AUTOMATIC REAL-TIME BARCODE LOCALIZATION IN COMPLEX SCENES

Chunhui Zhang, Jian Wang, Shi Han, Mo Yi
Microsoft Research Asia, Beijing, China
{chhzhang, jianw}@microsoft.com

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA
zhang@microsoft.com

ABSTRACT

Many applications exist for automatically finding and reading barcodes in complex scenes with a camera. The key problem is to search barcodes in a complex scene and supply them for a reading subsystem. However, illumination, rotation, perspective distortion and multiple barcodes circumstances make barcode localization difficult. By jointly analyzing texture and shape, we propose a real-time barcode localization method that is robust to the above problems. A user only needs to put a barcode in front of a camera at around 15cm for common web cameras, and the system then automatically locates and decodes the barcode. This paper only focuses at automatic localization. We will demonstrate the complete system live at the conference.
Keywords: Bar codes, Image Processing, Localization

1. INTRODUCTION

Barcode has been widely used to improve the speed and accuracy of computer data entry. But the prevalent barcode readers are devices with laser-scanners or special CCD cameras, which are quite uncommon in our daily life. If barcode reading can be performed with an ordinary camera (e.g. web cam or phone camera), barcode will bring more convenience or even new experience into our daily life.

The key problem is to search barcodes in a complex 3-D scene and supply them for a reading subsystem [2], [3]. As an image analysis task, barcode detection is, in a sense, simpler than edge detection [4] because bars are parallel, and their widths and gaps between them are multiple of a unit bar width. Although this permits the label reading to be insensitive to geometric effects such as label size, orientation, and distance [5], the accuracy and efficiency of edge features deteriorates when a barcode only occupies a small region of a cluttered image or when an image is not well focused. Indeed, the problem of edge interaction has been the main factor that limits the working range of current barcode readers [6]. However, simple enlargement of image size and resolution is not advantageous because it requires better focusing in a wider range, more computing time, more memories, and hence much more resources. From this point of view, two-stage processing[l], [2], i.e., finding barcodes using a region-based analysis of low-resolution image, and then focusing and reading them using a high-resolution one, is more desirable.

Some researchers directly take advantage of gradient information as barcode extraction tool [1], [2]. Ando and Hontani [1] proposed a method of "feature extraction after categorization and projection" for edges, ridges, corners, and vertices. But this method is not robust especially for multiple edge noise appearing around barcode position.

Others consider texture in their detection criteria [3], [7], Jain and Karu [7] have implemented a texture segmentation algorithm in a neural network architecture. The performance of two learning algorithms, the centroid and the gradient descent algorithms was evaluated. Since only texture is considered in this method, barcode location may be confused if similar textures such as words and bars appear in the image.

In many applications, texture and shape are closely related [8]. Single texture feature is not enough for distinguishing image patterns. Space relation description alone can not comprehensively and precisely describe image content either. However, it can enhance image content description and discrimination power. Many retrieval methods use this joint information [9],[10], [11],[12]. We therefore jointly estimate texture and shape to improve the efficiency of barcode localization.

2. BARCODE LOCATION

From above analysis, two-stage processing, i.e., firstly finding barcodes using a region-based analysis of a down-sampled image (low resolution), and then focusing and reading them using the original image, is applied in our method.

In low resolution, the first step is the estimation of the main orientation from edge directions, which is similar to edge histogram [8]. The second step is area pickup with precise orientation calculation and code projection along the bar axis and code axis respectively (Fig. 1).

In high resolution, image pre-processing such as barcode pose regularization, effective part finding and highlights removal are performed before decoding. Fig. 2 is a diagram illustrating the main steps of the barcode localization process.

Fig. 1: Barcode feature analysis

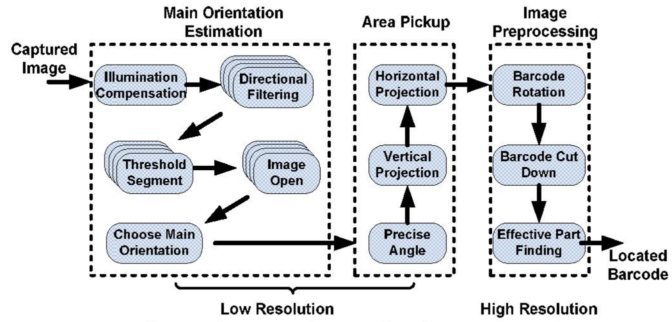


Fig. 2: The Barcode Localization Framework

2.1. Main Orientation Estimation

The most distinctive feature of a barcode is its texture information. Contrast, coarseness and directionality are three main statistic texture descriptors. A barcode's main orientation is realized in four steps: image preprocessing, directional filtering & segmentation, directional image open operation and main orientation determination.

