BAR CODE RECOGNITION IN HIGHLY DISTORTED AND LOW RESOLUTION IMAGES

Ramtin Shams and Parastoo Sadeghi

Research School of Information Sciences and Engineering (RSISE)
The Australian National University
Canberra ACT 0200 Australia
{ramtin.shams};{parastoo.sadeghi}@anu.edu.au

ABSTRACT

In this paper, we present a novel approach to detection of one dimensional bar code images. Our algorithm is particularly designed to recognize bar codes, where the image may be of low resolution, low quality or suffer from substantial blurring, de-focusing, non-uniform illumination, noise and color saturation. The algorithm is accurate, fast, scalable and can be easily adjusted to search for a valid result within a specified time constraint. Our algorithm is particulary useful for real-time recognition of bar codes in portable hand-held devices with limited processing capability, such as mobile phones.

***Index Terms*—** Bar codes, Feature extraction, Image segmentation, Pattern recognition, Peak Detection

1. INTRODUCTION

1.1. Motivation

Bar codes are still the prevalent mechanism of encoding of machine readable information on most products and services. Today, a wide range of general purpose hand-held devices, such as mobile phones, come with an optical imaging system. Enabling these general purpose hand-held devices with the capability to recognize bar codes is a cost-effective alternative to conventional bar code scanners. The hardware is readily available to billions of people. Combined with new services, this can revolutionize everyday shopping experience, including in-store price check, retrieving product information, access to product reviews, locating similar products and services, and on-the-spot price comparison.

Detecting bar codes from images taken by general purpose hand-held devices is particularly challenging due to limitations of the integrated imaging system and the processing capabilities of the device. These devices often have lower quality lens systems and lower resolution imaging circuitry compared to dedicated digital cameras. The optical system is often not designed for taking pictures in close proximity of the lens, which results in de-focused and poorly illuminated images with saturated colors. These limitations often mean that using conventional thresholding methods cannot produce a suitable pattern for recognition of bar codes. An algorithm that can invariably be implemented on a wide range of hand-held devices must take these limitations into consideration.

1.2. Previous Work

Conventional bar code readers project a laser beam raster over the bar code area and measure the amount of reflected light from the alternating bar and space patterns that constitute the bar code. The resulting waveform is then processed in order to determine the width of each module (a bar or a space). A comprehensive introduction to bar code information theory and the principles of decoding can be found in [1].

An image processing framework for detection of 2D bar codes is presented in [2]. The proposed algorithm requires a code density (length of the smallest bar code module) of at least three pixels.

Extensive theoretical work by Joseph and Pavlidis formulates the problem of 1D bar code detection using peak locations from a distorted signal, which results from scanning a bar code using a laser beam scanner [3], [4]. The methods are suitable for laser beam scanners and apply to continuous waveforms or, where a continuous waveform can be restored from sampled data. An important assumption in [3], [4] is that the major source of distortion in the waveform is convolution with the point spread function (PSF) of the scanning device and the effect of non-uniform illumination is not considered.

An algorithm for estimating module widths is presented in [5] by extracting codewords directly, from the intensity map. Local extrema are used as estimations of module centers and high-curvature points as estimates of module midpoints. The algorithm seems to be able to decode symbols with a code density of four pixels or more.

Conventionally, raster scanning has been used to obtain an estimation of the bar code's ideal bi-level representation, emulating the operation of a laser beam scanner. An alternative approach using the Hough transform [6] is suggested by [7], where the distance of parallel lines, forming the bar code, from the origin can be calculated by writing the line equation as

$$\rho = x\cos\theta + y\sin\theta, \tag{1}$$

where ρ is the distance from the origin and θ is the angle of the line normal. This method is sensitive to perspective distortions and also, where the bar code is not printed on a flat surface, for example on plastic bags. The effect of Hough transform in obtaining the bi-level waveform is akin to averaging bar code intensities across the height of the bar code. That explains why the method exhibits some resistance to noise but is not suitable in the presence of non-uniform distortions such as perspective or lighting.

