can be further decomposed to get another four subbands at the coarser scale (one low-frequency subband $LL2$ and three high-frequency subbands $LH2$, $HL2$ and $HH2$ as shown in Fig. 2(b)). The low-frequency subband is one approximation of the original image, while the three high-frequency subband lists, i.e. LH list, HL list and HH list, reflect the horizontal, vertical and diagonal edges of the image respectively.

From the coefficients arrangement of the wavelet image, at the same spatial region of the image, each coefficient at a coarse scale corresponds to four neighboring coefficients at the next finer scale (refer to Fig. 3). The pyramid data structure built from the corresponding high-frequency coefficients is normally called a pyramid



(a) The one-level decomposition (b) The two-level decomposition

Fig. 2. The wavelet decomposition.

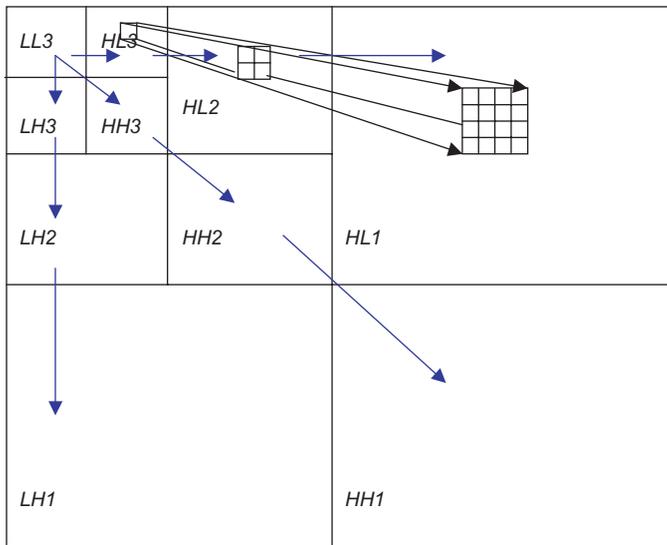


Fig. 3. The pyramid structure of the wavelet image.

tree. The relationships of wavelet coefficients can be used for image compression and image texture detection.

Other issues on wavelet transform to be considered are image edge extension and the selection of wavelet filters.

2.1.1. Image edge extension

Before the image is decomposed with wavelet transform, it is necessary to extend the edges of the image. To minimize the “ringing” artifacts of the image introduced by wavelet transform, the symmetric extension method is selected to extend the edges of the image before the wavelet transform on the image.

2.1.2. The selection of wavelet filters

Regarding the computing limitation of mobile devices, Daubechies 5/3-tap symmetric filters¹² are selected for the image wavelet decomposition. The coefficients of the filters are given in Table 1. The advantages of the selection are obvious, firstly all the filters have few taps, and secondly the filtering computation (convolution computation) can be implemented only by shift and plus operators. In addition to the advantages, the filters have good localization properties, their symmetry allows for simple edge treatments, and they produce good results empirically.

2.2. Bar code region location

Any 1D bar code is composed of parallel and adjacent bars and spaces, which are aligned horizontally. Therefore the bar code region should be obviously dominated with vertical textures. In the different subbands of the wavelet image, the coefficients of *HL* subbands in the bar code region would be much bigger than the ones of *LH* subbands or *HH* subbands spaced at the same region (see Fig. 4).

In order to describe the process of bar code location using the characteristics of high-frequency wavelet subbands, the following two definitions are given:

Definition 1. A **pyramid tree** is composed of one coefficient on the coarsest high-frequency subband and the corresponding coefficients from all the finer high-frequency subbands at the same spatial region. For example, Fig. 5 shows a three-level *HL* pyramid tree, which includes 21 coefficients from the *HL* list, i.e. one from *HL3*, four from *HL2*, and the rest from *HL1*.

Table 1. The coefficients of the 5/3-tap symmetric wavelet filters.

	-3	-2	-1	0	1	2	3
High Pass	0	0	-0.5	1	-0.5	0	0
Low Pass	0	-0.125	0.25	0.75	0.25	-0.125	0



(a) The original bar code image (b) The 3-level wavelet image

Fig. 4. The wavelet image of a bar code image after 3-level wavelet decomposition.

Definition 2. The energy of a pyramid tree is the amount of the absolute volumes of all the coefficients in the pyramid tree. In Fig. 5, it is assumed that the energy of the coefficient(s) on the i th level of the pyramid tree is defined as $E_{i,coef}$, $i = 1, 2, 3$, then the energy of the pyramid tree is $E_{coef} = E_{1,coef} + E_{2,coef} + E_{3,coef}$

From the energy point of view, in the same pyramid tree, there is energy similarity between different levels, in other words, if the energy at one level in the pyramid tree is big, the energy at the other levels in the same tree is big as well, and vice versa. From the frequency characteristics perspective, in a bar code region, the vertical textures are always dominant. Hence, the following criteria are defined for bar code region location:

- (i) $E_{HL1,coef} > E_{LH1,coef}$ and $E_{HL1,coef} > E_{HH1,coef}$
- (ii) $E_{HL1,coef} + E_{HL2,coef} > E_{LH1,coef} + E_{LH2,coef}$ and $E_{HL1,coef} + E_{HL2,coef} > E_{HH1,coef} + E_{HH2,coef}$
- (iii) $E_{HL1,coef} + E_{HL2,coef} + E_{HL3,coef} > E_{LH1,coef} + E_{LH2,coef} + E_{LH3,coef}$ and $E_{HL1,coef} + E_{HL2,coef} + E_{HL3,coef} > E_{HH1,coef} + E_{HH2,coef} + E_{HH3,coef}$

If we check the wavelet image on Fig. 4(b), as expected, most of coefficients in the bar code region meet these criteria.

