Classification and Decoding of Barcodes: An Image Processing Approach

R. Dinesh, R. G. Kiran and M. Veena

Abstract Barcodes are being widely used in many fields of applications of great commercial value, which provide a means of representing data in machine readable format. Various symbologies exist to map the data into barcodes. Image based barcode readers provide many advantages over laser scanners in terms of orientation independence, image archiving and high read rate performance even when barcodes are damaged, distorted, blurred, scratched, low-height and low-contrast. The availability of imaging technology provides a platform for decoding barcode rather than the use of the conventional laser scanner which is lack of mobility. In this paper, image based technique for classification of the given 1-D barcode into respective symbology and its decoding have been proposed. The proposed method first localizes the barcodes and subsequently, a classifier which classifies the given 1-D barcode into respective symbology is applied. Decoding is then performed based on the specification of the symbology. To establish the superiority of the proposed approach in classification and decoding of barcodes, we have conducted extensive experiments on various datasets, both standard as well the images captured from low resolution cameras (mobile camera). The results reveal the superiority of the proposed method in terms of better read rate for standard and even for blurred barcode images.

Keywords Barcode · Image acquisition · Localization · Tree classifier · Decoding standards

R. Dinesh (✉) · R. G. Kiran · M. Veena
HCL Technologies, Bangalore 560068, Karnataka, India
e-mail: dinesh.ramegowda@hcl.com

R. G. Kiran
e-mail: kiran.gachchi@hcl.com

M. Veena
e-mail: veena.m@hcl.com

P. P. Swamy and D. S. Guru (eds.), *Multimedia Processing, Communication and Computing Applications*, Lecture Notes in Electrical Engineering 213,
DOI: 10.1007/978-81-322-1143-3_24,

1 Introduction

A barcode is an optical machine-readable representation of data relating to the object to which it is attached. There are different types of barcodes, often referred to as barcode symbologies [1], but all have the common purpose of encoding a string of alphanumeric information as a set of bars and spaces of varying widths printed on a product. Barcodes can be one dimensional or two dimensional. 1-D barcodes are referred as linear barcodes as they are made up of a collection of bars and spaces frequently known as *elements* or *modules*. In these barcodes, the height of the barcode provides added redundancy to the system if part of the symbol becomes damaged or occluded. Processing the barcode to find the data encoded involve several stages.

- Barcode localization: Searching the image to find the barcode containing region
- Decoding (Reading): Extracting the information encoded in the barcode.

2 Related Work

Several methods have been proposed for localization and decoding the barcodes. Alexander [2] used DCT (Discrete Cosine Transform) properties to distinguish bar code from other texture. The precondition is that a bar code has to occupy at least 10 % of the whole image. Because the weighting matrix coefficients are not determined self-adaptively, the robustness of the result is not as good as desired. This approach also brings a serious disadvantage of whole image being processed more than once. Hence hinders its use in real time applications. Jain and Chen [3] suggested the barcode localization using multi-channel Gabor filter.

Muniz, Junco and Otero [4] investigated using the Hough transform for locating barcodes. It leads to a computationally expensive and time consuming approach since Hough transform takes more time to detect lines in the image. Sheng et al. [5] proposed a new method for barcode localization and recognition to overcome the disadvantage of using Hough transform. Initially, Sobel edge detector is applied to remove irrelevant image background and preserve barcode edges. Later, angle of fitting lines is computed by using the turning points. The area of lines with the same or similar angles is considered the area of bar code. If the angle is not 90 degrees with respect to the horizontal direction, rectify the bar code by bilinear interpolation. Juett and Xiaojun Qi [6] proposed non-traditional localization algorithm using a bottom hat filter. In this approach, directional opening was performed at different orientation followed by density region analysis. All these approaches mainly focus on localization of barcode rather than decoding and add more overhead in terms of computations. Hence they are time consuming. To overcome the limitation of existing approaches, we use a simple approach for localization based on edge detection and morphological closing. Chai and Hock

[7] proposed a vision based technique for locating and decoding EAN13 barcodes captured from digital cameras. A block based approach is used to localize barcodes. An image is divided in to 32*32 blocks and finds each block's angle. Blocks with the same angles are selected to form a bar code area. Decoding is performed by obtaining bar widths. Only EAN13 barcodes were considered. Liyanage [8] described an edge detection based method to classify and decode only a subset of symbologies such as EAN13 and Code39. Localization of barcode in an image is performed manually. The barcode classifier used in this method fails for other symbologies. Wachenfeld et al. [9] used an image analysis and pattern recognition methods which rely on knowledge about structure and appearance of 1D barcodes. Symbologies such as UPC-A/EAN-13/ISBN-13 were considered for decoding using this approach.