1.3. Approach and Contributions

In this paper we present a novel algorithm for detection of bar codes in low resolution, highly distorted and noisy images, where the code density may be less than two pixels. Such an image is depicted in Fig. 1. The length of EAN13 [1] bar code displayed in the image is 147 pixels, which results in a code density of 1.54 pixels. The

Fig. 1. A low resolution, highly distorted and noisy bar code

image is dark, highly distorted and suffers from a low quality JPEG compression, but can be successfully decoded using our algorithm. The algorithm achieves sub-pixel accuracy by dividing edge pixels between the two adjacent modules.

We use the conventional raster scanning method to retrieve a sequence representation of the ideal bi-level image. This method is susceptible to noise. We address the noise removal problem by applying a variable ripple threshold, when we estimate the location of sequence local extrema. The peak and edge detection techniques discussed in [3] are not readily applicable to the class of images, we consider in this paper, where the length of a module is less than two pixels. We introduce an algorithm that scales the height of peaks and valleys and uses a sliding threshold to estimate the location of edges. This also allows us to successfully decode bar codes in the presence of non-uniform lighting, which is a major source of distortion in the images that we consider.

The edge to edge and extremum to extremum distances are stored in a feature vector. The feature vector is then compared with pre-computed feature vectors of the ideal bi-level signal. A least square measure is used to decode the input sequence by selecting an ideal signal, whose feature vector is closest to the estimated sequence. We also prove a lemma for early detection of a class of sequences, which improves the speed of the algorithm.

The algorithm was implemented for EAN13 type of bar codes, which belongs to the more challenging class of 1D symbologies, known as delta codes [1]. This is due to variable width of bar and space modules and the distortion effect of wider modules on the extrema locations of neighboring narrow modules. The method can be applied to binary symbologies such as Code39 [1].

2. BAR CODE RECOGNITION

In the following sections we provide a description of the general bar code recognition problem by introducing five major components of our bar code recognition methodology with an emphasis on the scanner and decoder components, which contain the bulk of our contributions.

2.1. Preprocessor

The preprocessor acquires the input from the imaging device. This is often in the form of a compressed JPEG stream. The input stream is then decompressed and converted to a gray-scale intensity map. Assuming a 24-bit input image, we have

$$i[m,n] = 1 - \frac{r[m,n] + g[m,n] + b[m,n]}{3 \times 255}, \tag{2}$$

where r, g and b are the red, green and blue components of the input signal, respectively and $i[m,n]$ is the gray-scale intensity at location $[m,n]$ in the image. $i[m,n]$ is scaled between 0 and 1, where 0 denotes a white pixel and 1 a black pixel. We prefer this representation of black and white pixels, as it will associate bars with peaks and spaces with valleys in later stages of the algorithm. Large images (>2MP) are sampled down to reduce the processing cost of often time-consuming region of interest (ROI) detector. Once the ROI's are detected, the original size image is used by the segmenter.

2.2. ROI Detector

An ROI detector is used to find areas within the input image, where there is a chance that a bar code may be found. The directional nature of 1D bar codes is utilized for this purpose by computing a gradient map of the input image and selecting areas, where the gradient phase exhibits some directional characteristics [2].

The magnitude and phase of image gradient are calculated using 3x3 Sobel masks:

$$S_h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \tag{3}$$

$$S_v = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \tag{4}$$

$$G_h = I \otimes S_h, \tag{5}$$

$$G_v = I \otimes S_v, \tag{6}$$

where S_h and S_v are the horizontal and vertical Sobel masks, G_h and G_v are horizontal and vertical gradient maps, respectively and $\otimes$ is the convolution operator.

The gradient maps are more useful in terms of magnitude G_M and phase G_Φ components

$$g_M[m,n] = \sqrt{g_h[m,n]^2 + g_v[m,n]^2}, \tag{7}$$

$$g_\Phi[m,n] = \arctan \frac{g_v[m,n]}{g_h[m,n]}, \tag{8}$$

$$g_h[m,n] \in G_h, g_v[m,n] \in G_v,$$

where $g_M[m,n]$ and $g_\Phi[m,n]$ are the elements of G_M and G_Φ, respectively.