In order to locate the bar code region, the postprocessing on the detected result above is necessary. The postprocessing process is composed of the morphological processing process and the object clustering labeling process.¹³

The purpose of the morphological processing here is to impose the closing operation on the detected result based on the energy criteria of wavelet pyramid trees for filling up the thin gulfs and small holes.

Closing operation:

$$X \bullet B = (X \oplus B) \ominus B, \tag{1}$$

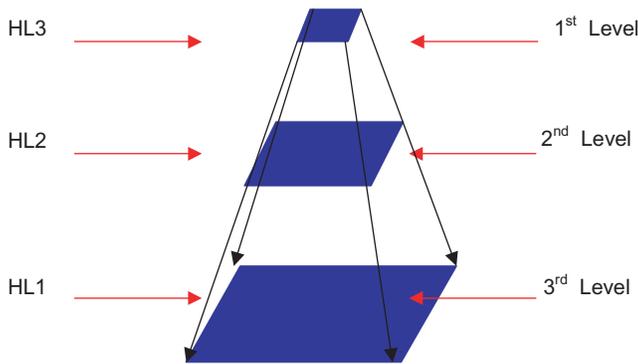


Fig. 5. A coefficient pyramid tree.

where X is the result to be processed and B is the morphological structure element. As all the bar code bars are aligned horizontally, their high-frequency coefficients, which meet the energy criteria above, are separated by the bar code spaces. In order to connect these coefficients of the bar code bars to facilitate the bar code location, the structure element B is defined as a horizontal vector, e.g. $[1111111]$. It should be noted that the length of B could be adjusted based on the size of the bar code to be processed. For a VGA bar code image, empirically, it is a good choice to define the length of B as 7 if the length of a horizontal bar code is about two third of the width of the whole image.

The object clustering labeling process¹³ is used to compute the size of all the separate object regions in the morphological processing result. Generally, the maximum labeled region is the bar code region in a bar code-dominant image. Figure 6 shows the bar code location result of the image in Fig. 4(a).

The bar code location method described above works for any 1D bar code standard if the bar codes are captured horizontally, i.e. the bars and spaces of the bar codes are aligned horizontally.

3. Bar Code Region Sampling

3.1. *The horizontal sampling data of a bar code*

In order to acquire the bar code information for decoding, a general way is to sample the data along a horizontal scan line through the bar code. If the bar code is not tilted, the bar code image is not heavily contaminated with noise, and there is no distortion in geometry, the sampling data along only one scan line may be enough to decode the bar code. But in practice, in order to acquire enough information on the bar code, it is usually necessary to use multiple scan lines for sampling data. The sampling data is highly correlated. Of course, due to the geometric distortion and/or the slope of the bar code, morphing some sampling data is very beneficial so that all the sampling data along different scan lines can be aligned.



Fig. 6. The bar code area located.

Based on the bar code location, we sample eight groups of data along eight scan lines for each bar code, as shown in Fig. 7(a). The eight groups of sampling data have very similar waveforms, as shown in Fig. 7(b).

In order to weaken the influence of noise, the direct idea of computing the target bar code sampling data is that all the sampling data except the maximum and minimum among them are averaged to get the target sampling data on each point of the sampling curve. It should be stressed here that the averaging process will blur the sampling data if all the sampling data along different scan lines are not aligned due to the geometric distortion and/or the slope of the bar code. However, because of the robust bar code character segmentation, the blurring will not be enough to make the whole recognition system invalid.

Assuming that $F_i(j)$ is the j th sampling point on the i th scan line, $i = 1, 2, \dots, 8$, and $j = 1, 2, \dots, L$ (L is the width of the bar code area), then the target sampling data on the j th point is:

$$F(j) = \frac{\left[\sum_{i=1}^8 F_i(j) \right] - \max - \min}{6}, \tag{2}$$

where $\max = \max\{F_i(j), i = 1, 2, \dots, 8\}$ and $\min = \min\{F_i(j), i = 1, 2, \dots, 8\}$. As shown in Fig. 7(b), the last curve is the target sampling data for the bar in Fig. 7.

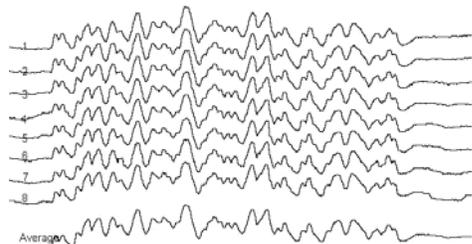
From the waveform profile, the target sampling data is rather similar to each of the eight original sampling data, but the target data has less noise.

3.2. The second derivative of bar code sampling data

On the curve of the bar code sampling data, each zero crossing of second derivative (ZC2) is corresponding to one edge from bar to space or space to bar. Hence, ZC2s are very important information to locate the edges of bar code bars and spaces, which are the base of segmenting the characters from the bar code.



(a) The eight scan lines



(b) The sampling data

Fig. 7. The horizontal sampling data of a bar code.

3.2.1. The second derivative of bar code sampling data

Assuming that a bar code sampling curve is $F(i)$, then the second derivative of the curve is:

$$F''(i) = \frac{\partial^2 F(i)}{\partial i^2}. \quad (3)$$

In discrete implementation, $F''(i) = F(i-1) + F(i+1) - 2F(i)$, $i = 1, 2, \dots, L$ (L is the width of the bar code area). Figure 8 shows the second derivative of the target curve in Fig. 7.

