Vision-Based 1D Barcode Localization Method for Scale and Rotation Invariant

Inyong Yun and Joongkyu Kim
College of Information and Communication Engineering
Sungkyunkwan University
Suwon-si, Korea
E-mail: {iyyun, jkkim}@skku.edu

***Abstract*—In this paper, we use a vision-based image processing technique to propose a method of real-time one-dimensional barcode localization robust for rotation and scale. The proposed method consists of three main steps. The first step generates and analyzes an orientation histogram of input images and removes their background regions and small clutters. The second step analyzes a local entropy-based orientation for the segmentation of a large region that is highly likely to be barcodes. Finally, the third step analyzes and verifies the barcode structures of the selected candidate region. The test results based on the data set available to the public demonstrate the proposed method is more accurate than existing methods by 10% or more.**

I. Introduction

Due to the relatively easy maintenance and management of data, pattern information is widely used not only in everyday life but also in industries including logistics and management. In particular, barcode patterns applied to factory automation save product management costs and improve productivity. Therefore, barcode patterns are used across the board as part of the automation system. Figure.1 shows the most widely used one-dimensional barcodes. As in the Figure, one-dimensional barcodes are the arrays of parallel lines, carrying some pattern information. In the past, to read the pattern information, they mostly used laser scanners. Laser scanners ensure the accuracy of information with refined sampling whereas they have to be placed close to barcodes for the extraction of information.

To address the challenge, research on the methods of recognizing barcodes based on images has been continuously published. To detect the positions of one-dimensional barcodes with image processing technology, Chai [1] performed a morphological processing known as skeletonizing and detected the positions of barcodes from images. However, when barcodes are slanted, their frequency properties change, which hinders the detection. Gallo [2] suggested a method of accelerating the detection of barcode positions. The algorithm calculates the horizontal gradients of each pixel, and uses the box filtering and global binarization to segment the barcode region. However, Gallo method failed to detect the slanted barcodes as their method focused on the horizontal gradients only. Sörös [3] built on the matrix structure suggested by Ando [4] to propose a barcode detection algorithm. From the calculated matrix structure, their derived a map of 1D/2D barcode segmentation, and detected the barcode region using Gallo method. Their method proved robust for rotation but slow.

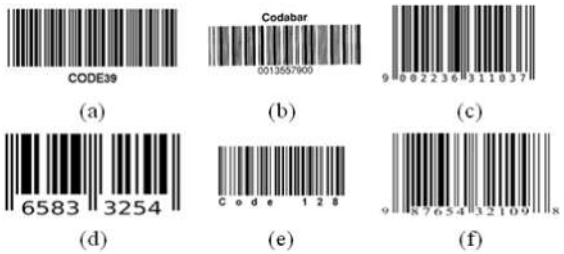


Fig. 1. Type of one-dimensional barcodes. (a) CODE 39 (b) codabar (c) EAN-13 (d) EAN-8 (e) CODE 128 (f) UPC-A

Katona [5] proposed a method based on the morphological operation. Capitalizing on bottom-hat filtering and distance mapping, the author detected barcode regions, which however proved unstable on the grounds that the barcode region decreased whilst the noise region increased as the number of barcodes in the image increased. Zamberletti [6] proposed a barcode detection algorithm based on the hough transform and machine learning, where the accuracy of detection varied with the results of hough transform. As one-dimensional barcodes consist of parallel straight lines, Bounar [7] used the probabilistic hough transform(PHT) to find densely distributed parallel lines and detect the barcode region. However, as they drew on the PHT only, the barcode detection was less accurate when barcodes included linear noises. Tekin [8] used an orientation histogram to detect barcode regions. However, when input images were blurred, the thin bars in the barcode region were removed, which increased the discontinuity of the region, leading to a detection failure. Jain [9] detected barcode regions by generating feature images using the gabor filter. Their method proved robust for scale and rotation but complex and slow.

This paper proposes a robust method of recognizing multiple barcodes and addressing the foregoing challenges arising from slanted barcodes and linear noises. To detect the barcode positions, we proposed an entropy-based method of detecting salient regions, and compared it with existing methods based on data sets available to the public. The comparative analysis highlighted the proposed method was less affected by scale and rotation.