ROI's are formed by tessellating the image with a fixed block size and connecting image blocks that exhibit a strongly modal gradient histogram in some direction.

2.3. Segmenter

The segmenter detects bar code boundaries and adjusts the orientation. The orientation is already calculated by the ROI detector. The

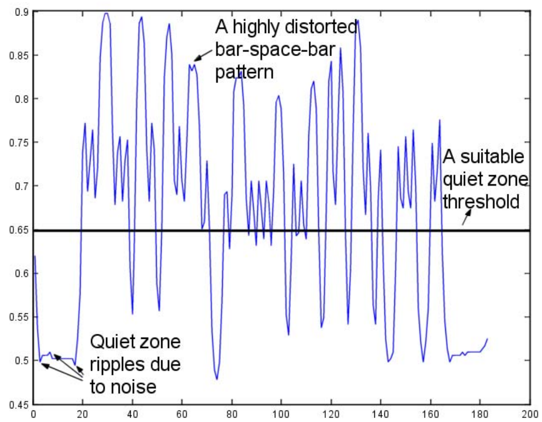

Fig. 2. The intensity sequence of a selected row of the bar code image shown in Fig. 1.

segmenter, rotates the image such that bar code modules are parallel to the vertical axis. This simplifies the scanning process.

In order to find the boundaries, we search for horizontal quiet zones (large areas of background around bar code modules). For example, for EAN13 symbology the minimum quiet zone width must be $9d_c$, where d_c is the code density. At this stage the code density is unknown and we assume the smallest possible code density that can be detected by our algorithm (1 pixel) in order to ensure the most accurate results. This will result in detection of invalid quiet zones inside the area of a bar code for larger bar codes. We can quickly discard these abrasions by counting the number of extrema in the scanning phase.

Due to noise and illumination distortions, the quite zone may exhibit variations in the sequence that resemble narrow modules of a high density bar code. Fig. 2 shows this situation, where a bar-space-bar pattern is similar to noise patterns in the quiet zone. This suggests that using a ripple threshold to remove noise from quiet zone may also wash away important extrema information pertaining to narrow modules. To address this problem we need to estimate a constant threshold T_{qz} that cleanly cuts the quiet zone but does not affect low height peaks.

The output of ROI detector is an image mostly occupied by the bar code and the surrounding quiet zone. Empirically, we have

$$0.5 < \frac{A_{qz} + A_{\text{space}}}{A_{\text{ROI}}} < 0.65, \tag{9}$$

where A_{qz} is the area occupied by the quiet zones, A_{space} is the area occupied by space modules and A_{ROI} is the total area of the region of interest. Since, we approximately know the relative area occupied by the background, we can select an appropriate intensity level for T_{qz} from the cumulative intensity histogram (CIH) [8]. For each intensity level L, CIH provides the number of pixels, whose intensity is not bigger than L. We choose T_{qz} such that $H_{\text{cumulative}}[T_{qz}] = 0.55$. Then we replace pixel intensities below T_{qz} level with T_{qz}. This method allows us to eliminate background intensity variations and the unwanted waveform extrema and improves the robustness with which the quiet zones can be detected.

2.4. Scanner

The scanner converts the input image segment into a discrete one-dimensional sequence and creates a feature vector, which is used later to decode the symbols. We use the raster scanning technique, which produces the sequence by tracing one or more lines across the image and registers intensity information as sequence values.

The ideal sequence can be written as a series of step functions

$$b[n] = k \sum_{i=1}^{N} (-1)^{i-1} u[n - n_i], \tag{10}$$

where $u[n]$ is the discrete unit step function, N is the number of modules in a bar code, n_i is the location of i^{th} edge and k represents a constant height.

In practice due to PSF distortion, illumination distortion, sampling errors and noise the resulting sequence deviates from the ideal flat bi-level signal as shown in Fig. 2.