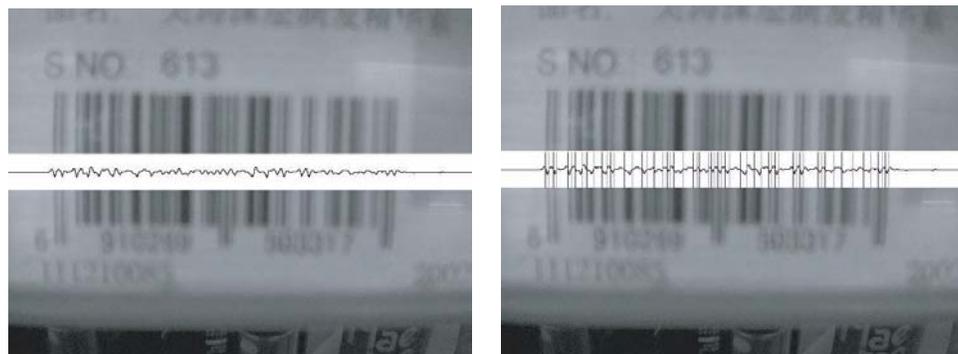
3.2.2. The computation of zero crossings of second derivative

On the second derivative curve $F''(i)$, a zero crossing of second derivative (ZC2) is at zero level, and the two nearest integer points at two sides have opposite signs. As shown in Fig. 9, between the i th and $(i+1)$ th integer points, there is one ZC2 on the curve. In order to facilitate the description of the following content, we define the property for each ZC2 (refer to Fig. 9) as:

$$\{i + \delta_i, L\text{-height}, R\text{-height}, BarToSpace\}, \quad (4)$$

where $i + \delta_i$ means the position of the ZC2, $L\text{-height}$ is the height of its left extremum, while $R\text{-height}$ is the height of the right extremum. $BarToSpace$ is a *BOOL* parameter, which defines the category of the ZC2, i.e. if the left side of the ZC2 is a bar and the right side is a space, the parameter is *TRUE*, otherwise, the left side is a space and the right side is a bar, it is *FALSE*.

It should be noted that $i + \delta_i$ might not be the actual position of the ZC2, because it is only the result of the interpolation between i and $i+1$, i.e. $i + \delta_i$ is the cross of the line and the dotted horizontal line in Fig. 9, but the actual ZC2 position locates at the cross of the sampling curve and the dotted horizontal line in Fig. 9. However, the difference will not be enough to have a big influence on the bar code segmentation and recognition.



(a) The second derivative

(b) The zero crossings

Fig. 8. The second derivative and its zero crossings for the target curve in Fig. 7.

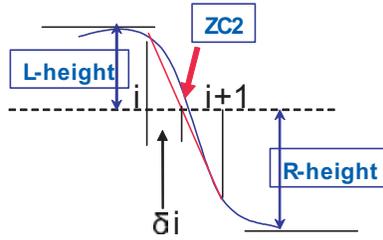


Fig. 9. Definition of ZC2.

Due to the influence of noise, there may be some noisy ZC2 points, as well as those real ZC2 points. Normally, a noisy ZC2 point has a small *L*-height volume and/or a small *R*-height volume. Based on this obvious characteristic, we can efficiently remove most the noisy ZC2 points. Figure 8(b) shows the computation result of the ZC2 points on the target curve in Fig. 7. The black vertical line means the *BarToSpace* of the ZC2 point is *FALSE*, i.e. at the left side of the ZC2 point is a space, and at the right side is a bar, while the gray vertical line means the *BarToSpace* of the ZC2 point is *TRUE*, i.e. at the left side of the ZC2 point is a bar, and at the right side is a space.

4. Knowledge-Based Bar Code Segmentation

The following bar code segmentation is performed according to the structure of EAN-13 specific code standard. However, the method can be generalized to almost all the other existing 1D bar code standards, e.g. Code 128, etc.

4.1. The brief of EAN-13 bar code standard

EAN-13 standard¹⁴ is one of the popular international bar code standards. 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 Fig. 10, a typical EAN-13 bar code

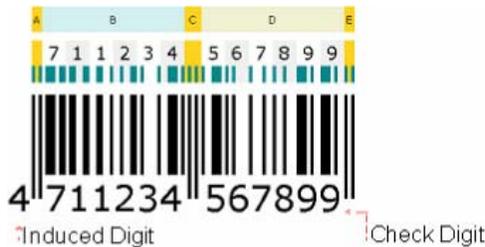


Fig. 10. The structure of an EAN-13 bar code.

is composed of 95 modules:

- (i) **A**: Left-hand guard bar (3 modules, the structure is bar-space-bar);
- (ii) **B**: Left-hand six characters of code (7 modules each character, the structure is space-bar-space-bar);
- (iii) **C**: Center bar (5 modules, the structure is space-bar-space-bar-space);
- (iv) **D**: Right-hand six characters of code (7 modules each character, the structure is bar-space-bar-space);
- (v) **E**: Right-hand guard bar (3 modules, the structure is bar-space-bar).

4.2. Guard bar segmentation of an EAN-13 bar code

Definition 3. The left border and the right border of each component in an EAN-13 bar code are marked with “*st*” and “*en*” respectively, for example, the left border and right border of the left-hand guard bar (*A*) are A_{st} and A_{en} respectively.

Definition 4. The δ scope of a point is the set of the left δ segment combining with the right δ segment centered to the point P , i.e. $(P - \delta, P + \delta)$.

4.2.1. The left-hand guard bar *A* and the right-hand guard bar *E* segmentation

As the left-hand and right-hand guard bar segmentations are quite similar (except in the opposite direction), in this section just the left-hand guard bar *A* segmentation is described.

