IMAGE PRE-PROCESSING FOR BAR CODE DETECTION IN MOBILE DEVICES

Radu Ciprian Bilcu, Adrian Burian, Markku Vehvilainen

Multimedia Technologies Laboratory
Nokia Research Center
Visiokatu 1, 33720, Tampere, Finland

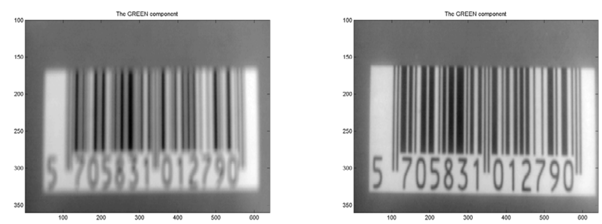
ABSTRACT

In this paper we present an image pre-processing procedure for bar code detection in mobile devices. The goal of our method is to improve the quality of the input image, thus making bar code detection and decoding possible even in difficult situations. The implementation details and the results obtained with the proposed method on real images taken with a camera phone, are discussed.

1. INTRODUCTION

In the near future there will be more and more demand for mobile devices, such as camera phones, equipped with bar code detection facility. Usually, the devices utilized for bar code detection are using laser beams (for example, the dedicated devices used in supermarkets and stores). Another alternative is to use a digital camera for bar code reading. Since many mobile devices are already equipped with digital cameras, they can incorporate this kind of application as well.

The physical dimension of the bar codes to be detected have a wide variety of ranges. In fact, one major reason why the bar codes are widely used today and they keep developing is their high scanning reliability and the possibility to include high quantity of information in a very small printed area. Illustrative examples are given, e.g., in [1, 2]. Bar codes that have low information density printed on a large area can be detected easily with a normal camera phone. This is because, for such bar codes, the distance between the printed bar code and the lens of the camera phone must be large in order to include the complete bar code in the viewfinder. In this case the image containing the bar code is situated in the in-focus range of the camera, and a good enough quality for detection and correct decoding is obtained. For the bar codes printed on small areas and with high information density, their resolution comes close to the maximum resolution of the optical system. In order to deal with this problem the bar code is captured at a distance that is much smaller than the focal distance of the digital camera, and the captured image is highly influenced by the distortions of the optical system and sensor of the camera (out of focus). Typically, the captured image is distorted by: sensor noise, optical blur and optical vignetting. An example of such image obtained with a camera phone for a small distance between the camera and the printed bar code is depicted in Fig. 1 a). It is clear that bar code decoding applied directly to this image will fail due to the amount of blurring, therefore image preprocessing becomes a necessary step.



a) b)

Fig. 1. A bar code captured with a camera phone: a) at out of focus distance, b) at in-focus distance.

We must also specify that other ways to improve the quality of the captured image, for small bar codes, exists. A second solution (which replaces to some point the need of preprocessing) is to use macro lenses with a very small focal distance that introduce very little blur in the captured image. The same image as in Fig. 1 a) captured with a camera phone equipped with macro lens is depicted in Fig. 1 b). However, the use of such lens requires additional external attachments to the mobile phones that can cause discomfort to the users and add more costs. A third solution can be to use an imaging mobile phone equipped with auto-focus.

In this paper we will consider the images obtained with imaging phones that are not equipped with macro-lenses and without auto-focus. Due to the miniaturization of the digital camera module, the optics is a limiting factor in the mobile devices industry. The captured image will always be blurred and suffer from high noise content. Due to these realities, there will always be need for image pre-processing in order to obtain robust bar code detection.

Previous research work has shown that this problem is very difficult. Different de-blurring techniques for bar code application than the ones we are using are presented in [3]. The only similarity between our methods is the estimation of the point-spread function (psf) in the observed image. How-

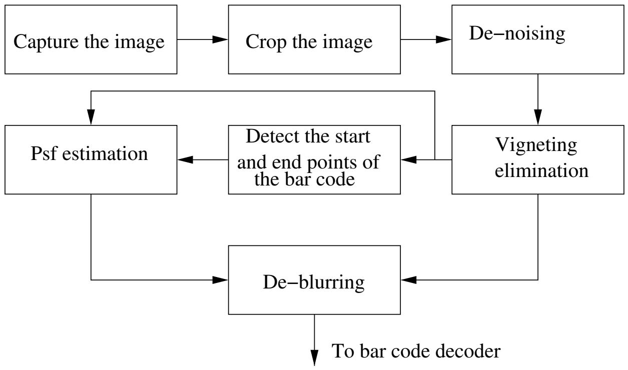


Fig. 2. The block diagram of the proposed method.