The scanner estimates the code density and the sequence's local extrema. The location and number of local extrema are important features. The location of local extrema changes due to distortion and unwanted local extrema may be introduced due to noise. A variable ripple threshold T_{ripple} is used to remove noise. T_{ripple} limits detection of further extrema in the vicinity of an already selected extremum and within T_{ripple} distance of the extremum value. The minimum T_{ripple} that results in detection of a correct number of extrema that matches the symbology profile is chosen.

Our scanner module estimates the average code density (as seen by the imaging device). The object's orientation and non-planar shape may cause the code density to vary across the imaged bar code. We use extrema locations to estimate codeword boundaries. A scaling factor is then applied locally to each codeword to normalize and compensate for the effect of perspective and imaging non-planar bar code labels (e.g. bar codes printed on cans).

Another important feature for our detection algorithm is edge-to-edge distances. This is a technique used by laster beam scanners to remove the effect of uniform ink spread distortion introduced in printing bar codes [1]. It serves a similar purpose here, since the effect of non-linear lighting in the vicinity of a codeword is akin to leaking of module edges into neighboring regions. Edge-to-edge distance (between pairs of falling and rising edges) shows some invariance to this effect.

Some of the bar codes that we consider may have a code density of less than two pixels. Sometimes there are $1-2$ pixels between a neighboring peak and valley. We normalize every two neighboring extrema locally, such that the peak is normalized to 1 and the valley to 0. Other pixels in between are scaled such that they retain their relative value compared to the extrema. That is,

$$\hat{s}[n] = \frac{s[n] - \min\left(s[x_i], s[x_{i+1}]\right)}{\mid s[x_i] - s[x_{i+1}] \mid}, n \in [x_i, x_{i+1}], \tag{11}$$

where $s[n]$ is the scanned sequence, $\hat{s}[n]$ is the normalized sequence and x_i is the location of the i^{th} extrema.

We define a threshold, called normalized extremum threshold T_{ne} to estimate the edge. We assume that all pixels above T_{ne} belong to the neighboring bar and those below to the space, except for (up to) two pixels closest to the threshold. We split these pixels between the two neighboring modules depending on their distance to the threshold. For example, a pixel that lies on the threshold itself is equally split between the bar and the space and contributes half a pixel to the width of each module. This allows us to achieve sub-pixel accuracy, where there might be only one or two pixels between the neighboring extrema. T_{ne} is varied between 0.2 and 0.8.

By selecting appropriate values for T_{ripple} and T_{ne}, we create a feature vector $\hat{V}_k$ for each codeword c_k consisting of peak-to-peak , valley-to-valley and edge-to-edge distances.

2.5. Decoder

Let $F_k = \{V_{k1}, V_{k2}, ..., V_{km}\}$, be the set of feature vectors V_{ki} calculated for each ideal bi-level signal b_{ki} at k^{th} codeword location and $i \in [1, m]$, where m is the number of possible encodings at k^{th} location. The decoding problem can be defined as finding a symbol, whose feature vector $\hat{V}_k$ satisfies

$$\min \parallel V_{ki} - \hat{V}_k \parallel, V_{ki} \in F_k \quad (12)$$

Hence, the decoding problem is solved by finding the symbol, which is closest to the estimated feature vector. This process is repeated for each combination of T_{ripple} and T_{ne} and could be time consuming. We show that the performance of this process can be improved.

$Lemma$:

The distance between a real vector $V = [v_1, ..., v_n], v_i \in \mathbb{R}$ and an integer vector $\tilde{V} = [\tilde{v_1}, ..., \tilde{v_n}], \tilde{v_i} \in \mathbb{N}$ in an n-dimensional space, is minimal if $\tilde{v_i} = \lfloor v_i + 0.5 \rfloor, i \in [1, n]$, where $\lfloor a \rfloor$ is the smallest integer value not bigger than a.