Based on the located bar code region (refer to see Sec. 3.2), the bar code length can be estimated, for example, the estimated length is BC_Length_esti . Hence, the estimated width of one module is $Module_Length_esti = \frac{BC_Length_esti}{95}$, because each EAN-13 bar code has 95 modules.

Segmenting *A* from the bar code includes locating A_{st} and locating A_{en} .

As the left-hand guard bar is composed of three modules, and its structure is bar-space-bar (the width of each element is one module), at the A_{st} position, there should be a ZC2 point of which *BarToSpace* is *FALSE*. In order to separate a bar code from its surroundings, while the bar code is printed, a rather wide background space is normally present at the left side of A_{st} .

With the prior knowledge above, when we search A_{st} starting from the left border of the image, if the *BarToSpace* of a ZC2 point is *FALSE*, the ZC2 point can be considered as a candidate if the ZC2 point A_{st} also meets the following conditions:

- (i) The distance between the ZC2 point and its left ZC2 point or the left border of the image (at the case, the ZC2 point is the first one) is more than 5 times $Module_Length_esti$, e.g. $W_0 > 5 \times Module_Length_esti$.

- (ii) At the right side of the ZC2 point, if there exist three neighboring elements with the width of W_1 , W_2 and W_3 respectively (drawn in Fig. 11), and each of them is narrower than the element at the left side of the ZC2 point, i.e. W_0 .

The number of the A_{st} candidates searched may be more than one. All the candidates need to be saved. Of course, we also could delete some candidates which are not really obvious, for example, if there are too many ZC2 points, say more than ten ZC2 points at the left side of the candidate, the candidate A_{st} could be regarded as a false one.

For each A_{st} candidate, its corresponding A_{en} should locate in the δ scope of the position (assumed that $\delta = \frac{Module_Length_esti}{2}$, and the position is P) which is 3 times $Module_Length_esti$ far away from A_{st} on the right side (shown as in Fig. 11).

In the δ scope of P , if there exists a ZC2 point with $BarToSpace=TRUE$, then the ZC2 point is considered as A_{en} . Otherwise, the position P is directly defined as A_{en} .

Each segmented left-hand guard bar will create a pair together with one segmented right-hand guard bar. For example, using the search scheme above, if two potential left-hand guard bars and two potential right-hand guard bars are searched, then we will have four segmentation pairs (shown as in Fig. 12), in which there

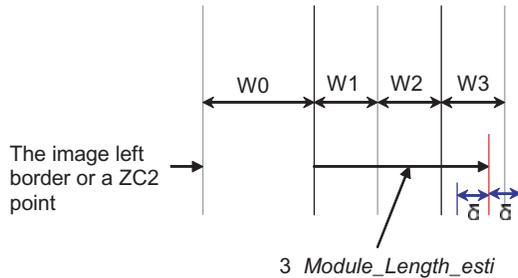


Fig. 11. Searching A_{st} : the black line means the ZC2 point with $BarToSpace=FALSE$, while the gray line means the ZC2 point with $BarToSpace=TRUE$.

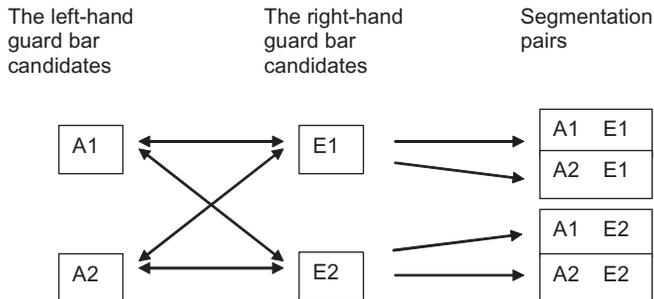


Fig. 12. The segmentation pairs of the A bar & E bar candidates.

may be only one right pair probably. But at the moment, we have to register all the candidate pairs for the further filtration.

4.2.2. The center guard bar (C) segmentation

The segmentation of the center guard bar (C) is based on the corresponding A-E pair.

For each A-E pair, the length between A_{en} and E_{st} can be easily calculated, for example AE_Length , the length of one module can be updated as $Module_Length_esti = \frac{AE_Length}{95-6} = \frac{AE_Length}{89}$. The reason for subtracting 6 here is because the length of A and E are 3 modules respectively.

Given an A-E pair, the segmentation of the corresponding center guard bar C can be described as follows (refer to Fig. 13):

- (i) $C_i (i = 0, 1, 2)$ is a reference mid-point of the center guard bar to be segmented. C_0 is the mid-point between A_{en} and E_{st} , and C_1 and C_2 are the left and right points which are $\frac{Module_Length_esti}{2}$ far away from C_0 respectively.
- (ii) Referring to the reference mid-point $C_i (i = 0, 1, 2)$, the left border C_{ist} and right border C_{ien} of the corresponding center guard bar are computed.
- (iii) Computing the distance $Dist(A_{en}, C_{ist})$ between A_{en} and C_{ist} , and the distance $Dist(E_{st}, C_{ien})$ between C_{ien} and E_{st} , we can easily compute the difference between the two distances, i.e. $Diff(C_i) = |Dist(A_{en}, C_{ist}) - Dist(E_{st}, C_{ien})|$.
- (iv) Considering the geometric distortion of the input bar code image, from the three segmentation candidates, only the candidate with the smallest distance difference $Diff(\cdot)$ is selected for center guard bar segmentation.