This paper covers the following sections. Section 2 describes

978-1-5090-1134-6/17/$31.00 ©2017 IEEE

the proposed method. Section 3 compares the proposed method with existing methods based on the test results. Finally, Section 4 draws a conclusion.

II. Proposed Method

This section describes the proposed new method of one-dimensional barcode localization. The proposed method consists of the following three steps, i.e. pre-processing, barcode region segmentation and barcode verification. The preprocessing step uses the sobel operator to generate a global orientation histogram and orientation image. The global orientation histogram is used to separate the principal orientation components from the entire image, while the orientation image is used in the next step, or barcode region segmentation. To segment the barcode region, we calculate the local entropy of the orientation, and thus generate a saliency map. The region scoring high in saliency is defined as a candidate region to be used in the barcode verification step. Finally, by analyzing the one-dimensional barcode structure in the selected candidate region, we detect the final barcode region.

A. Pre-Processing

As for the position detection, one-dimensional barcodes are characterized by multiple parallel edges sharing a single orientation. To define the edges in the image, we detected the orientation whose gradient magnitude was greater than the fixed threshold. To that end, we transformed the gray scale of the input image, and used the sobel operator to calculate the images gradient. We used the following equation to find the magnitude and orientation of the pixel p, exceeding the threshold.

$$mag\,(p) = \begin{cases} |\nabla I(p)| & if \quad |\nabla I\,(p)\,| > T_{mag} \\ 0 & else \end{cases} \tag{1}$$

$$ang\,(p) = \begin{cases} tan^{-1}\left(\frac{\nabla I_y(p)}{\nabla I_x(p)}\right) & if \quad |\nabla I\,(p)\,| > T_{mag} \\ 0 & else \end{cases} \tag{2}$$

Where, $|\nabla I_x|$ and $|\nabla I_y|$ are the x-/y-oriented gradients of the pixel p in the image I, or $|\nabla I| = |\nabla I_x| + |\nabla I_y|$. T_{mag} is the threshold used to define the minimum gradient magnitude. To extract the principal orientation components from the entire image using the calculated orientation information, we created the global orientation histogram h_G. In generating the histogram, to secure a sufficient resolution for detecting the barcode region, we quantified the orientation into 18 bins, each of which covered $10°$. The following $V_{h_G}^{map}$ separates the principal orientation components from the weak orientation components analyzed in the h_G.

$$V_{h_G}^{map}(b) = \begin{cases} O_s & if \quad count(h_G(b)) > T_{hist} \\ O_w & else \end{cases} \tag{3}$$

$$T_{hist} = max(count(h_G(b))) * \alpha \tag{4}$$

Here, O_s denotes the principal orientation component. O_w is the weak orientation component. T_{hist} is the threshold used to separate the principal orientation component. α is the ratio constant used to calculate the threshold. Figure.2 illustrates the key aspects of the pre-processing.

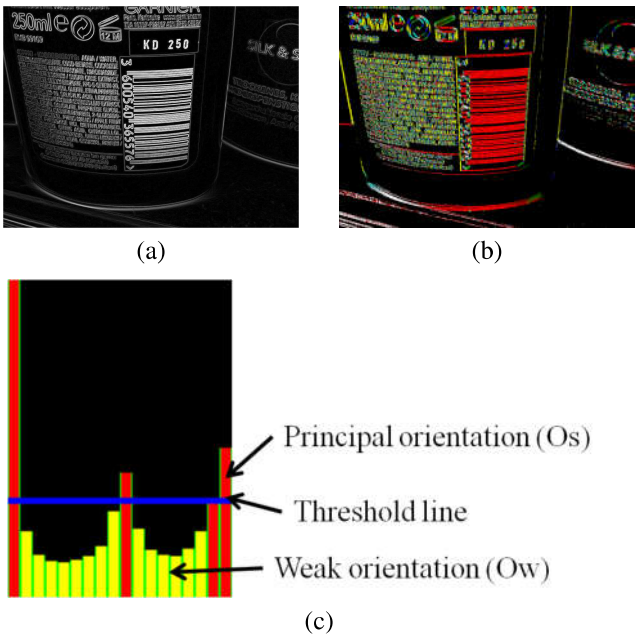


Fig. 2. Overview of the pre-processing step. (a) magnitude image, mag(p), (b) orientation image, ang(p), (c) global orientation histogram with $V_{h_G}^{map}$