$Proof$:

$$if\ V' = [v'_1, ..., v'_n], v'_i \in \mathbb{N} \quad (13)$$

$$|\tilde{v_i} - v_i| \leqslant \frac{1}{2}, |\tilde{v_i} - v'_i| \geqslant 1, \tilde{v_i} \neq v'_i \quad (14)$$

$$|\tilde{v_i} - v'_i| - |\tilde{v_i} - v_i| \geqslant \frac{1}{2} \quad (15)$$

$$|v_i - v'_i| \geqslant \frac{1}{2} \quad (16)$$

$$|v_i - v'_i| \geqslant |\tilde{v_i} - v_i| \quad (17)$$

$$\sum_{i=1}^{n}(v_i - v'_i)^2 \geqslant \sum_{i=1}^{n}(\tilde{v_i} - v_i)^2 \quad (18)$$

$$D_{V',V} \geqslant D_{\tilde{V},V} \quad (19)$$

Therefore, in order to improve the performance of the decoding algorithm, we can first calculate $\tilde{V}_k$ by rounding the elements of $\hat{V}_k$. If $\tilde{V}_k = V_{ki}, V_{ki} \in F_k$ then V_{ki} satisfies condition (12) and we no longer need to calculate the distance of estimated feature vector from every possible symbol at each codeword location.

Our decoding algorithm may result in a sequence of symbols that do not match the symbology specification (e.g. invalid checksum). We throw away these results and continue searching for a valid response up to a specified time limit and/or when sufficient number of matching results with a distance error below a certain threshold are found.

3. RESULTS

We tested our algorithm against a database of EAN13 bar code images taken by an NEC 616 mobile phone. The mobile phone has a low resolution of 352x288 pixels. The images were taken in various lighting, orientation, distance and perspective conditions. Our algorithm was able to successfully decode 47% of the test subjects. The results showed 57% improvement over a popular commercial bar code reader (by Axtel Inc. [9]), which was only able to recognize less than 30% of the images.

4. FUTURE WORK

The algorithm can be adapted for 2D bar code symbologies. The performance of the algorithm can be improved by estimating an optimum ripple threshold T_{ripple} and an optimum normalized extremum threshold T_{ne}.

5. CONCLUSIONS

In this paper, we presented an image processing framework for recognition of 1D bar codes. We presented new algorithms for scanning and decoding of low resolution bar codes under significant distortion, noise and under-sampling conditions. We also proved a lemma to improve performance of the bar code decoder and compared the results with a commercial bar code reader.

6. REFERENCES

[1] T. Pavlidis, J. Swartz, and Y. P. Wang, “Fundamentals of bar code information theory,” *Computer*, vol. 23, no. 4, pp. 74–86, Apr. 1990.

[2] E. Ouaviani, A. Pavan, M. Bottazzi, E. Brunelli, F. Caselli, and M. Guerrero, “A common image processing framework for 2d barcode reading,” in *Proc. Seventh Int. Conf. on Image Processing and Its Applications (Conf. Publ. No. 465)*, vol. 2, July 1999, pp. 652–655.

[3] E. Joseph and T. Pavlidis, “Bar code waveform recognition using peak locations,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 630–640, June 1994.

[4] ——, “Waveform recognition with application to bar codes,” in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics. 'Decision Aiding for Complex Systems, Conference Proceedings.*, vol. 1, 1991, pp. 129–134.

[5] H. I. Hahn and J. K. Joung, “Implementation of algorithm to decode two-dimensional barcode PDF-417,” in *Proc. IEEE 6th Int. Conf. on Signal Processing*, vol. 2, Aug. 2002, pp. 1791–1794.

[6] P. V. C. Hough, “A method and means for recognizing complex patterns,” in *U.S. Patent No. 3,069,654*, 1962.

[7] R. Muniz, L. Junco, and A. Otero, “A robust software barcode reader using the hough transform,” in *Proc. Int. Conf. on Information, Intelligence and Systems*, Oct. 1999, pp. 313–319.

[8] B. K. P. Horn, *Robot Vision*. The Massachusetts Institute of Technology: MIT Press, 1986.

[9] *Axtel Home Page*. http://www.axtel.com/: Axtel Inc., 2006.