The computing steps of the left border C_{st} and right border of a center guard bar C_{en} are explained as follows:

- (i) Based on the given reference mid-point C_i of the center guard bar to be segmented, the reference left and right borders, C_{i_rs} and C_{i_re} , of the center guard bar are defined respectively. Both of them are 2.5 times $Module_Length_esti$ far away from C_i .
- (ii) In the δ scope of C_{i_rs} ($\delta = \frac{Module_Length_esti}{2}$), if there exists a ZC2 point with $BarToSpace = TRUE$, the ZC2 point is considered as the left border C_{st}

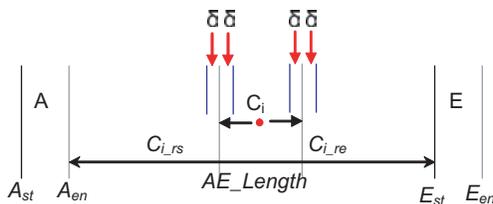


Fig. 13. The segmentation of the center guard bar (C).

of the center bar C. Otherwise, $C_{i_{rs}}$ is directly regarded as C_{st} . In the same way, C_{en} can be computed. The only difference is that, if C_{en} is a ZC2 point, the *BarToSpace* of the point should be *FALSE*.

4.3. EAN-13 bar code character segmentation

According to the definition of EAN-13 bar code standard, the following prior knowledge about EAN-13 bar code characters in the B code set can be concluded.

- (i) There are six characters aligned one by one between the left-hand guard bar A and the center guard bar C. Each character includes two bars and two spaces with the format of *bar-space-bar-space*.
- (ii) All the characters have the same width, and each is composed of seven modules. Assumed that the distance between A_{en} and C_{st} is AC_Length (refer to Fig. 14), the estimated width of one module can be further updated to $Module_Length_esti = \frac{AC_Length}{42}$.
- (iii) From left to right, the left border of the first character joins with the right border of the left-hand guard bar A, i.e. A_{en} , while the right border of the sixth character joins with the left border of the center guard bar C, i.e. C_{st}
- (iv) Due to the possible geometric distortion of the input bar code image, the width difference between the characters at the two ends of the B code set might not be small. But the width of any two adjacent characters would still be quite close.

Based on the prior knowledge above, an iterative process is presented to segment the characters in the B code set (refer to Fig. 14). The process can be easily applied to segment the characters in the D code set after minor modification based on the prior knowledge of the code set.

- (i) **Initializing.** The reference length for the first character (the character at the left end) is set as $Length_1 = 7 \times Module_Length_esti$. δ is starting from $Module_Length_esti$.
- (ii) **Looking for the left border of each character.** The left border of the first character ($i = 1$) actually joins with the right border of the left-hand

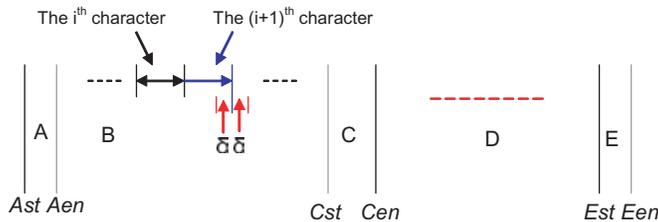


Fig. 14. The segmentation of EAN-13 bar code characters.

guard bar A, i.e. A_{en} . The left border of i th character is the right border of the $(i - 1)$ th character ($i = 2, 3, \dots, 6$).

- (iii) **Looking for the right border of each character.** From left to right, assuming the reference length of segmenting the i th character is $Length_i$. for example, the reference length of segmenting the first character is $Length_1$. If there is a ZC2 point with the $BarToSpace=TRUE$, which is in the δ scope of the point being $Length_i$ far away from the left border of the i th character ($i = 1, 2, \dots, 5$), the ZC2 point is considered as the right border of the i th character, otherwise, the point P is directly set as the right border of the i th character.
- (iv) **Computing the width difference of any two adjacent characters.** If the width difference of any two adjacent characters is not more than a given threshold, e.g. $\frac{\delta}{2}$, or $\delta < \frac{Module_Length_esti}{4}$, the segmenting process ends. Otherwise, after δ is decreased by $\frac{Module_Length_esti}{4}$, turn to (ii) for a new segmenting process.

5. Statistical Recognition and Training Alogirithm

5.1. Statistical recognition

The structure of the EAN-13 bar code character statistical recognition block is given in Fig. 15. Each segmentation result output from the segmentation system above includes 12 characters, represented by 12 vectors $a_i, (i = 1, 2, \dots, 12)$. The first six vectors $a_i, (i = 1, 2, \dots, 6)$ represent the right-hand six characters, while the remaining six vectors $a_i, (i = 7, 8, \dots, 12)$ represent the left-hand six characters. From the definition of EAN-13 standard, there are two decoding sets (Set A and Set B) for decoding the left-hand six characters of code, but there is only one decoding set (Set C) for decoding the right-hand six characters of code. Hence, three nearest neighbor classifiers are built for decoding each input vector output from the segmentation system, i.e. classifier A and B are used to decode $a_i, (i = 7, 8, \dots, 12)$,

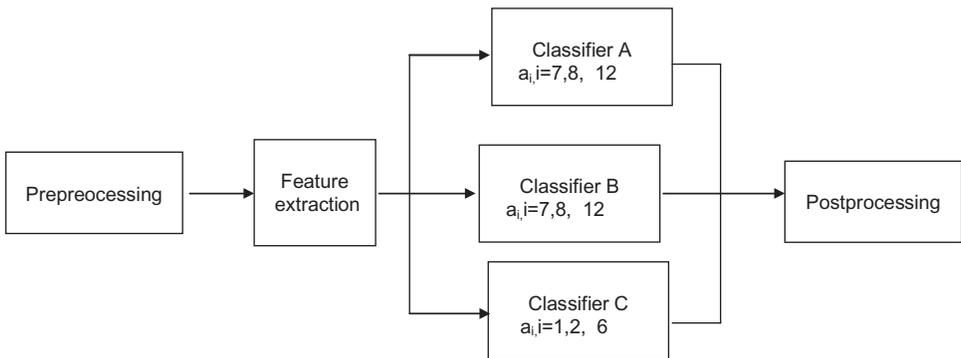


Fig. 15. The structure of the EAN-13 bar code character statistical recognition.

and classifier C is used to decode a_i , ($i = 1, 2, \dots, 6$). In the postprocessing part, all the outputs from the three classifiers are combined for giving out the final recognition result.

