A SUPERRESOLUTION APPROACH FOR BAR CODE READING

[1]*Rusen Oktem,* [2]*Levent Oktem*

[1]Electrical and Electronics Engineering Department,
Atilim University, Kizilcasar Koyu, Incek 06836 Ankara, Turkey
phone: + (90) 312 586 8337, fax: + (90) 312 586 8091, email: rusen@.atilim.edu.tr
[2]Synplicity, Inc., Ankara, Turkey
phone: + (90) 312 242 0394, email: levent@.synplicity.com

ABSTRACT

This paper presents a computer vision based method for bar code reading. Bar code's geometric features and the imaging system parameters are jointly extracted from a tilted low resolution bar code image. This approach enables the use of cost effective cameras, increases the depth of acquisition, and provides solutions for cases where image quality is low. The performance of the algorithm is tested on synthetic and real test images, and extension to a 2D bar code (PDF417) is also discussed.

1. INTRODUCTION

Computer vision based bar code recognition is an alternative to laser scanner type recognition systems. Such systems consider the problem in two parts: localize the bar code, and then decode the information encoded in the bar code. Laser scanner bar code readers are generally based on 1D scanning, and they can fail when height/length ratio is low. Furthermore, some applications, e.g. identification of radioactive material containers, require remote identification, hence higher depth of field. The use of computer vision increases the performance of both bar code localizers and readers (decoders) [1-5]. Use of neural networks [2], zero crossing detection [3], and finding the peak locations [4] are examples of computer vision bar code reading methods. However, they require that the barcode image resolution is high enough. In [5], super-resolution is used for reconstructing high-resolution bar code samples from low-resolution ones. This approach enables the use of low-cost cameras and/or increases the depth of field. In [5], the phase difference between consecutive lines in the bar code image is exploited. The blur in the newly reconstructed image is removed by inverse filtering. Two main shortcomings of this algorithm are, the need to compensate for blur, and the difficulty in extending it to 2D bar code reading. In this paper, we also use super-resolution approach, but define a bar code by its geometric features. This definition is independent of imaging parameters like rotation, skew, and resolution. It enables the reconstruction of a high quality bar code image at any desired resolution, and is generalized to 2D bar code reading.

2. BAR CODE READING RULES

2.1 1D Bar Code Reading

1D bar codes are composed of parallel and alternating dark and light stripes. The start, the end, and the middle of the code are identified with guard bands, and a 5 units wide centre pattern, 11 units wide in total. Each digit is represented by two bars and two spaces, and is seven units wide in total. In UPC standard, a bar code encodes 12 digits. Altogether, a bar code width is 95 units (12.7+11). The position and the width of each bar in 7 units identify that digit. See [6] for more details.

2.2 2D PDF417 Bar Code Reading

2D bar codes enable encoding of higher range of information, and provide advanced error correction capabilities. For example, PDF417 can hold up to 2000 characters of information and up to half of the symbol can be destroyed or missing, yet it can still be decoded by its own error correction utility. There exist different types of 2D bar codes such as Maxicode, data matrix, and PDF417. In this work, we deal with PDF417 [7] case. Each PDF417 symbol consists of a stack of vertically aligned rows with a minimum of three rows. Each row includes a minimum of one symbol character, excluding start, stop, and row indicator columns. The start and stop columns are the same for each row. The first codeword (numeric value of the symbol character) always encodes the total number of data codewords in the symbol. The final rows include error correction symbols. See [7] for more details.

3. PROBLEM FORMULATION

In this section, bar code refers to 1D bar code. The formulation will be generalized to 2D case in the latter section. We define a bar code by its relevant geometric features. Using an observed digital image of the bar code, we aim to jointly extract these geometric features and the imaging parameters. For the 1-D bar codes and the simplified imaging model we consider here, the geometric features and imaging parameters consist of a vector containing the distances of start and end points of each bar from a reference point, and the tilt angle with respect to the camera. Start and end points of each bar identifies the relative location and the

