Volume 49, Number 3, 2008

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

CAMERA PHONE BASED BARCODE DECODING SYSTEM

Romulus TEREBES(1) Otmane LAHLOU(2)
(1)The Technical University of Cluj-Napoca, Romania, (2)ENSEIRB, France
15 C. Daicoviciu Street,Cluj-Napoca, Romania, 1 av. du docteur Schweitzer,33402, Talence cedex, France

Abstract: We propose a 1D barcode acquisition and decoding system based on the use of J2ME enabled mobile phones. The approach relies on image processing techniques to correct the distortions introduces by the acquisition device, segments the useful information, decodes it and subsequently sends it to a web server for further processing.

Key words: *Image processing, 1D barcodes, Java 2 MicroEdition.*

I. INTRODUCTION

Barcodes are used nowadays to a large extent to represent in optical machine readable format large quantities of data. Should they been used in sales, document management, item tracking applications or web link encoding, their main advantages rely on the increased processing speed and accuracy.

Originally, bar codes represented data in a 1D format with widths and spacings of parallel lines; *Figure 1* represents such a format following the EAN (European Article Number) 13 standard.

Figure 1. EAN 13 barcode

An EAN 13 barcode encodes the following piece of information: a system (or country) code, a manufacturer code, a product code and a check digit for allowing correct decoding when processing such an image with an optical scanner. Further details about the specific encoding of a EAN 13 barcode can be found in [6].

Besides the 1D format several 2D formats have been standardized recently and their main advantage is the increased data encoding capacity. Such a barcode symbology can encode text, web links or telephone number; *Figure 2* shows an example of such an encoding using the QR (Quick Response) 2D barcode standard. The actual information encoded is: "*Welcome to SPSWC2008. Visit us at:http://www.sp.utcluj.ro/html/spswc2k8.html*".

An overview of the 2D barcodes and their specific encoding rules can be found in [6].

Among other telecommunication services and technologies the wireless and mobile communications technologies are witnessing probably the highest penetration and development rates and, together with the advent of mobile phones, came the need to implement new services for these emerging technologies.

Within this context we propose a mobile phone based 1D barcode decoding system with possible uses in m-commerce or mobile warehouse like applications.

Figure 2. QR 2D barcode

The subject addressed by this paper is quite new and relevant publications in the domain date from 2006. In [4]-[5] the authors propose a method similar in spirit with our approach, devoted to high-end mobile phone endowed with Symbian technology. Essentially they rely on a multiple scan line decoding approach working on high resolution images captured with auto focus or macro enabled mobile phone cameras.

In contrast with this method our approach is designed to work on entry-level mobile phones using the Java 2 MicroEdition programming environment. Furthermore, our algorithm does not need auto focus or macro like features as it corrects the distortions introduced by the acquisition device using image processing techniques.

II. BARCODE DECODING AND PROCESSING SYSTEM

We propose the following architecture for our barcode processing system (*Figure 3*):

Manuscript received September 1, 2008; revised October 5, 2008

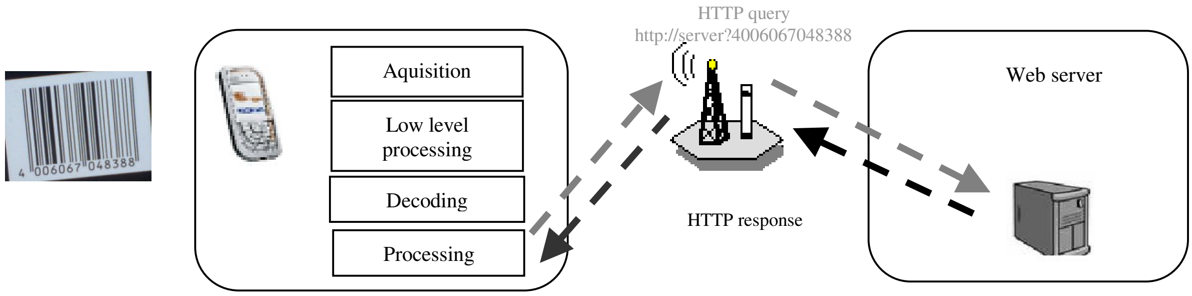


Figure 3. High level description of the proposed system

Most of the intelligence of the approach relies on the client side that implements all the required operation for decoding the barcode image. The decoding algorithm and the low level image processing tasks used for pre-processing the input images are detailed in the next section.