The existing methods quoted for decoding the barcodes can decode only few symbologies. In this paper we present classification and decoding of most of the existing 1-D barcode symbologies. We first perform the localization of the barcode in the input image i.e. to find the location of the four corners of the barcode. And classify the localized barcode into respective symbology. Finally, decoding is then performed based on the specification of the symbology.

3 Proposed Methodology

This section presents the proposed method for Barcode localization and decoding of 1-D barcodes. The architecture of the proposed method for barcode localization, classification and decoding is given in Fig. 1. In the initial stage given image is processed to locate the region containing the barcodes. Subsequently a classifier is applied to find the symbology of the localized barcode. Later, decoding is performed according to the specification of its respective symbology.

3.1 Barcode Structure

A typical 1-D barcode image is shown in Fig. 2. It consists of following parts: a quiet zone, a start character, data characters, an optional check digit, a stop character, and another quiet zone.

Quiet zones are non-printed zones immediately before and after the barcode. It is recommended that the quiet zone be at least 10 times the narrow bar width. A quiet zone smaller than this, can make the barcode unreadable. Start/Stop character is a pattern of bars and spaces that provide the scanner with start and stop reading instructions. An optional check digit included within a barcode whose value is used to perform a mathematical check that ensures the accuracy of the read. It is placed immediately after the barcode data. Length of label includes left and right quiet zones. A read cannot be made if the quiet zones are not large enough. The

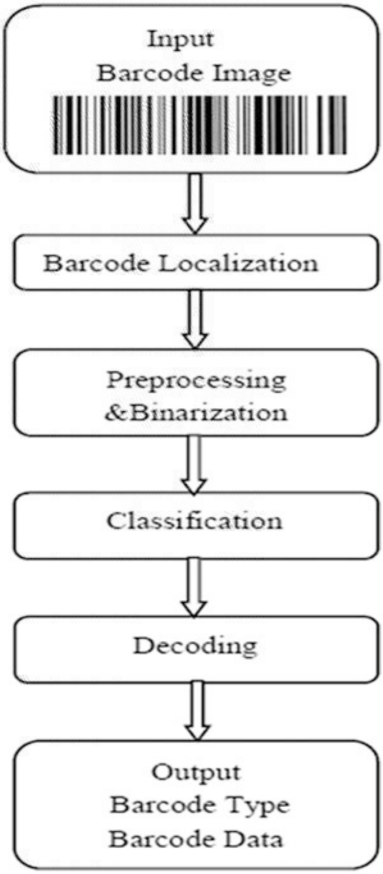

Fig. 1 Stages of proposed methodology

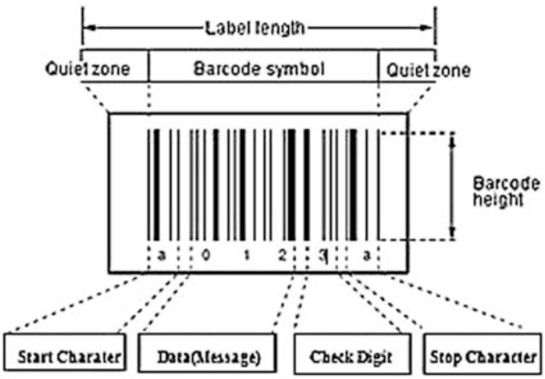

Fig. 2 Typical 1-D barcode and its parts

height of the barcode provides added redundancy to the system if part of the symbol becomes damaged or occluded.

3.2 Barcode Localization

Barcode location within the given image is obtained by performing Barcode localization. Figure 3 shows an image containing the barcode. Initially edge detection is applied over the image to remove irrelevant background. First order gradient operator such as Sobel, shown in Fig. 4 is used to obtain the edges. Barcode area is retained since it is made up of black and white bars. The transitions from black bar to white bar or vice versa results in an edge in the image. The result of edge detection after applying the Sobel masks is shown in Fig. 5.

In the next stage, morphological closing operation is performed on the resultant image of edge detection. Closing expands the white regions in the image without altering the regions that are already white. A rectangular structuring element used for closing operation needs to be as wide as widest bar in the image. (In this work we have used the structuring element of size 15 × 10). The result of closing operation is shown in Fig. 6. This effectively highlights the barcode region. Other regions containing the dark on light patterns are also highlighted, which will be removed in the later stages.