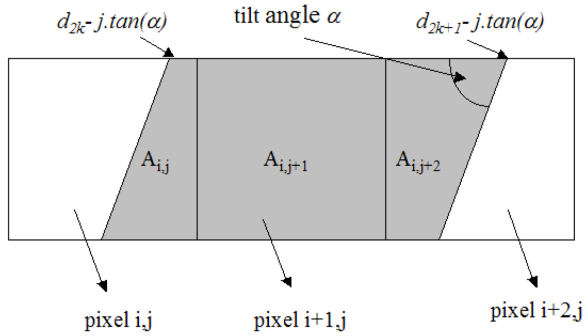

Figure 1. Constructing $f_s(i, j)$. $A_{i,j}$ is the area of the shaded region in pixel (i,j).

width of bars and spaces. A digit corresponding to two bars and two spaces can be recognized from relative locations and widths of these bars and spaces [6]. The tilt angle and pixel size need to be known for calculating how the locations of bars are displaced from one scan line to another. By using all of the above parameters, an ideal bar code image from a low-quality sample can be reconstructed.

3.1 A Mathematical Definition For A Bar Code

Suppose that $f(x, y)$ is a 2D continuous function, representing the intensity value of the bar code at location (x, y), i.e.

$$f(x,y) = \begin{cases} 1, & (x,y) \in B_k \\ 0, & else \end{cases}, \qquad (1)$$

where B_k represents the region bounded by the kth bar. Assume that $f_s(i, j)$ is a discrete function acquired from $f(x, y)$ such that

$$f_s(i,j) = \int_j^{j+1} \int_i^{i+1} f(x,y)dxdy, \qquad (2)$$

where $(i, j) = (0,0),(0,1),\cdots,(1,0),\cdots,(N,M)$. From eqn. 2, (and as shown in Fig. 1), it is seen that $f_s(i, j)$ is the area of the region in pixel (i,j), bounded by the bar passing through that pixel, where each pixel is considered as a square region enclosing 1unitx1unit area. Let $\boldsymbol{d}$ be a $2n+1$ sized vector, the first odd-indexed n elements denoting the distance from the left most pixel, point (0,0), to the point where the left boundary of B_k (kth bar) intersects the top of the pixels at j=0, the first even-indexed n elements denoting the distance from the left most pixel to the point where the right boundary of B_k (kth bar) intersects the top of the pixels at j = 0, and the $2n+1$th element being α, the tilt angle. Boundaries of B_k shift by a factor of $tan(\alpha)$ at each scan line, as shown in Figure 1. Hence, once $\boldsymbol{d}$ is known, boundaries of B_k can be calculated for all scan lines, and bar code image at any size (by increasing M), orientation (by changing α), or resolution (by multiplying the first $2n$ elements of $\boldsymbol{d}$ by a constant) can be reconstructed. Note that number of bars (or n) can be assumed known, since this number is fixed for each bar code standard.

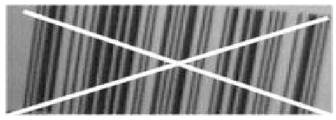

Figure 2. A scratched bar code image.

3.2 Estimating the d vector given a low resolution bar code image sample

We use Newton-Raphson method for nonlinear systems of equations [7], for approximating the $\boldsymbol{d}$ vector given $f_l(i, j)$, an undersampled discrete version of $f(x, y)$. Let us define

- o $\boldsymbol{d_0}, \boldsymbol{d_i}$ as the initial $\boldsymbol{d}$, and $\boldsymbol{d}$ at ith iteration, respectively,
- o $\vec{f}_s^{\,d_i}, \vec{f}_l$ as the vectors whose elements are $f_s^{\,d_i}$ and f_l, respectively, at locations where f_l data are available,
- o $f_s^{\,d_i}$ as the bar code image constructed from $\boldsymbol{d_i}$,
- o $\nabla\vec{f}^{\,d_i}$ as the vector whose elements represent the change in $\vec{f}_s^{\,d_i}$, when $\boldsymbol{d_i}$ is moved in positive x direction by $\boldsymbol{\delta d}$.

Our goal is to converge to a $\boldsymbol{d_i}$ such that the norm of the error $\|\vec{e}\| = \left\|\vec{f}_l - \vec{f}_s^{\,d_i}\right\|$ is minimized.