B. Salient Region Detection

To detect a probable one-dimensional barcode region, we used the entropy scheme [10]. To calculate the entropy, we segmented the input image into non-overlapping N patches. We calculated the entropy for each patch region as follows.

$$E(f) = \begin{cases} J & if \quad V_{h_G}^{map}(max_{bin}(h_L)) = O_s \\ 0 & else \end{cases} \tag{5}$$

$$J = \left(\sum_{i=0}^{K-1} count(h_L(i))\right) - max(count(h_L)) \tag{6}$$

In equation (6), K means a quantized orientation value set (i.e., bin). h_L is the local orientation histogram of a segmented patch region f. In equation (5), $max_{bin}(h_L)$ indicates the bin when h_L is maximum. When the patch f's maximum orientation component is identical to the main orientation component in $V_{h_G}^{map}$ generated in the pre-processing step, $E(f)$ is substituted by the value of entropy J.

If the value of entropy J is close to zero, the patch region f is highly likely to be a complex region, say, with characters or a background region carrying null information. By contrast, if J is close to the maximum value, the patch region f is likely to have a certain value of orientation. As the one-dimensional barcodes of interest here showed arrays of parallel edge components sharing an orientation, J has the maximum value in the barcode region. We used the following equation (7) to set an important region of a high saliency score including one-dimensional barcodes.

$$S(f) = \begin{cases} E(f) & if \quad E(f) > T_{salient} \\ 0 & else \end{cases} \tag{7}$$

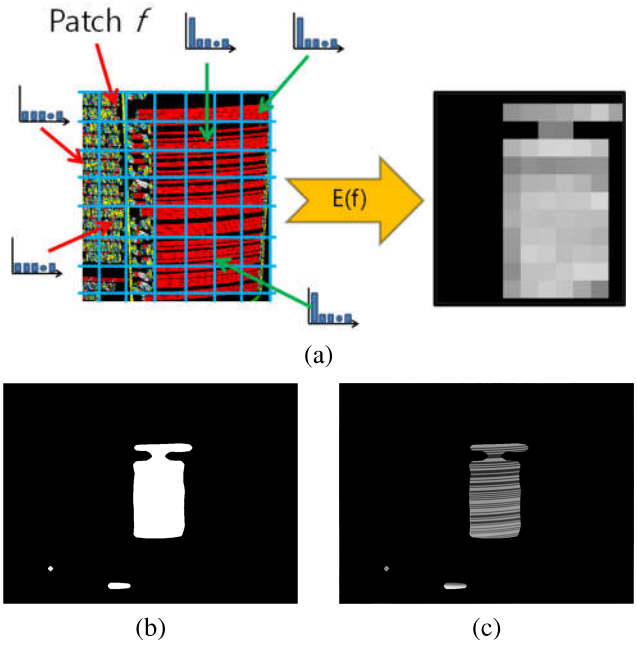


Fig. 3. An example showing the barcode localization process by saliency detection in the Muenster barcode DB. (a) saliency map, (b) binarization by otsu method, (c) candidate barcode region(s)

where, $T_{salient}$ is the threshold used to separate the high saliency score region. When the saliency score is smaller than threshold value, $S(f)$ is filled by zero. $S(f)$ is the final saliency map.

After setting the important region, we used a box filter measuring $M \times M$ to blur $S(f)$, to eliminate the noise region and to connect the separated barcode regions. The blurring helped connect the separated barcode regions, which exerted robust impacts on scale. We determined the box filter window size (25 pixels) based on its ratio to the input image size. Then, we performed the image binarization as suggested by Otsu [11]. The binarized image was used to determine the barcode positions in the input image. The binarized saliency map $S(f)$ contained more than 1 blobs. To determine each blobs central point and bounding box, we used the connected components labeling. Figure.3 shows an example of the proposed method of setting the region of interest.

C. Barcode Verification

After estimating the region of interest, we determined the barcode region with reference to Tekin [8]. The bar structures of one-dimensional barcodes were iterative at regular intervals. Thus, we analyzed the number of bar structures and the distance between them in the region of interest. This process commenced at the central point of the candidate barcode region. As in equation (5), based on the orientation of $max_{bin}(f)$, the scanning expands vertically or horizontally. Figure.4 shows an example of the proposed verification process. Once the analysis of the central point of the blob and both directions was completed, we performed additional tests by changing the orientation within the range of [$max_{bin}(f)$ + 1, $max_{bin}(f)$ -1], which added to the accuracy of detection.