5.2. Preprocessing

As the length of the input vectors normally are not constant, and the values of vector components are quite random, all the input vectors need to be normalized before recognition. The normalization steps include:

- (i) Each input vector is extended by one module on both sides. If one module is normalized into 7 data points, then the length of an input vector will be normalized to be 63 data points.
- (ii) The average value of all the components in each input vector is normalized to be zero.
- (iii) The sum of the squared values of all the components is normalized to be 1.

5.3. Feature extraction and nearest neighbor classification

In the subspace defined with a $m \times n$ projection transformation matrix Ψ , an m -dimensional input vector \mathbf{y} is projected into the corresponding n -dimensional feature vector \mathbf{x} ($m > n$):

$$\mathbf{x} = \Psi^T \mathbf{y}. \quad (5)$$

The distance between an input vector \mathbf{x} and a reference vector class is defined as:

$$d_k^j(\mathbf{x}) = d(\mathbf{x}, \mathbf{r}_k^j) = (\mathbf{x} - \mathbf{r}_k^j)^T (\mathbf{x} - \mathbf{r}_k^j), \quad (6)$$

where \mathbf{r}_k^j is the k th reference vector, i.e. the center of the k th sub-class, of the j th class. In order to better represent a class, several sub-classes for each class are designed.

Therefore the distance $d^j(\mathbf{x})$ between the input vector \mathbf{x} and the j th class is defined as the smallest value of the list $\{d_k^j | k = 1, \dots, K\}$, i.e.

$$d^j(\mathbf{x}) = \min\{d_k^j | k = 1, \dots, K\}, \quad (7)$$

where K is the number of sub-classes in a class.

5.4. Postprocessing

Postprocessing is used to combine the recognition results from classifiers A, B and C, and give out the combination with the smallest summed distance as the final recognition result based on the code constraint in EAN-13 standard bar codes.

5.5. Training algorithm

The training algorithm aims to calculate the best feature extraction matrix and class reference vectors based on a modified generalized learning vector quantization (GLVQ) method.

LVQ is a supervised learning method for computing optimized reference vectors from amount of training samples. The GLVQ is a generalization of the LVQ,¹⁵ and is able to control the learning process of the method.¹⁶ As an implementation and a further generalization of the GLVQ method, the modified GLVQ method is to make it feasible to deal with the case in which there are several reference vectors designed for one class, and to optimize the feature extraction matrix simultaneously in the optimization process of the reference vectors.

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

$$\mu(\mathbf{x}) = \frac{d^m(\mathbf{x}) - d^j(\mathbf{x})}{d^m(\mathbf{x}) + d^j(\mathbf{x})}, \quad (8)$$

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. Obviously, the smaller μ is, the higher the confidence that \mathbf{x} belongs to the class m .

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

$$Q = \sum_{i=0}^S f(\mu_i), \quad (9)$$

where S is the number of training samples, and $f(\cdot)$ is a monotonously increasing function. In the practical computing process, actually the real definition of $f(\cdot)$ is not needed, it is enough to have its derivative:

$$\frac{\partial f}{\partial \mu} = F(\mu, t)(1 - F(\mu, t)), \quad (10)$$

$$F(\mu, t) = \frac{1}{1 + e^{-\mu(\mathbf{x})t}}. \quad (11)$$

Hence, the iteration process of the modified GLVQ method for an input vector \mathbf{x} can be described as

$$\mathbf{r}_{i,t+1}^m = \mathbf{r}_{i,t}^m + \alpha \cdot \frac{\partial f}{\partial \mu} \frac{d^j}{(d^m + d^j)^2} (\mathbf{x} - \mathbf{r}_{i,t}^m), \quad (12)$$

$$\mathbf{r}_{k,t+1}^j = \mathbf{r}_{k,t}^j - \alpha \cdot \frac{\partial f}{\partial \mu} \frac{d^m}{(d^m + d^j)^2} (\mathbf{x} - \mathbf{r}_{k,t}^j), \quad (13)$$

$$\Psi_{t+1} = \Psi_t - \beta \cdot \mathbf{y}^T \cdot \frac{\partial f}{\partial \mu} \frac{d^j \cdot (\mathbf{x} - \mathbf{r}_{i,t}^m) - d^m \cdot (\mathbf{x} - \mathbf{r}_{k,t}^j)}{(d^m + d^j)^2}. \quad (14)$$

The initial feature extraction matrix and reference vectors are given by the linear discriminant analysis (LDA)¹⁷ and the K -means clustering method respectively.

6. Experiments and Discussion

In the experiment, the projection transformation matrix Ψ of the subspace is defined as a 63×16 matrix, which projects all the input vectors from 63 dimensions to 16 dimensions before classification (refer to Eq. (5), $m = 634$ and $n = 16$). Besides, one class is defined for each code pattern of the three decoding sets (Set A, B and C. Each set includes ten code patterns.), as such, there are altogether 30 classes defined for the classification of all the input vectors. In order to better represent a class, four sub-classes are further designed. Referring to Eq. (7), j is from 1 to 30, and k is from 1 to 4.