Such $\boldsymbol{d_i}$ can be found by the following iteration [8]:

Using Taylor expansion to express $\vec{f}_s^{\,d_{i+1}}$,

$$\vec{f}_s^{\,d_{i+1}} = \vec{f}_s^{\,d_i} + \delta d . \nabla \vec{f}_s^{\,d_i}$$

$$\vec{f}_l = \vec{f}_s^{\,d_i} + \nabla\vec{f}_s^{\,d_i}.\delta d \Rightarrow \vec{e} = \nabla\vec{f}_s^{\,d_i}.\delta d$$

$$\delta d = [(\nabla\vec{f}_s^{\,d_i})'\nabla\vec{f}_s^{\,d_i}]^{-1}.(\nabla\vec{f}_s^{\,d_i})'.\vec{e}$$

$$\boldsymbol{d_{i+1}} = \boldsymbol{d_i} + \boldsymbol{\delta d}$$

The iteration stops when $\left\|\vec{f}_s - \vec{f}_s^{\,d_i}\right\| < \varepsilon$.

When the acquired image is of low-quality/resolution, one scan line may not include sufficient information to resolve bars and spaces. The above algorithm combines information from other scan lines corresponding to the missing information, and computes the best fit. Even if a bar code is scratched diagonally from one end to the other (Fig. 2), this algorithm is able to converge to the correct $\boldsymbol{d}$ vector, where a scan line bar code reader will fail.

Figure 3. A PDF417 bar code including 10 rows and 4 columns.

4. GENERALIZATION TO 2D BAR CODES

There exist different types of 2D bar codes, each generated and read in different ways. Yet, they all are composed of black and white bars or rectangles, and they are decoded by relative width and position of those bars or rectangles. Hence, the above formulation can be extended to reconstruct any 2D bar code, too, but the extension changes for each type of bar code. In this work, we focus on PDF417, and present the extension of the reconstruction formulation. PDF417 can be considered as a series of 1D bar codes, concatenated in vertical direction (Figure 3). For that case, ***d*** vector is replaced by **D** matrix, where each row of **D** is formed by ***d*** vector of the corresponding row of PDF417 code. B_k, the kth bar notation is replaced with $B_{k,m}$, denoting kth bar of mth row. Note that row does not mean a scan line, but a number of scan lines, which encode the same information repeatedly, i.e. 1D bar code can be considered to include only 1 row. The difficulty arises at row crossings when the bar code is inclined, because one scan line will include samples from different rows. As shown in Figure 4, a scan line may start with sampling from row 1 and end in sampling from row 2. For this reason, in the $f_s(i,j)$ construction part of the algorithm, a check is included. The check decides which row of the PDF417 code the current pixel belongs to, and compute the area bounded by $B_{k,m}$, accordingly. Check can be evaluated if the tilt angle and row length (the same for each row) are known.

Note that the start and stop patterns in PDF417 are fixed, and are the same for each row. This information is used as a constraint in the reconstruction algorithm. Otherwise, the formulation is just the same as in Section 3.

Also note that PDF417 provides error correction capabilities; the first codeword encodes the number of total codewords, the number of rows, number of columns and security level are encoded alternately in left row indicator and right row indicator codewords, etc. PDF417 decoding performance can be increased extensively by exploiting the information encoded in error correction and row indicator columns. This is left as a future work.

5. EXPERIMENTAL RESULTS

The bar code images are inverted and scaled, i.e. intensity values are subtracted from 255 and scaled with 255, in order for to comply with the formulation. In the experiments UPC standard is assumed, and the algorithm is forced to find

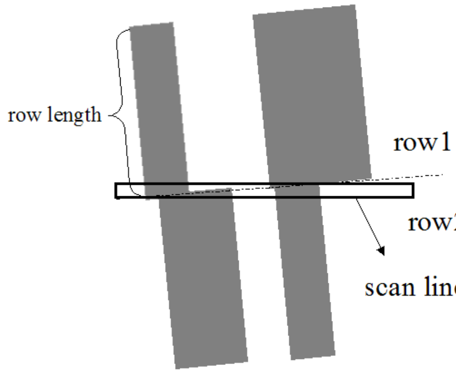

Figure 4. Scanning a line over a tilted 2D bar code image.

Figure 5. a) Real bar code image, 100 pixels wide b) Synthetic bar code image, 50 pixels wide. c) Reconstructed bar code image from a. d) Reconstructed bar code image from b.

predetermined number of bars, which is 30 for UPC. The tilt angle is included as the final element of the ***d*** vector, and is estimated from the data itself. The resolution (width of the bar code in pixels) is assumed unknown. Our experimental findings can be summarized as follows:

- With a real bar code image (Fig. 5.a), acquired with a webcam and resized using a commercial image editing software, convergence to a ***d*** that provides correct reading is achieved when a scan line of the bar code spans minimum 100 pixels. At between 75 and 100 pixels, converged ***d*** vector provides incorrect readings for one to three digits. This is mainly due to camera blur.