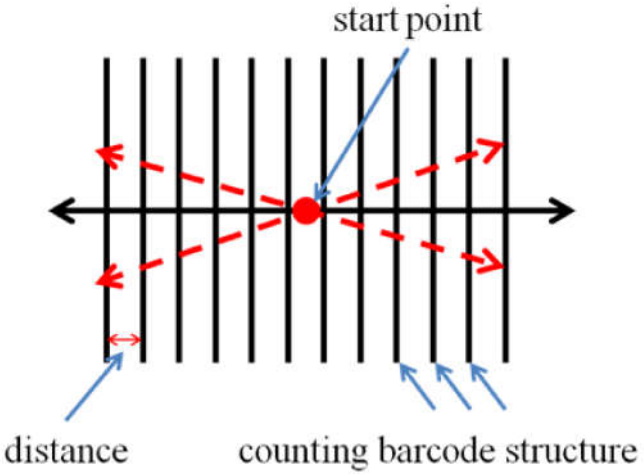


Fig. 4. An example of barcode verifIcation process.

III. Experimentations

We developed the proposed algorithm using the OpenCV framework and Microsoft Visual Studio. The PC used for the test featured Intel Core i5 3.2 GHz CPU and RAM 8GB. We compared the accuracy and speed of the barcode detection between three existing methods - Gallo [2], Sörös [3] and Tekin [8] and the proposed method. We implemented Gallos and Sörös methods to get the bounding boxes, whilst using the release version of Tekins method of generating a single scan line. To test the proposed method, we applied fixed parameters to the entire test image. In equation (1), the control parameter T_{mag} is set at 30. In equation (4), α is 0.3. In equation (7), T_{salent} is set at 0.5 with normalization.

A. Dataset

In the test, we used the Muenster Barcode Database [12], and the images carrying a range of scales, orientations, blurriness and multitude of barcodes. In addition, we collected 338 pieces of barcode images. Also, our barcode images are available to the public[1]. Figure.5 shows the tested images of diverse settings.

B. Performance Measurement

To measure the performance, we used the dice similarity coefficient (DSC) and determined the overlap ratio between the ground truth and the bounding box of the detected barcode region. To get the bounding box of the ground truth, we selected and saved the corner points of barcodes in the entire test images. We measured the overlap ratios based on the following DSC.

$$DSC(A,B) = \frac{2C}{A+B} = \frac{2|A \cap B|}{|A|+|B|} \quad (8)$$

[1]http://dspl.skku.ac.kr/index.php?mid=DB

Fig. 5. A set of test images. (a) Muenster barcode DB [12], (b) our barcode DB.

TABLE I
AVERAGE DICE SIMILARITY COEFFICIENTS ON TEST IMAGES.

Methods	dataset1 [12]		dataset2	
	avg. DSC	std. dev	avg. DSC	std. dev
Gallo [2]	0.744	0.334	0.516	0.396
Sörös [3]	0.816	0.143	0.588	0.267
Tekin [8]	0.737	0.300	0.626	0.395
Proposed	0.888	0.159	0.789	0.280

Where, A is the bounding box of the ground truth. B is the bounding box of the detected region (or the single scan line). C is the overlapping region between A and B. In the test, a coefficient exceeding 0.8 indicated a good match. When the test returned a single scan line, we calculated the DSC only of the one-dimensional lines of the code orientation because Tekin had a single line printed instead of a bounding box.

C. Comparison with Previous Works

As in Table 1, we tested the accuracy of DSC. Here, the dataset 1 is Muenster Barcode DB (1050 images), while the dataset 2 is our barcode DB (338 images) we used. We measured the overlap accuracy with the mean coefficient of the entire images. The proposed method returned 0.839 as the mean overlap coefficient of the barcode regions out of more than 1300 pieces of images. This finding corroborated the greater accuracy of the proposed method than the others. Figure. 6 shows the result of barcode detection performance of the proposed method along with the ground truth bounding box.