a) Image preprocessing The main task of this step is to remove the shadows on the object and regularize the range of image intensities. We adopt illumination compensation technique in this step to: 1) compensate non-uniform lighting; 2) stretch image intensities to 0-255. Non-uniform lighting is modeled as an additive 2nd-order surface, which is divided from the original image to get a uniformly illuminated image.

b) Directional Filtering & Segmentation Four directional filters are respectively applied to the illumination compensated image to get gradient information at angle 0, 45, 90 and 135. Then binary gradient images are generated through iterative segmentation [13]. The termination condition of segmentation iteration is when the difference between successive threshold values is less than 5 gray levels. Besides, the final threshold value is multiplied by a coefficient in order to get thinner edge. The coefficient taken in PC version is 0.5 in order to reduce background disturb, while it is 0.7 in PDA version in order to get more edges in blurry images.

c) Directional Image Open Operation In the case with complex background, segmentation results still contain many isolated edges in similar orientation. Only barcodes have strong continuity and coherence at a certain orientation. Using image open operator in the same orientation as orientation filters, background noise can be well removed.

d) Main Orientation Estimation At this point, main orientation can be estimated by comparing these four binary images. Orientation filtering can produce distinctive difference between orientation parallel to barcode and orientation vertical to it. We count the number of edges in each direction, and the main orientation of the barcode is the direction has the biggest number of edges.

2.2. Area Pickup

After determining the main orientation, precise barcode region should be picked up using the main orientation binary image. Barcode's continuity and coherence along orientation is still used here. Two steps are taken here: angle calculation, and vertical & horizontal projections.

a) Angle Calculation Precise angle is required for next projection step. Angle can be directly obtained from edges. A barcode has high density in the edge image. The edge image is down sampled at 32 pixel interval, and a density image is produced by counting the number of edges in each local region (Fig. 3-1). A barcode edge (pixel value 1) is searched in a circle centered at the highest density position ***Cen***. Once a barcode edge is detected, the whole line is picked up (Fig. 3-2). If it is a qualified line (longer than MAXLINELENGTH=25), its angle is calculated from its end points. And this angle is considered as the expression of the bar axis orientation.

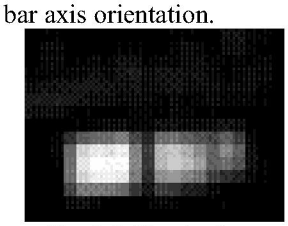


Fig. 3-1: Density Image


Fig. 3-2: Angle count

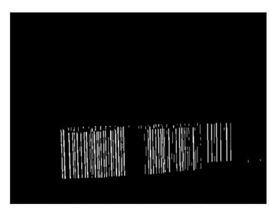


Fig. 3-3: Code axis proj.

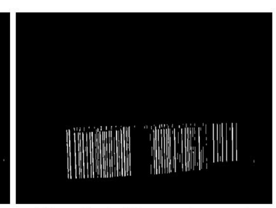


Fig. 3-4: Bar axis proj.

b) Code Axis Projection and Bar Axis Projection Code axis projection array ***CP*** is first made by calculating each edge point's code axis projection. The longest sub-array is regarded as barcode position, whose projected bar is higher than average bar height in code axis projection histogram. The length of the sub-array is taken as the barcode height ***BH*** (Fig. 3-3). Among the qualified points along code axis, bar axis array ***BP*** is made by calculating each edge point's bar axis projection. If the bar axis projected bar is higher than **0.1*BH**, it is regarded as barcode position. Each barcode position is extended OFFSET (20) pixels in two sides to link adjacent barcode position. Thus bars inside one barcode are linked as a continuous sub-array. Because multiple barcodes (from 1 to 3) may occur in the same line, multiple qualified sub-arrays may exist. Sub-array with maximum width is located as the barcode position. Then if other sub-arrays, with width larger than **0.5*BH**, are beside the located barcode, they are merged into the valid barcode position (Fig. 3-4). The barcode width ***BW*** can then be obtained.