The regions obtained by closing operation are then labeled using connected component labeling. Since closing operation retains other regions along with the barcode, such regions are removed in this stage. For each labeled region, the

Fig. 3 Image containing the barcode

Fig. 4 Sobel edge detection masks

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Fig. 5 Result of edge detection

Fig. 6 Result of closing operation

boundary coordinates in X and Y directions are determined. From these coordinates width and height of each region is calculated. Since the barcode region appears rectangular or square in most of the images, other non-rectangular regions are removed by thresholding based on width and height. Rectangular region are thus retained. The resultant image obtained after this stage is shown in Fig. 7.

The resultant image may contain non barcode rectangular regions. To locate exact barcode region, the intensities are projected on to a horizontal profile. The barcode region will result in abrupt edges in the profile because of the black and white bar pattern as shown in Fig. 8. Thus the barcode region in the image is located.

3.3 Preprocessing and Binarization

Preprocessing is an essential step in any image processing application and pre-processing can significantly increase the reliability of the vision system. Several filter operations which intensify or reduce certain image details enable an easier or

Fig. 7 Result of thresholding regions based on width and height

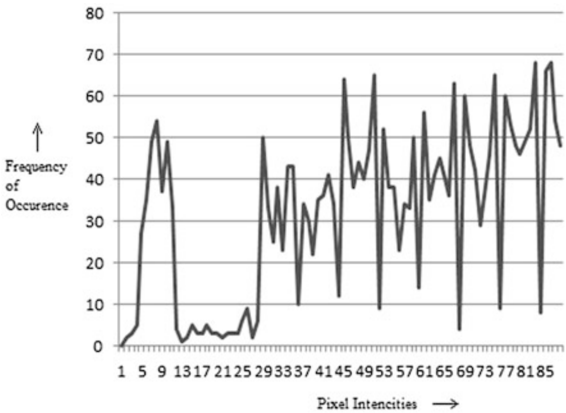

Fig. 8 Horizontal profile of a barcode region

faster evaluation. Barcode preprocessing is the key step for the accurate recognition of the given barcode.

We use median filter and Gaussian filter for removal of noise in the barcode image. Median filter is a non-linear filter which replaces each entry by the median of neighboring entries. This filter performs well in denoising the salt and pepper noise, Speckle noise by preserving the sharp edges. Gaussian filter removes the impulse noise. The sigma values are varied to improve the denoising process. Later Binarization is performed automatically using Otsu's histogram shape-based image thresholding. This approach assumes that the image to be thresholded contains two classes of pixels or bi-modal histogram(e.g. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread is(intra-class variance) minimal. We assume the barcode image is corrected for skewness.

3.4 Classification

All barcodes are constructed from a series of bars and intervening spaces. The relative size of these bars and spaces and the number of them is decided by the specification of the symbology (or barcode type) which is being used. There are a number of 1-D barcode symbologies in common use. Such symbologies are shown in Fig. 9. Each symbology differs in the way data is encoded and often also in the type or amount of data encoded [1]. Generally, only one symbology is chosen for a particular application.

Given a 1-D barcode image, Tree classifier is applied to classify it into a specific symbology. Such classification is done based on its widths of bars and spaces appearing in start and/or stop patterns [1]. Following Fig. 9 shows the Tree classifier for 1-D barcode (numbers on arrow marks shows the start and/or stop pattern of the barcode). Each 1-D barcode has unique start and/or stop pattern. The initial and final bars and spaces contribute to form these patterns. From the widths of bars and spaces determined, we compare them to the unique patterns of each symbology as in [1]. Since code 128, code-39 and Codabar has the largest start and/or stop pattern among all other barcode symbologies, these patterns are compared first. Later code-93, Interleaved 2 of 5 and code 11 are compared. Finally MSI and UPC are compared.