- With a synthetic test image (Fig. 5.b), convergence to a ***d*** which provides correct reading is achieved at the following resolution levels:
    - Over 77 pixels (one scan line includes 77 pixels through the width of the bar code), when $\boldsymbol{d_0}$ is estimated from f_l.
    - Over 50 pixels, when $\boldsymbol{d_0}$ is obtained by rounding a correct distance vector to nearest integers (i.e. when $\boldsymbol{d_0}$ is the best integer estimate).
- The algorithm provides correct reading when the bar code is scratched from one diagonal to the other (Fig. 2).

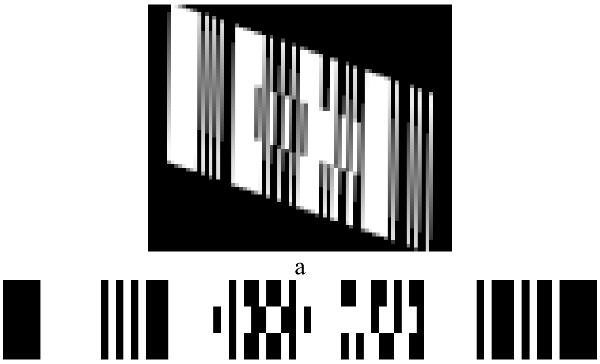

Figure 6. a) Synthetically generated undersampled 2D barcode. b) Reconstructed 2D barcode from a.

In order to show the proof of concept for 2D bar code decoding, we used a toy example. A 65x80 2D bar code image is generated (Figure 6). This is not a valid 2D barcode in the sense that the encoded information is meaningless, but it obeys the rules of PDF417 generation otherwise. The bar code is composed of 3 rows and 5 columns including start, stop, and row indicator columns. Since each column should be 17 units wide [7], and $80 < 17 \cdot 5$, the bar code is undersampled. The image is tilted with 85-degree angle. The tilt angle and row size are assumed known. Figure 6.b shows the reconstructed 2D bar code by using the extended bar code recognition algorithm.

6. CONCLUSIONS

We propose a technique to extract the geometric features of a bar code from its 2D image. This technique enables the reconstruction of a higher quality bar code sample, and an efficient bar code reading algorithm in the following cases:

- when the bar code image is acquired with a low resolution camera (or webcam)
- when the bar code image is acquired at a distance (in both cases, bars are often not recognizable due to aliasing and blur.)
- when the bar code image is degraded with scratches.

In all of the above cases, some bars of the code may not be distinguishable in some or all of the scan lines. The proposed algorithm uses start and end points, and the tilt angle to recognize the location and width of bars. It collects the available start and end points from each scan line to a vector, and finds the best fit. Experimental results show that the proposed technique effectively produces high resolution bar code image from a low-resolution or low quality sample. The extension to 2-D bar code reading is discussed for PDF417 case.

References

[1] Kuroki, M., Yoneoka, T., Satou, T., Takagi, Y., Kitamura, T., Kayamori, N.: ‘Bar-code recognition system using image processing’, *6th Int. Conf. on Emerging Technologies and Factory Automation,* pp. 568-572, 9-12 September 1997.
[2] Liu, S., Liao, H., Chen, L., Tyan, H., Hsieh, J.: ‘Camera based bar code recognition system using neural net’, *Int. Conf. on Neural Networks*, vol. 2, pp. 1301-1305, 25-29 October 1993.
[3] Normand, N., Viard-Gaudin, C. ‘A two-dimensional bar code reader’, 12th Int. Conf. on Pattern Recognition, vol. 3, pp. 201-203, October 9-13, 1994.
[4] Joseph, E., Pavlidis, T.: ‘Bar code waveform recognition using peak locations”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, June 1994.
[5] Bailey, D.G.: ‘Super-resolution of bar codes’, Journal of Electronic Imaging, pp. 213-220, vol. 10, 2001.
[6] http://www.adams1.com/pub/russadam/upccode.html
[7] http://www.pdf417.com
[8] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: ‘Numerical Recipes in C’, Cambridge University Press, 1992.