2.3. Image Postprocessing

If ***Cen*** is inside the picked up barcode region, the localization result is regarded as a valid one. The barcode region is cut out (Fig. 4-1) from the original image. Further image processing is then carried out. The postprocessing includes image rotation, image cut down and effective part finding. Mirrored images can also be dealt with.

a) Barcode Rotation Angle has been calculated in section 2.2. Barcode region is rotated with bilinear interpolation. The whole barcode image is preserved and blank area is remedied by mirror projection (Fig. 4-2).

b) Barcode Cutout Original barcode width ***BW*** and barcode height ***BH*** can be used to calculate the most effective region of the barcode. It is mostly useful when the barcode has been rotated with a large angle. The center part of size ***BW* BH*** is cut out (Fig. 4-3).

c) Effective Part Finding There are other factors which influence decoding accuracy such as numbers inside the barcode region and highlights area. On the contrary, effective barcode part represents high continuity in vertical orientation. The barcode region is first binarized with image mean T_{mean} as the segmentation threshold. Correlation between adjacent scan lines, $C_{(j,j+1)}$, is calculated as in Equation 1. The effective part is the broadest region whose correlations of the scanning beams are above average correlation (Fig. 4-4).

$$C_{(j,j+1)} = \sum_{i=0}^{i=Width} 1 \left| p_{i,j} < T_{mena}, p_{i,j+1} < T_{mean} \right. \qquad (1)$$

Fig. 4-1 Barcode Region

Fig. 4-2 Rotated Result

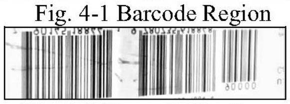


Fig. 4-3 Barcode Cut Down

Fig. 4-4 Effective Part

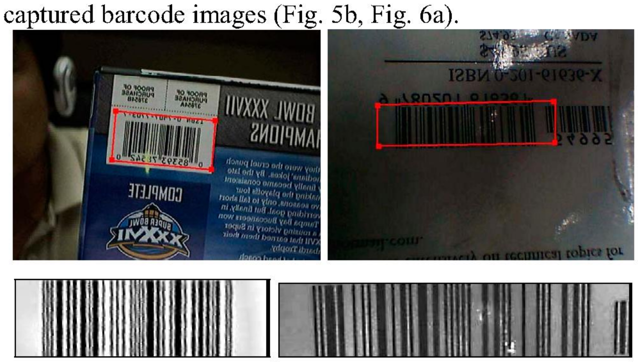
3. EXPERIMENTS

The test images are taken of resolution 640x480(VGA) with different cameras on different platforms. We use a web camera from Creative for Desktop PC, built-in camera of Pocket PC from Lenovo and Smart Phone from Dopod565 for mobile platform. Images of different quality (Fig. 6) in different circumstances are fully tested as below.

3.1. Robustness

a) Complex Background Design As Fig. 5a shows, barcodes are always printed on all kinds of products. Complex cover designs including words and pictures increase the difficulty of barcode reading by computer.

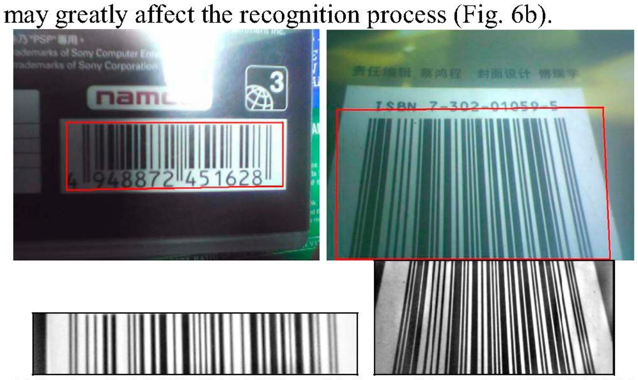
b) Environment Illumination Barcodes may be captured in different environment illumination conditions, which greatly affect the gray levels of bars and spaces in captured barcode images (Fig. 5b, Fig. 6a).



(a)0085393785420 (b)9780201616361

Fig. 5: Complex Background Design(a) and Environment Illumination on WebCam(b)

c) Perspective Deformation A robust barcode recognition system should allow a user to capture a barcode image from a wide range of angles. The perspective effect may greatly affect the recognition process (Fig. 6b).

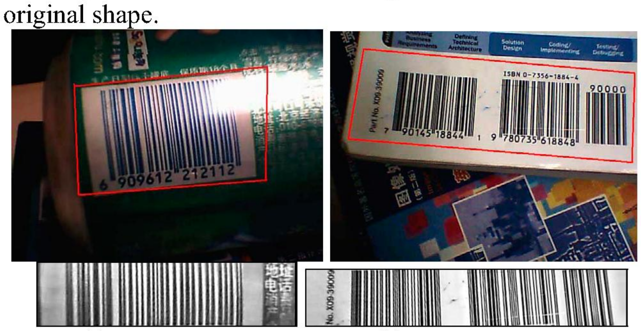


(a)Pocket PC 948872451628 (b) Smart Phone 9787302010593

Fig. 6: Environment Illumination on Pocket PC(a) and Perspective Deformation on Smart Phone(b)

d) Highlights and embedded digits Though Barcode region can be effectively located, highlights and embedded digits influence the decoding efficiency because the decoding process count on the number of bars and their widths (Fig. 7a). Barcode region can not simply be sent to the "Barcode Identification" module without removing this disturbance. "Effective Part Finding" as the last step in "Barcode Localization" accomplishes this task.

e) Curved Surface Barcodes may appear on the cover of cans or bottles as well. Since they have curved surface (Fig. 7b), it is hard for a camera to decode because bars are of different distortion at different areas. "Area Pick Up" can correctly find the curved barcode region and restore it to the original shape.