III. PROCESSING AND DECODING ALGORITM

The proposed flowchart for the low level image processing techniques used to correct the distortions introduced by the acquisition device and to decode the image is shown in *Figure 4*.

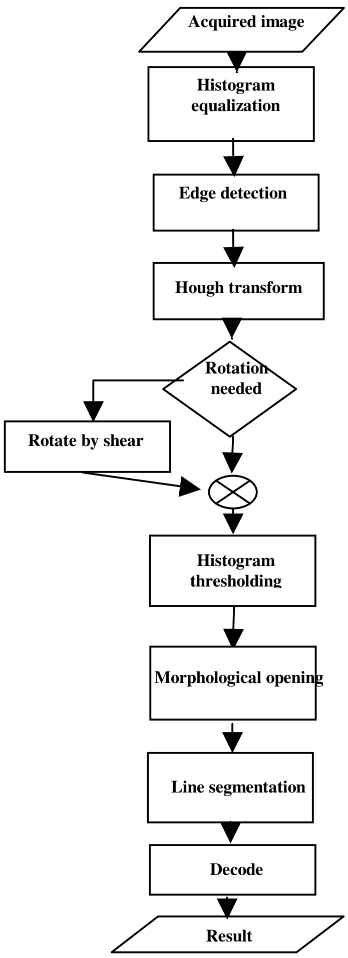
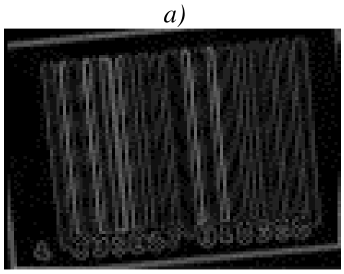


Figure 4. Flowchart of the proposed approach

A. Histogram equalization and edge detection

These preprocessing steps are devoted for enhancing the contrast of the acquired image and for detecting the relevant line like information to be further processed by a Hough transform [1],[2]. The output of these steps is shown in *Figure 5.*

a)



b)

Figure 5. Histogram equalization and edge detection. a) Acquired image (320x240 pixels); b) Sobel edge image

B. Hough transform

The Hough transform is a technique which can be used to detect features of particular shapes within an image [2]. In the context of our approach we use it to detect eventual rotations of the edge image issued as a result of the image acquisition process.

The basic idea consists in specifying the desired feature in a parametric form; since we are interested in line like patterns we employ a polar representation:

$$r = x\cos(\theta) + y\sin(\theta) \qquad (1)$$

Using such a representation all collinear points i.e. pixels with high gradient norms, map to sinusoidal curves in the Hough space.

The values of an accumulator (a bidimensional matrix) are incremented for each r and θ allowing x,y to sweep the input image; a simple maximum detection yields then the orientation of the rotated image. In case of involuntary rotations during the acquisition process the image need to be further processed with an inverse operation.

C. Image rotation

All image rotation algorithms are susceptible of degrading the quality of the acquired image due to the inherent interpolation process needed to compute luminance values with sub pixel resolution.

We employ for our application the rotate-by-shear approach, classically reported in the literature to produce the best result with a low computation complexity [3], a desired feature when using mobile phones with limited amount of memory and low processing speed.

The rotation equation can be but in the following terms:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & -\tan\theta/2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin\theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan\theta/2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (2)$$

and it actually consists in three shear operations: a horizontal shear followed by a vertical and another horizontal one.

The results in rotating the input image from *Figure 5* are shown in *Figure 6*; the orientation detected using the Hough transform is used as a parameter.

Figure 6. Image rotation using the rotate by shear algorithm.
a) Acquired image; b) Shear in x;
c) Shear in y; d) Shear in x (final image)

D. Histogram thresholding and morphological opening

The goal of these preprocessing steps is to obtain a compact representation of the information included in the EAN 13 barcode.

We employ a histogram thresholding algorithm based on the IsoData clustering approach. The approach is fully automatic, we start with an initial threshold T_0 - separating background from foreground – then the threshold is refined according to the following equation:

$$T_k=(m_{f,k}+m_{b,k})/2 \quad (3)$$

In (3) $m_{f,k}$ and $m_{b,k}$ are denoting respectively the means computed at the *k*'th iteration on the foreground and background pixels, with the pixels being classified according with respect to their proximity to the closest mean.