ever, the used estimation method in [3] is based on the derivative of the observed image while we are using the mean squared error. In [4, 5], a partial de-blurring is performed first, followed by signal's peak detection. A bi-level bar code is blurred and fitted to the input line in [6, 7]. The psf is assumed to be gaussian with unknown parameters and the positions of the bar code edges are assumed to be unknown. The maximum likelihood approach is used to find the position of the bar code edges. Neural networks are used in [8], and mathematical morphology in [9].

In this paper we describe a pre-processing method applied to the captured image in order to reduce the different distortions introduced by the sensor and the optical system. The pre-processing is tailored for the use in a system for bar code detection and decoding with severe blurring and noise when close-up capture shots are needed. Moreover, the proposed solution can be implemented for raw data images as well as for interpolated ones. Our algorithms are adaptive such that the user does not have to set the value of any parameter (i.e., the psf or vignetting functions are automatically computed from the input image).

2. PROPOSED APPROACH

In this section we describe in detail the proposed method, whose main steps are illustrated in Fig. 2.

Image capturing: The algorithm described in this paper was applied to one of the green component of the camera raw image. It is also possible to use the gray scale image or the color image from the output of the camera chain.

Crop the image: just a small part of the input image is retained in order to increase the processing speed. For instance, several adjacent lines from one of the image components are retained (in our experiments we have used 20 adjacent lines around the line of interest). We have done this for the following reasons. First, the next step of de-noising can be implemented in several different ways using algorithms that necessitate more input data. Secondly we are trying to retain as much information as possible in order to obtain good de-noising performances.

De-noising: a noise reduction algorithm is applied to the cropped image. The de-noising method in [10] has been used. Just the filtered line of interest is passed further through the next processing steps.

Vignetting elimination: the process of vignetting introduced by the optical system of the camera is inverted. First, the white borders of the bar code are detected. Some local averages at the beginning and at the end of the selected line are computed. From the computed averages we select the larger ones that corresponds to the white areas surrounding the bar code. If the white borders are detected successfully, the next steps of the algorithm are executed. Otherwise, the whole process is skipped and a new input image is captured.

An example of a selected line containing a bar code is depicted in Fig. 3. Here we can clearly see the presence of the white border. The algorithm calculate two local mean values at the left of the bar code (denoted as A and respectively B in Fig. 3) and one mean value at the right side of the code (denoted as C in Fig. 3). We assume that the vignetting function can be modelled as a polynomial of second order:

$$y(x) = Dx^2 + Ex + F \tag{1}$$

where x is the horizontal pixel coordinate and D, E and F are constant coefficients.

The following system of equations is formed using the three above computed averages:

$$\begin{cases} A = Dx_1^2 + Ex_1 + F \\ B = Dx_2^2 + Ex_2 + F \\ C = Dx_3^2 + Ex_3 + F \end{cases} \tag{2}$$

where x_1, x_2 and x_3 are the coordinates of the three local averages A, B and respectively C.

The above system of equations is solved for D, E and F and the selected line is corrected dividing it by the vignetting function from (1). Fig 3 shows also the estimated vignetting function with dotted line and the corrected image line with dashed line.

Detect the start and the ending points of the bar code: From the experiments we made, we realized that one reliable way to detect the start and ending structures of a bar code is to detect a succession of large white area followed by a sharp transition to black. When considering the line we are processing, this is done by filtering with a window of length 15. The used one-dimensional filter is a difference between the maximum and minimum values of the signal inside the filter's window. The minimums of the filtered values closest to the both endings of the signal are considered to belong to the white areas of the corresponding image. First large transitions after these points in both directions (from left to right, and right to left), give the coordinates we are looking for. This algorithm can also be used to detect the white borders of the bar code in the previous step.