The numbers shown beside the arrows represent the widths of bars and spaces as multiples of the *module*. For example, consider the codabar barcode symbology made up of alternative bars and spaces. Given the module width as 2 pixels, the start and stop pattern for codabar is 1122121 refers to 1*2 = 2 pixels wide bar,

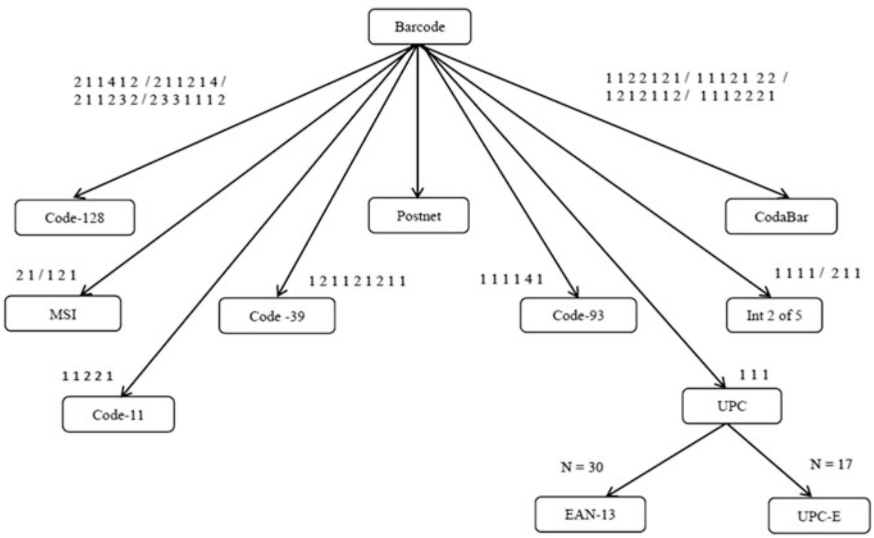

Fig. 9 Tree-classifier for 1-D barcode based on start and/or stop pattern (*for Postnet barcode, height of the barcodes are considered*)

1*2 = 2 pixels wide space, 2*2 = 4 pixels wide bar, 2*2 = 4 pixels wide space, 1*2 = 2 pixels wide bar, 2*2 = 4 pixels wide space and finally 1*2 = 2 pixels wide bar. Similar calculations are followed for all types of symbologies.

For UPC family of barcodes, an additional measure in terms of number of bars is considered for classification. This is in regard of the start and/or stop pattern being same for EAN-13 and UPC-E symbologies. This is shown as 'N' in Fig. 9. For EAN-13 symbology N = 30 and for UPC-E symbology N = 17.

Postnet symbology differs from all other symbologies as it encodes the data by varying heights of bars rather than varying widths of bars and spaces. Hence given barcode can be classified as Postnet by determining the height of bars [1].

3.5 Decoding

Decoding is the process of converting the bar and space patterns of the barcode into data characters. Our barcode decoding algorithm analyzes the barcode using multi scan line approach. Several scan lines contained in the detected barcode area are taken. This ensures that data loss in one scan line due to severe blurring, occlusion etc. can be recovered from decoding other scan lines. Each scan line results in decoded data characters. To find out the correct encrypted characters among these, at each position the character which is resulted maximum number of times by decoding several scan lines is taken as decode data.

Decoding of barcodes is performed according to the standard decoding algorithms [10] based on the type of the barcode detected. The decoding process requires the widths of the bars and spaces to be known. The thickness of bars and spaces in comparison with each other determines the digit they represent. The thinnest bar/space represents *module* or *element*. Estimate the widths of each bar and space of the barcode after start character till before the stop character. Get the data of the barcode from the respective barcode encoding table [1] using the obtained width of bars and spaces.

4 Experimental Results

We took images from various datasets, both standard as well the images captured from low resolution cameras (mobile camera). We conducted extensive experiments on clean and noisy barcode images. The noisy barcode images are generated by adding noise such as salt and pepper, Gaussian and Speckle noise.

The results of decoding few barcode images are presented below. Both the success and failure cases are listed.

Barcode Image	Noise added	Result of decoding
A123$+56B	Salt and pepper noise Noise density = 0.05	Symbology: codabar Data: A123$ + 56
123-4530	Salt and pepper noise Noise density = 0.05	Symbology: code-11 Data: 123-4530
ABCD123%	Salt and pepper noise Noise density = 0.05	Symbology: code-39 Data: ABCD123 %
123456	Salt and pepper noise Noise density = 0.05	Symbology: code-93 Data: 123456
(12)34barcode	Salt and pepper noise Noise density = 0.05	Symbology: code-128 Data: 1234 barcode
5 032423 434303	Salt and pepper noise Noise density = 0.05	Symbology: EAN-13 Data: 5032423434303
1234456781122475847587 1234456781122475847587	Salt and pepper noise Noise density = 0.05	Symbology: interleaved 2 of 5 Data:
12345	Salt and pepper noise Noise density = 0.05	Symbology: MSI Data: 12345
	Salt and pepper noise Noise density = 0.05	Symbology: postnet Data: 12345