(a)6909612212112 (b) 9780735618848

Fig. 7: Highlights & Curved Surface (a) and Multiple Barcodes (b)

f) Multiple Barcodes Sometimes, multiple barcodes are listed together. One is of American standard UPC-A, one is International coding rule EAN13, and the other shorter one is price barcode. Without separating barcodes, decoding process will not produce correct results. "Pre-segmentation" and "Multi barcode Separation" is added to "Barcode Identification" for this reason. Clearly, "Barcode Localization" picks up all the barcodes in one image (Fig. 7b).

3.2. Computation Time

We have designed a framework and developed a system to implement the proposed barcode reading algorithm. Under our framework, we can easily build different versions for Desktop PC, Pocket PC and Smart Phone platforms. The Desktop PC has an Intel Pentium 4 2.40GHz processor and 510MB RAM, with Windows XP Professional operating system. The Pocket PC is with PXA272 CPU (312 MHz) and Windows Mobile 2003 SE operating system. The Smart Phone uses Windows Mobile ™ 2003 2nd Edition, and its CPU is TI OMAP730 (200MHz). We have recorded the performance of our barcode reading system on a powerful platform (Desktop PC) and a less powerful one (Smart Phone) as shown below.

	Localization	Barcode Identify	Total Time
Desktop PC	62 ms	16 ms	78 ms
Smart phone	1274 ms	487 ms	1761 ms

Table 1: Computation Time of Barcode Reading

More details and results are available from our tech report.

4. SUMMARY

An automatic real-time barcode localization algorithm has been proposed in this paper. It consists of a two-stage processing, namely, first finding barcodes through a region-based analysis of a low-resolution image and then reading them in its original resolution. Extensive experiments have been conducted in complex 3-D scenes under various conditions, and very good results have been obtained. Future work includes reading of severely distorted barcode images.

5. REFERENCES

[1] S. Ando and H. Hontani, "Automatic Visual Searching and Reading of Barcodes in 3-D Scene", Vehicle Electronics Conference, Proc. IEEE pp. 49 – 54, Sept. 2001

[2] S.-C. J. Li, J. Xu, and T. Pavlidis, "A Window-Based Barcode Acquisition System", Proc. SPIE, vol. 2181, pp.125-132, 1994.

[3] H.-Y. Liao, S.-J. Liu, L.-H. Chen, and H.-R. Tyan, "A Barcode Recognition System Using Backpropagation Neural Networks", Engineering Applications of Artificial Intelligence, vol.8, no.1, pp.81-90, 1995.

[4] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "Comparison of edge detectors: A methodology and initial study," Comput. Vision, Graphics, Image Understanding, vol.69, no.1, pp.38—54, 1998.

[5] R. Boie and W. Turin, "Noise-limited reading of bar codes," IEEE Trans. Industrial Electronics, vol.IE-44, no.6, pp.816—824, 1997.

[6] E. Joseph and T. Pavlidis, "Bar code waveform recognition using peak locations," IEEE Trans. Pattern Anal. Machine Intell., vol.PAMI-16, no.6, pp.630—640, 1994.

[7] A.K. Jain and K. Karu, "Learning Texture Discrimination Masks," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, Issue 2, pp.195—205, Feb. 1996.

[8] N. Takahashi, M. Iwasaki, T. Kunieda, et al., "Image retrieval using spatial intensity features," Signal Processing: Image Communication, 16(1, 2): 45—57, 2000.

[9] M. Flickner, H. Sawhney, W. Niblack, et al., "Query by image and video content: the QBIC system," IEEE Computer, 28(9): 23—32, 1995

[10] L.D. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2D and 3D images," IEEE Trans. Pattern Analysis and Machine Intelligence, 15(11): 1131—1147, 1993.

[11] C. Nastar, B. Moghaddam, and A. Pentland, "Generalized image matching statistical learning of physically-based deformations," Proc. 4th European Conf. on Computer Vision (ECCV'96), Vol.1, pp.589-598, April, 1996.

[12] A. Bimbo, P. Pala, and S. Santini, "Image retrieval by elastic matching of shapes and image patterns," Proc. IEEE ICMSC, 215—218, 1996.

[13] M. Sonka, V. Hlavac, and R. Boyl, *Image Processing, Analysis, and Machine Vision*, Second Edition, ISBN 0-534-95393-X