Table 2 shows the success rates of barcode detection. Here, we considered the detection as successful when $DSC \geq 0.8$. The accuracy of detection in the proposed method averaged 85.68% out of the more than 1300 images, outstripping the other methods. The method suggested by Gallo [2] detected barcodes fast, but mostly returned smaller or larger results, and failed to detect the rotated barcodes. The method used in Tekin [8] proved robust for rotated barcodes but largely printed scanned lines shorter than the barcode region detected. Sörös method [3] returned largely shorter or longer barcodes than the ground truth and mis-detected complex characters in the background or the parts of certain patterns.

TABLE II
SUCCESSFUL 1D BARCODE DETECTION RATES ($DSC \geq 0.8$ A GOOD MATCH) ON 1300 IMAGES.

Methods	Detection Rate		
	Dataset 1 [12]	Dataset 2	Average.
Gallo [2]	78.65%	40.71%	59.68%
Sörös [3]	70.21%	24.48%	47.35%
Tekin [8]	54.08%	51.62%	52.85%
Proposed	92.60%	78.76%	85.68%

TABLE III
COMPARISON OF AVERAGE RUNNING TIMES.

Methods	Gallo [2]	Sörös [3]	Tekin [8]	Proposed
Time	7.91 ms	57.41 ms	26.72 ms	24.47 ms

Finally, we tested the detection speed. When the input image resolution was 1024×768, it took the proposed method 24.47ms on average. Specifically, the proposed method performed as follows: gray conversion (0.62 ms), gradient calculation (10.68 ms), calculation of saliency (1.79 ms), box filtering (3.72 ms), calculation of connected component with binarization(3.64 ms) and barcode verification (0.014 ms). The proposed method spent about 50% of the time on calculating the gradient. As in Table 3, we compared the proposed method with the existing ones in terms of the mean time spent on barcode detection. All source codes were written in C/C++. Gallos method [2] proved most straightforward and fastest, followed by the proposed method. This finding indicated the proposed method could be applicable to real-time detection.