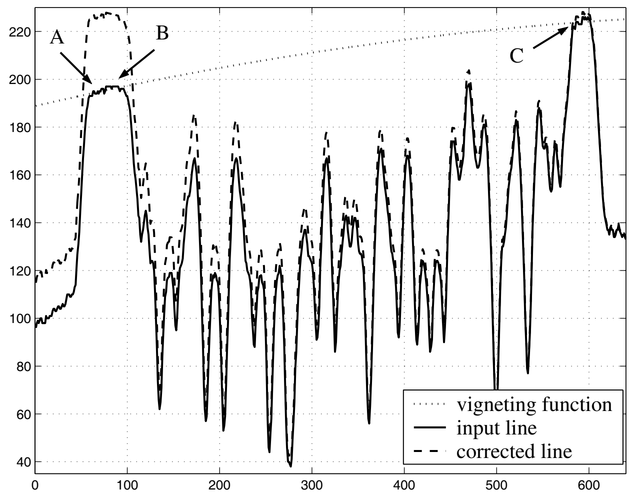


Fig. 3. The selected line, the corrected line, the estimated vignetting function and the three points used in vignetting estimation.

Point spread function (psf) estimation: the psf is estimated from the selected line based on the starting and ending points of the bar code using some a-priori known information related to the bar code we are trying to decode.

All existing bar codes, 1D or 2D, have embedded known lines or shapes in a-priori known positions related to the bar code structure. Let us consider as example the 1D bar code EAN13. There is a fixed and known number of bars composing the EAN13 bar code. The bar code structure starts, is centered, and ends with a group of 5 lines with equal width that are arranged as follows: WHITE LINE, BLACK LINE, WHITE LINE, BLACK LINE and WHITE LINE. This group of 5 lines are clearly seen in Fig. 1. They are longer than the rest of the lines and are situated at the beginning, center and ending parts of the code.

Having this information we detect the psf as follows:

1. Compute the length of one bar code line:

$$L = \frac{x_2 - x_1}{T} \tag{3}$$

   where x_1 and x_2 are the coordinates of the beginning and ending points of the bar code and T is the total number of bars ($T = 97$ in our experiments).

2. Knowing that every bar code starts with the above mentioned set of 5 lines, each of length L, we cut from the processed line the part that corresponds to the first 5 bars. We denote this part of the processed line as Y.

3. We assume that the psf, denoted as P, have a Gaussian shape with length equal to $5L$ and variance σ^2. We generate the ideal clean line V, corresponding to Y (the line containing 5 bars that was blurred by the optical system to obtain Y). We blur the ideal line V with generated psf's P for several values of σ^2 (we first start with a small σ^2 for instance 0.1 and we increase its value at every iteration). At each iteration, we compute the mean squared error between the blurred line and the input line Y. For the first iterations the mean squared error decreases. When the variance becomes too large (it goes over the optimum), the mean squared error starts to increase. At this point the iterative process is stopped and the last good value of σ^2 is retained.

4. We repeat 2. and 3. for the central and ending areas of the bar code. The obtained values of σ^2 will linearly establish the used variance along the bar-code structure.

Finally, we have an estimation of the psf that is Gaussian shaped, with length equal to $5L$ and variance computed as in the above algorithm.

Line de-blurring: the selected line after vignetting elimination is restored (de-blurred) using the estimated psf. This process can be done by several algorithms that exist in the open literature. In our pre-processing chain we have used the approach from [11].

3. EXPERIMENTAL RESULTS

Extensive experimental work was done in order to test the performances of our proposed pre-processing method. We have implemented this method in Symbian and installed it on a camera phone. The system equipped with our pre-processing method was able to correctly decode the more difficult captured bar codes that otherwise failed with our decoder (see for instance Fig. 1 a)).

In Fig. 4 we show an example of a restored image using our proposed approach (for input image shown Fig. 1 a)). The image in this figure is obtained repeating the de-blurred line 100 times (the lines from Fig. 4 are all identical). We have chosen to plot the restored line in this way just to have a more clear illustration of the achieved processing results.

In Fig. 5 the result after segmentation is presented. We notice the fact that all bars are present into the image and their sizes are correct. Since each step from the proposed method, except de-noising, processes only one line from the image, the whole process is very fast(the delay is not annoying to the phone's user).