6.1. Collection of bar code character samples for training

In order to build the three classifiers (refer to Fig. 15), more than 1000 EAN-13 bar code images were taken by NOKIA 7650. The rules of capturing bar code images satisfy:

- (i) Physical bar codes are placed as “in front of” the camera as possible.
- (ii) Physical bar codes are placed in the center of the camera view, and the length of the bar code region is adjusted to about two third of the width of the whole image, so as to avoid any bigger geometric distortion in the bar code region.
- (iii) The physical size of a bar code should not be shorter than 3 cm in length, and the capturing distance of the camera should be about 5 cm, so as to allocate enough pixels for each bar code character.

Figure 16 illustrates the above considerations, assuming that the capture angle of a NOKIA 7650 camera is 50° . If the length of the bar code region is adjusted to two thirds of the width of the whole image, the camera phone should be around 4.8 cm from the physical bar code while capturing. According to the resolution of a

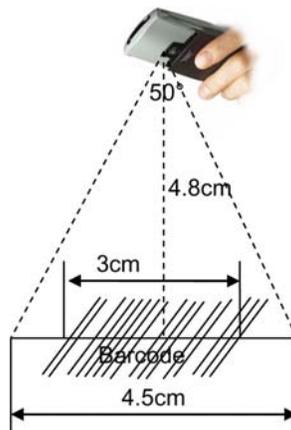


Fig. 16. The relation between bar code size and the capture distance of camera.

VGA camera, the width of the whole image is 640 pixels, so the length of the bar code region is about 427 pixels. As each EAN-13 bar code includes 95 module, each module is about 4.5 pixels in width.

By averaging the eight sampling scan lines across any captured bar code, at least 12 samples are extracted as training input vectors based on the bar code segmentation system followed by the normalization of the input vectors. As such, we get a total of 19584 training samples as the training database (5151 for Set A, 4641 for Set B, and 9792 for Set C respectively). As there are ten code patterns defined in each of the three code pattern sets of EAN-13 standard, on average, there are about 515 training samples for each code pattern in Set A, about 464 training samples for each code pattern in Set B, and about 979 training samples for each code pattern in Set C.

6.2. Performance testing

We captured 292 EAN-13 bar code images with NOKIA 7650 as the testbed. 24 out of the 292 bar code images are captured from the EAN-13 bar codes made by ourselves, and the remaining bar code images are all captured from surfaces of various products. The bar code capture conditions range from normal laboratory environment with rather good illumination to real supermarket environment with poor lighting and noise.

The testing results are given in Table 2. The mistakes mainly come from two aspects. One is from the bar code segmentation stage, in which a 94.18% correct segmentation rate is achieved. The other aspect is due to the statistical classifiers, Classifier A, B and C, at the recognition stage.

As the focal length of a normal camera phone without auto focus or macro modes, for example a NOKIA 7650, is about 30–50 cm, image blurring will be inevitable while a physical bar code is captured by the camera phone at 5 cm capturing distance. The blurring may result in the absence of some real zero crossings of second derivative bar code curve and/or shift of the detected zero crossings of second derivative waveform of a bar code. However, in the bar code segmentation and recognition system, the issue is fully considered. The border decision of either bar code character or guard bar is not always based on the computed zero crossings

Table 2. Bar code decoding results on the testbed.

Total No. of testing samples	No. of correct bar code segmentation	No. of wrong bar code segmentation	Correct bar code segmentation rate
292	275	17	94.18%
	No. of correct bar code recognition	No. of wrong bar code recognition	Correct bar code recognition rate
	250	42	85.62

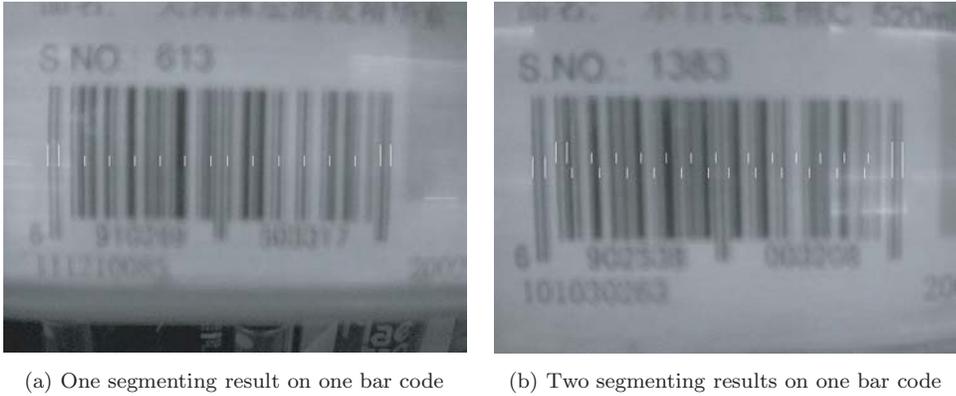


Fig. 17. Two examples of EAN-13 bar code segmentation.

of second derivative waveform. In this way, we allow multiple segmentation options for one bar code (see Fig. 17).

All the segmenting character results on one bar code output from the segmentation system are directly imported to the successive statistic recognition system, in which the recognition is not based on the accuracy of the segmenting borders of character. If multiple segmentation options are detected for a bar code at the segmentation stage, all the options are recognized and the corresponding recognition distances are compared. Only the character string (12 digits) with the smallest distance amount is output as the final recognition result of the bar code.

In the recognition system, only if all the 12 characters involved in one bar code are recognized correctly, the bar code can be considered as being recognized correctly. On the 292 testing bar codes, 250 out of them are recognized correctly, i.e. the correct recognition rate is 85.62%.