(continued)

(continued)

Barcode Image	Noise added	Result of decoding
A123$+56B	Gaussian noise Mean: 0 Variance: 0.007	Symbology: codabar Data: A123$+56
123-4530	Gaussian noise Mean: 0 Variance: 0.007	Symbology: code-11 Data: 123-4530
ABCD123%	Gaussian noise Mean: 0 Variance: 0.007	Symbology: code-39 Data: ABCD123 %
123456	Gaussian noise Mean: 0 Variance: 0.007	Symbology: code-93 Data: 123456
(12)34barcode	Gaussian noise Mean: 0 Variance: 0.007	Symbology: code-128 Data: 1234 barcode
5 032423 434303	Gaussian noise Mean: 0 Variance: 0.007	Symbology: EAN-13 Data: 5032423434303
1234456781122475847587 1234456781122475847587	Gaussian noise Mean: 0 Variance: 0.007	Symbology: Interleaved 2of5 Data:
12345	Gaussian noise Mean: 0 Variance: 0.007	Symbology: MSI Data: 12345

(continued)

(continued)

Barcode Image	Noise added	Result of decoding
	Gaussian noise Mean: 0 Variance: 0.007	Symbology: postnet Data: 12345
	Gaussian noise Mean: 0 Variance: 0.007	Symbology: UPC-E Data: 01234565
	Speckle noise Variance: 0.002	Symbology: codabar Data: A123$+56
	Speckle noise Variance: 0.002	Symbology: code-11 Data: 123-4530
	Speckle noise Variance: 0.002	Symbology: code-39 Data: ABCD123 %
	Speckle noise Variance: 0.002	Symbology: code-93 Data: 123456
	Speckle noise Variance: 0.002	Symbology: code-128 Data: 1234 barcode

(continued)

(continued)

Barcode Image	Noise added	Result of decoding
	Speckle noise Variance: 0.002	Symbology: EAN-13 Data: 5032423434303
1234456781122475847587	Speckle noise Variance: 0.002	Symbology: interleaved 2 of 5 Data:
	Speckle noise Variance: 0.002	Symbology: MSI Data: 12345
	Speckle noise Variance: 0.002	Symbology: postnet Data: 12345
	Speckle noise Variance: 0.002	Symbology: UPC-E Data: 01234565
	None	Symbology: EAN-13 Data: 8901286205863

Our approach fails to decode the following barcode images.

5 Conclusion and Future Work

In this paper, we have proposed an image processing based approach for classification and decoding of barcodes. This method gives a compact and robust solution for decoding and classification of most of the widely used 1-D barcode symbologies. It has been successfully tested on several symbologies. Further work is being done towards classification and decoding of 2-D barcode symbologies. Localization of the image containing several barcodes can also be done as extension to this approach.

References

1. http://www.barcodeisland.com/
2. Tropf A (2006) Locating 1-D barcodes in DCT domain. ICASSP 2(2006), pp 741–744
3. Jain AK, Chen Y (1993) Barcode localization using texture analysis. In: Proceedings of the 2nd international conference on document analysis and recognition pp 41–44
4. Muniz R, Junco L, Otero A (1999) A robust software barcode reader using the hough transform. In: Proceedings of the international conference on information intelligence and systems, pp 313–319

5. Xin-sheng W, Lian-zhi Q, Jun D (2009) A new method for barcode localization and recognition. In: Proceedings of the 5th international conference on image and signal processing, pp 1–6
6. Juett J, Qi X Barcode localization using a bottom hat filter. (http://digital.cs.usu.edu/~xqi/Teaching/REU09/Website/James/finalPaper.pdf)
7. Chai D, Hock F (2005) Locating and decoding EAN-13 barcodes from images captured by digital cameras. In: Proceedings of the international conference on information, communications and signal processing, pp 1595–1599
8. Liyanage J (2007) Efficient decoding of blurred, pitched, and scratched barcode images. In: Proceedings of the 2nd international conference on industrial and information systems
9. Wachenfeld S, Terlunen S, Jiang X (2008) Robust recognition of 1-D barcodes using camera phones. In: Proceedings of 19th international conference on pattern recognition, pp 1–4
10. GS1 (2009) General specification, Version 9.0, Issue 1, Jan 2009