Morphological opening plays an important role in mathematical morphology since it is an operator devoted to close small gaps affecting the foreground of the processed image. By denoting the input image with *U* and the structuring element with *B*, morphological opening consists in an erosion operation followed by a dilation as in Eq. (4):

$$U \circ B=(U \ominus B)\oplus B \quad (4)$$

The results of these preprocessing steps are shown in *Figure 7*; for the opening operator we employed a horizontal 7x1 pixels sized structuring element.

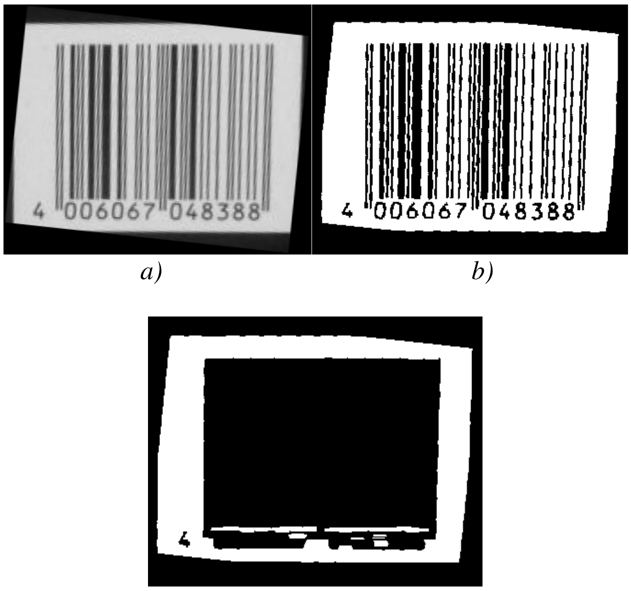
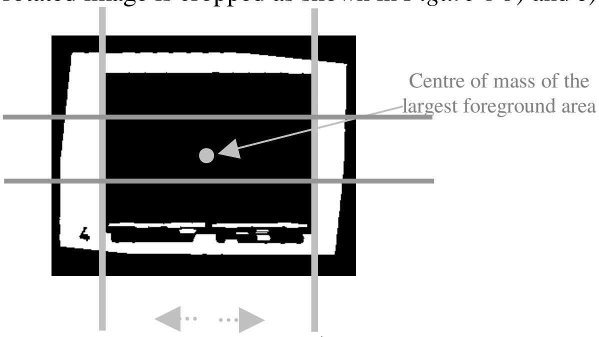


Figure 7. Image thresholding and morphological opening results. a) Rotated image;b) Thresholded image; c) Morphological opening result.

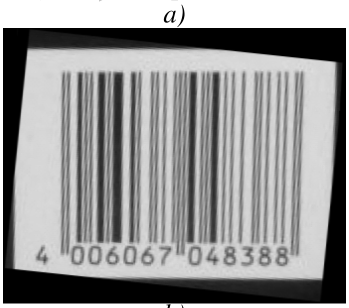
E. Line segmentation

As a result of the previous steps the preprocessed image is a rescaled and rotated version of the input image and, in order to accurately extract only the vertical patterns, a supplementary segmentation algorithm is necessary. Line segmentation is achieved starting from the centre of mass of the largest foreground area corresponding to the morphological opening result and is performed as a simple detection of the largest contiguous zone consisting in black pixels. On the vertical direction the principle is similar; however only one third of the detected height is retained in order to eliminate the alphanumerical characters sitting on bottom of the EAN 13 code.

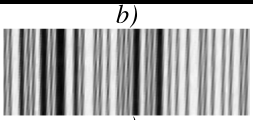
The principle is illustrated in *Figure 8 a)* and, once the relevant area has been detected on the morphological opened image, the corresponding region from the gray scale rotated image is cropped as shown in *Figure 8 b)* and *c)*.



a)



b)



c)

Figure 8 Line segmentation.a) Principle;b)Rotated image; c) Segmented line information

F. The decoding process

The first step in the decoding process is a second derivative based edge detection operation performed on the vertical projection of the segmented image; for the image in *Figure 8 c)* the following pattern is obtained:

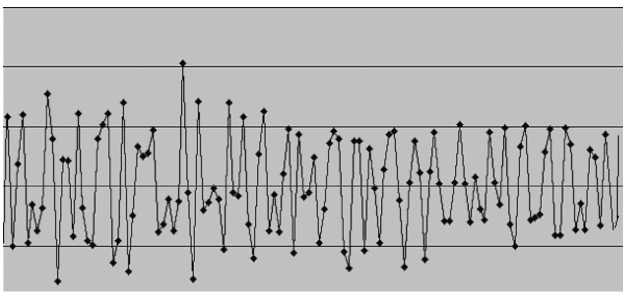


Figure 9 Second derivative based edge detection on the vertical profile of the segmented lines

As mentioned previously the image representing the actual information encoded in the vertical patterns of an EAN 13 image is a scaled version of the input one; consequently we perform a calibration process in order to detect the standard width of a line after all preprocessing steps have been done. This a achieved by a simple division operation:

$$standard\_width = image\_width/95 \qquad (5)$$

with 95 representing the number of elementary vertical bars in a EAN 13 code as specified in [6].

Once the standard width has been computed, the second order derivative information can be used to reconstruct the barcode in terms of black and white bars of standard width each. The principle is illustrated in *Figure 10:*

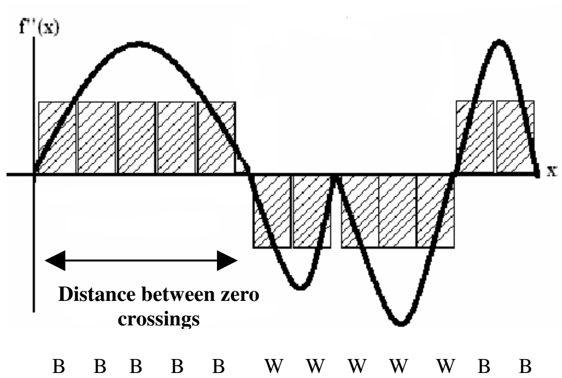


Figure 10. The decoding process

The alphanumerical string corresponding to the information encoded in the barcode itself is found using a Pearson correlation coefficient, computed between the code formed by the black and white reconstructed pixels and previously stored representations of the possible character encodings.

Finally we compute the check digit in order to insure that the information has been decoded successfully and dedicated messages boxes are used to inform the user about the outcome of the decoding process.

IV. THE CLIENT APPLICATION

The client application was developed using the Netbeans 4.1 IDE shipped with the supplementary Mobility Pack, and it uses besides the standard J2ME MIDP profile the MMAPI package, needed for capturing images using a VideoControl object supporting playback of video.

In terms of visual design and of MIDPs *lcdui APIs* the application is shown in *Figure 11.*

The obtained ASCII string is displayed on the mobile phone or it is sent for further processing at a web server using the embedded HTTP communication APIs of J2ME on a cellular 2.5G/3G support.

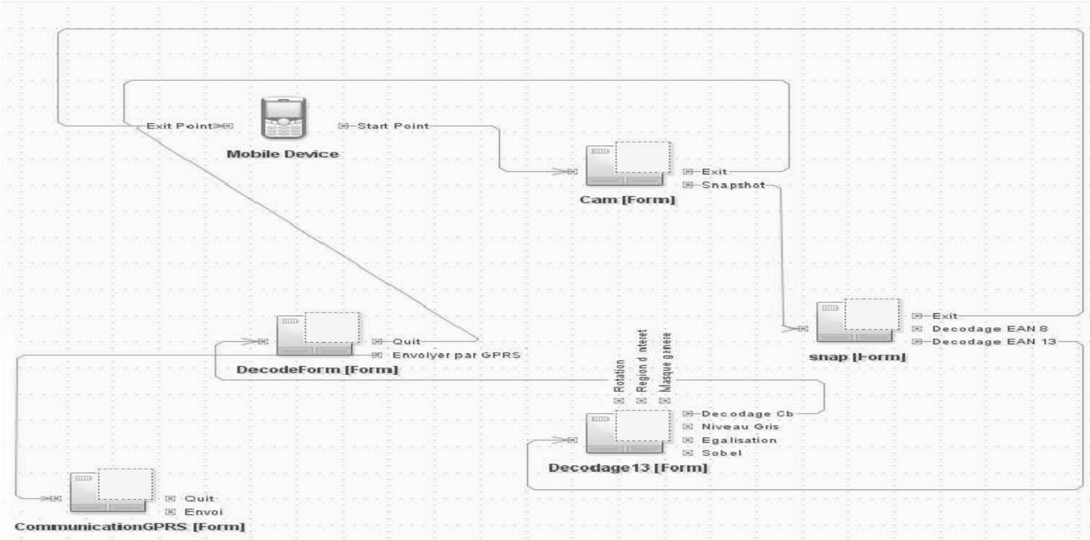


Figure 11 The client application

IV. EXPERIMENTAL RESULTS

We tested our algorithm on variety of images, affected by involuntary rotations, low contrast and non-uniform illumination conditions. On this type of images we achieved a 88% recognition rate - computed on a database consisting of 100 images taken at different resolutions. *Figure 12* shows some examples of correctly decoded images:

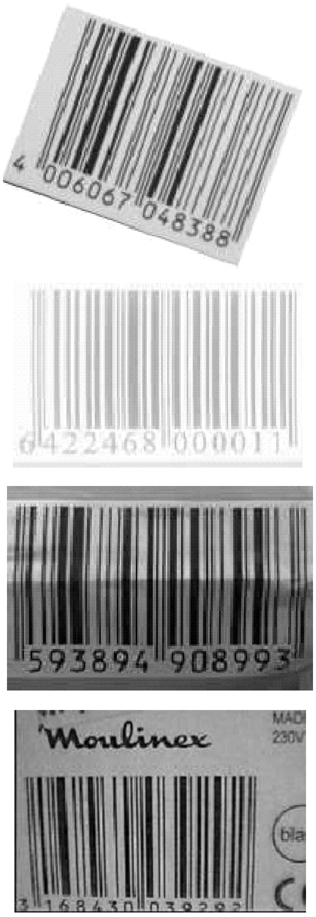


Figure 12. Successfully decoded images

The current version of the algorithm cannot cope with tilt like distortions or with out of focus images; however dedicated image processing tasks can be easily implemented in order to handle those types of degradations *(Figure 13).*

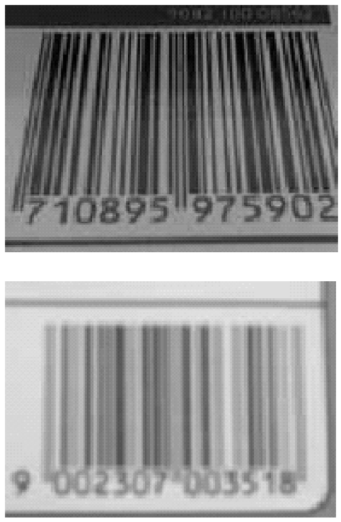


Figure 13. Examples of incorrectly decoded images

V. CONCLUSIONS AND FUTURE WORK

We proposed a barcode decoding algorithm that runs a mobile phone. Contrary to existing methods [4]-[5], our approach can be used on J2ME enabled entry level mobile phones equipped with low resolution (at most 320x240 pixels in all our experiments), without macro/ auto focus features and Symbian operating systems.

In terms of performances we achieved an 88% recognition rate for EAN13 codes using a database consisting in 100 images. Future work will be devoted for generalizing the application to handle other types of 1D and 2D barcodes.

REFERENCES

[1] R. Gonzales, R.Woods, Digital Image Processing, *Prentice Hall,* second edition, 2001.
[2] Al. Bovik, Al.C. Bovik, Handbook of Image and Video Processing, *Academic Press,* 2005
[3] M. Unser, P. Thevenaz, L. Yaroslavsky, "Convolution-based interpolation for fast, high-quality rotation of images", *IEEE Transactions on Image Processing* vol.4, no.10, 1995, pp.1371-1381.
[4] R. Adelmann, M. Langheinrich, and C. Floerkemeier, "A toolkit for barcode recognition and resolving on camera phones - jump starting the internet of things", *Workshop Mobile and Embedded Interactive Systems (MEIS'06)* at Informatik 2006, Oct. 2006.
[5] R. Adelmann, "Mobile Phone Based Interaction with Everyday Products - On the Go**",** *The 2007 International Next Generation Mobile Applications, Services and Technologies*, NGMAST' 07, 2007. 12-14 Sept. pp.63 – 69
[6] EAN System specification: http://gs1.se/global/Om-GS1-Systemet/GenSpec_v8_i1_Jan_2008.pdf
[7] QR codes overview: http://www.denso-wave.com/qrcode/aboutqr-e.html
[8] S Li, J. Knudsen, Beginning J2ME: From Novice to Professional (3rd Ed.), *Apress Inc,* 2005.