Supposing that the bar code recognition procedure is independent, it can be deduced that two attempts for recognizing one bar code will bring the probability of the correct recognition:

$$1 - (1 - \rho)^2 = \rho(2 - \rho) = 97.93\%, \tag{15}$$

where $\rho = 85.62\%$ stands for the one-shoot recognition rate. The scheme on how to select the final code string from the multiple-shoot recognition is still based on their recognition distances. According to our experiments, the correct recognition distance on a bar code is obviously shorter than that of any wrong recognitions on the same bar code.

From the experimental results, and the discussion above, the following conclusion is drawn: the presented method can robustly extract and recognize EAN-13 bar codes from low-quality images captured by a camera phone without auto focus or macro modes, and multiple shots of one bar code will result in very reliable recognition, which makes the bar code reading method practical for use on low-end

camera phones. Although the method is intentionally developed for reading EAN-13 standard bar codes, it can be easily generalized to read other 1D standard bar codes as well.

Acknowledgments

The authors would like to thank Jari A. Kangas (senior research manager), Markku Vehviläinen (research manager), Adrian Burian (senior engineer) from Nokia Research Center (NRC) and Jukka Yrjänäinen from Technology Platform, Nokia, for their support and encouragement that made this work possible. We would also like to thank Professor Moncef Gabbouj from Tampere University of Technology, and Dr. Kari Pulli, research fellow from NRC, for their technical opinions on the paper. Finally, we would like to thank Dr. Xuelong Li and Ian Hunneybell from Birkbeck College, University of London for revising and commenting on the paper.

References

1. J. Uusikylä, *Bar Code Recognition in Digital Image*, Master of science thesis, TUT (2000).
2. E. Joseph and T. Pavlidis, "Bar code waveform recognition using peak locations," *IEEE Trans. on PAMI* **16**(6), 630–640 (1998).
3. S. J. Shellhammer, "Novel signal-processing techniques in bar code scanning," *IEEE Robotics & Automation Magazine*, pp. 57–65 (1999).
4. W. Turin and R. A. Boie, "Bar code recovery via EM algorithm," *IEEE Trans. on PAMI* **46**(2), 354–363 (1998).
5. M. Kuroki, T. Yoneoka, T. Satou, Y. Takagi, T. Kitamura and N. Kayamori, "Bar code recognition system using image processing," *6th International Conference on Emerging Technologies and Factory Automation Proceedings*, pp. 568–572 (1997).
6. E. Joseph and T. Pavlidis, "Waveform recognition with application to bar codes," *IEEE International Conference on Systems, Man and Cybernetics* **1**, 129–134 (1991).
7. S. Kresic-Juric "Edge detection in bar code signals corrupted by integrated time-varying speckle," *Pattern Recognition* **38**, 2483–2493 (2005).
8. A. K. Jain and Y. Chen, "Bar code localization using texture analysis," *Proceedings of the Second International Conference on Document Analysis and Recognition*, pp. 41–44 (1993).
9. R. Oktem, "Bar code localization in wavelet domain by using binary morphology," *Proceedings of Signal Processing and Communications Applications*, pp. 499–501 (2004).
10. S. J. Liu, H. Y. Liao, L. H. Chen, H. R. Tyan and J. W. Hsieh, "Camera-based bar code recognition system using neural net," *Proceedings of 1993 International Joint Conference on Neural Networks* **2**, 1301–1305 (1993).
11. K. Wang, Y. Zou and H. Wang, "Bar code reading from images captured by camera phones," *2nd IEE International Conference on Mobility Technology, Applications and Systems* (2005).
12. I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. Pure Appl. Math* **41**, 909–996 (1988).
13. M. W. Mak, "A lip-tracking system based on morphological processing and block matching techniques," *Signal Processing: Image Communication* **6**, 335–348 (1994).
14. <http://www.export911.com/e911/coding/ean13.htm#xEAN13Sym>.

15. T. Korhonen, *Self-Organization and Associative Memory*, 3rd ed. Springer-Verlag (1989).
16. T. Fukumoto, T. Wakabayashi, F. Kimura and Y. Miyake, "Accuracy improvement of handwritten character recognition by GLVQ," *IWFHR VII*, pp. 271–280 (2000).
17. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd Edition, New York: Academic Press (1990).

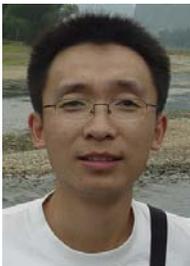


Kongqiao Wang received his MSc in 1996 and his PhD in 1999 in signal and information processing from Hefei University of Technology and University of Science and Technology of China respectively.

He joined Nokia Research Center (NRC) China in December 1999. From 1999 till now, he has been participating and managing quite a lot of projects relating to image analysis and pattern recognition. He was promoted to be a research manager in September 2001. Currently he is leading visual interaction systems research team in System Research Center (SRC) Beijing, NRC, meanwhile, he is also the principal expert of natural human interaction in SRC Beijing. His research focus is UI interaction by using image analysis and pattern recognition. He has achieved more than 20 granted patents and patent applications, published four book chapters and more than 20 conference and journal papers as the first author.



Yanming Zou received his BSc degree and a PhD degree in electronic and communication system from Beijing Institute of Technology in 1994 and 1999 respectively. He joined Nokia Research Center (NRC) China in April 1999. Now he is a member of research staff in System Research Center Beijing, NRC. His research interests include genetic algorithms, statistical pattern recognition and image processing.



Hao Wang is a member of research staff in System Research Center Beijing, Nokia. He received his BSc degree and MSc degree in electronics engineering from Tsinghua University, Beijing, in 1998 and 2000 respectively. He is currently pursuing his PhD in the Electrical and Communications Engineering Department of Helsinki University of Technology, Finland, majoring in computational engineering. His research interests include image processing and analysis, pattern recognition and computer vision.