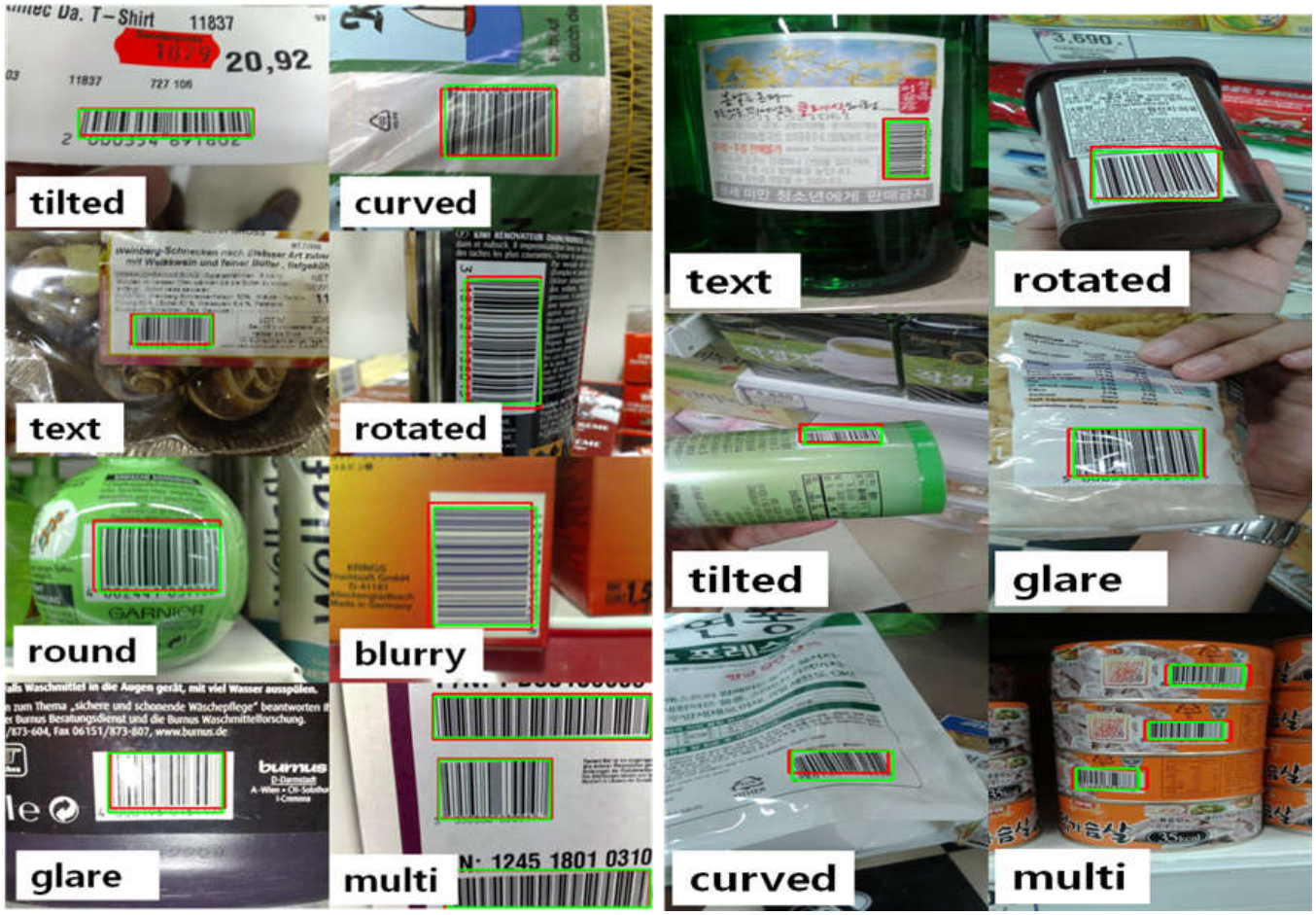
IV. CONCLUSION AND FUTURE WORK

This paper proposed a new barcode detection method of detecting entropy-based similar structures and orientations. We compared the detection speed and accuracy between the proposed method and the existing three methods, using open data sets. Taken together, the proposed detection method outperformed the existing methods by 10% or more. Also, the proposed barcode detection method of analyzing the entropy of orientation in our algorithm experimentally showed fewer false positives than the existing methods. In near future, we are to build on the proposed method to develop an additional algorithm for decoding barcodes.

REFERENCES

[1] D. Chai and F. Hock, "Locating and decoding ean-13 barcodes from images captured by digital cameras," in *Information, Communications and Signal Processing, 2005 Fifth International Conference on*. IEEE, 2005, pp. 1595–1599.

[2] O. Gallo and R. Manduchi, "Reading 1d barcodes with mobile phones using deformable templates," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 9, pp. 1834–1843, 2011.



(a) (b)

Fig. 6. Positive example from the database. A red rectangle indicates our detection result, greed rectangles represent the hand-labeled ground truth. (a) Muenster barcode DB [12], (b) our barcode DB.

[3] G. Sörös and C. Flörkemeier, "Blur-resistant joint 1d and 2d barcode localization for smartphones," in *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 2013, pp. 11–18.

[4] S. Ando, "Image field categorization and edge/corner detection from gradient covariance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 179–190, 2000.

[5] M. Katona and L. G. Nyúl, "A novel method for accurate and efficient barcode detection with morphological operations," in *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*. IEEE, 2012, pp. 307–314.

[6] A. Zamberletti, I. Gallo, S. Albertini, and L. Noce, "Neural 1d barcode detection using the hough transform," *Information and Media Technologies*, vol. 10, no. 1, pp. 157–165, 2015.

[7] P. Bodnár and L. G. Nyúl, "Improving barcode detection with combination of simple detectors," in *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*. IEEE, 2012, pp. 300–306.

[8] E. Tekin and J. Coughlan, "Blade: Barcode localization and decoding engine," *Tech. Rep. 2012-RERC. 01*, 2012.

[9] A. K. Jain and Y. Chen, "Bar code localization using texture analysis," in *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*. IEEE, 1993, pp. 41–44.

[10] S.-K. Chang and C.-C. Yang, "Picture information measures for similarity retrieval," *Computer vision, graphics, and image processing*, vol. 23, no. 3, pp. 366–375, 1983.

[11] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[12] S. Wachenfeld, S. Terlunen, and X. Jiang, "Robust recognition of 1-d barcodes using camera phones," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.