4. CONCLUSIONS

In this paper we have proposed a practical approach of image pre-processing for bar code detection in mobile devices. Our proposed method shown good performances in restoring bar code images. A real bar code detection system, implemented in Symbian on a Nokia 6630 camera phone and using

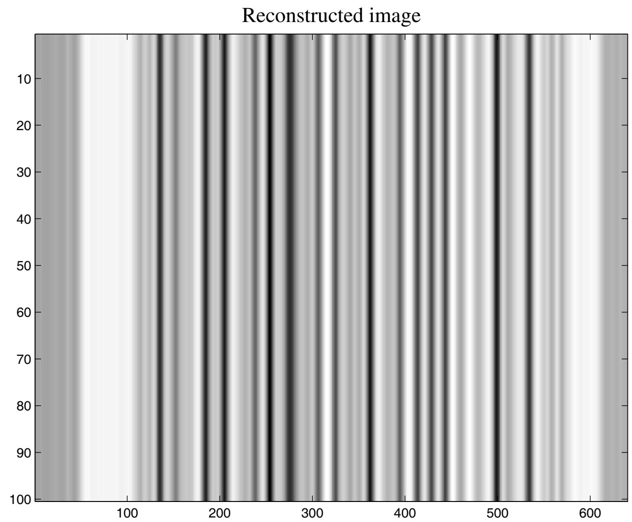


Fig. 4. The restored bar code image.

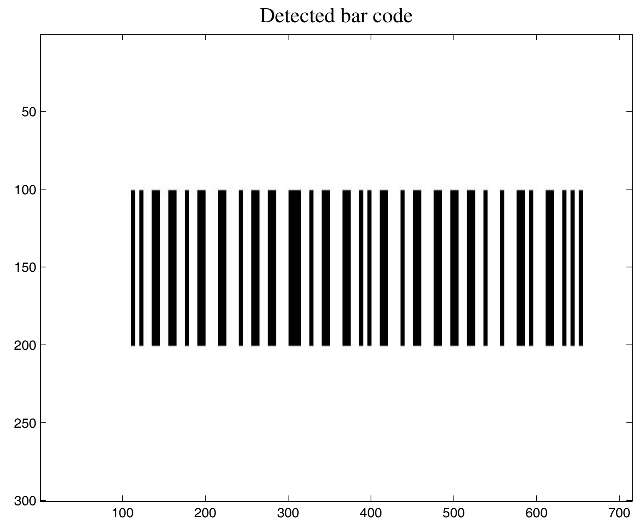


Fig. 5. The detected bar code after segmentation of the restored image.

our method, successfully decoded difficult bar codes that otherwise failed to be decoded. Moreover the complete system is able to operate in real time.

5. REFERENCES

[1] D. Kirovski and N. Jojic, "Cryptographically Secure Identity Certificates," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1-1, May 2004, pp. V413–V416.

[2] Z. Cai, "A new Decode Algorithm for Binary Bar Codes," *in Elsevier Pattern Recognition Letters*, vol. 1, no. 1, pp. 1191–1199, Dec. 1994.

[3] E. Josepha and T. Pavlidis, "De-blurring of Bilevel Waveforms," *in IEEE Transaction on Image Processing*, vol. 2, no. 2, pp. 1–1, Feb. 1993.

[4] E. Josepha and T. Pavlidis, "Bar Code Waveform Recognition using Peak Locations," *in IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 1–1, Feb. 1994.

[5] E. Josepha and T. Pavlidis, "Waveform Recognition with Application to Bar Codes," in *Proceedings of IEEE International Conference on Decision Aiding for Complex Systems*, 1-1, Oct. 1991, pp. 129–134.

[6] W. Turin and R. A. Boise, "Bar Code Recovery via the EM Algorithm," *in IEEE Transaction on Signal Processing*, vol. 46, no. 2, pp. 1–1, Feb. 1998.

[7] W. Turin and R. A. Boise, "Minimum Discrimination Information Bar Code Decoding," in *Proceedings of Nineteenth Convention of Electrical and Electronics in Israel*, 1-1, Nov. 1996, pp. 255–258.

[8] 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," in *Proceedings of 1993 International Joint Conference on Neural networks*, 11, Feb. 1993, pp. 1–1.

[9] G. J. Awcock S Arnould and R. Thomas, "Remote Bar-Code Localization using Mathematical Morphology," in *IEE International Conference on Image Processing and its Applications*, 1, Feb. 1999, pp. 642–646.

[10] Radu Ciprian Bilcu and Markku Vehvilainen, "A New Image De-Noising Technique Based on Image Decomposition and Sigma Filtering," *in WSEAS Transactions on Communications*, vol. 4, no. 8, pp. 579–586, Aug. 2005.

[11] L. Liang and Y. Xu, "Adaptive Landweber Method to Deblur Images," *in IEEE Signal Processing Letters*, vol. 10, no. 5, pp. 129–132, May 2003.