A novel method for accurate and efficient barcode detection with morphological operations

Melinda Katona and László G. Nyúl
Department of Image Processing and Computer Graphics
University of Szeged
Árpád tér 2, H-6720 Szeged, Hungary
katona.melinda@stud.u-szeged.hu, nyul@inf.u-szeged.hu

***Abstract*—Barcode technology is the pillar of automatic identification, that is used in a wide range of real-time applications with various types of codes. The different types of codes and applications impose special problems, so there is a continuous need for solutions with improved effectiveness. There are several methods for barcode localization, that are well characterized by accuracy and speed. Particularly, high-speed processing places need automatic barcode localization, e.g. conveyor belts, automated production, where missed detections cause loss of profit.**

In this paper, we mainly deal with segmentation of images with 1D barcode, but also analyze the operation of different methods for 2D barcode images as well. Our goal is to detect automatically, rapidly and accurately the barcode location by the help of extracted features. We compare some published method from the literature, which basically rely on the contrast between the background and the shape that represent the code. We also propose a novel algorithm, that outperforms the others in both accuracy and efficiency in detecting 1D codes.

***Index Terms*—barcode detection, computer vision, mathematical morphology, bottom-hat filter**

I. Introduction

Barcode detection is required in a wide range of real-life applications. Computer vision algorithms vary considerably and each application has its own requirements for accuracy and detection speed. Many barcode localization methods have been developed for automatically segmenting barcode patterns from images.

The term barcode can be used for various types of visual codes. In this paper, we deal with classical 1D barcodes and stacked 2D barcodes segmentation. Barcodes are not human-readable and traditional devices have been widely adopted for personal use. The traditional barcode structure is simple: the variation of different thickness of parallel light and dark bars represent information. Such codes can be read optically by a machine. The code types vary from each other in what black and white bars correspond to a given character. The most frequent application of barcodes is the trade, e.g. in goods packing, where it permits fast identification of goods data such as manufacturer's country, manufacturer's identification number and the product's item number. The barcode identification is the most known element of the GS1 system, which is an inseparable part of trading procedures. The identification number visualized with symbols permit the use of electronic reading by machines which support and speed up the information streaming. Barcodes can have fix or variable length. In fix length codes, the standard specifies how many characters are represented in a code, while the other type may encode an arbitrary number of characters. The different standard code types have specific features that help their localization.

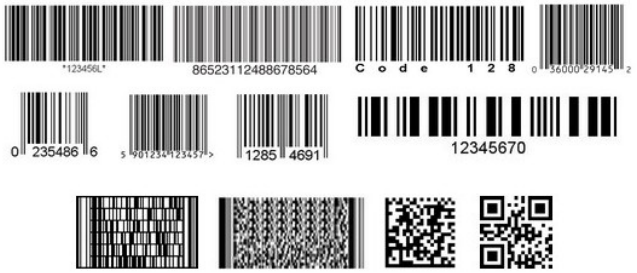


Fig. 1. Barcode patterns (from left to right). Top row (1D codes): Code39, Codabar, Code128, UPC-A; Middle row (1D codes): UPC-E, EAN-13, EAN-8, I2of5; Bottom row (2D codes): Codablock, PDF417, Data Matrix, QR.

Figure 1 shows a set of standard 1D barcode types and some widely used 2D barcode types. [1]

Barcode localization methods have two objectives, speed and accuracy. For industrial environment, accuracy is crucial since undetected (missed) codes may lead to loss of profit. Processing speed is a secondary desired property of the detectors. On smartphones, the accuracy is not so critical, since the device interacts with the user and re-shoting is easily possible, but a fast (and reasonably accurate) barcode detection is desirable. Various techniques are used to locate and decode barcodes from photographs, from the classical line scanning technique [2], through the widely studied morphological approaches [3]–[9], and recent studies using wavelets [10].

We propose a novel barcode detection algorithm based on bottom-hat filtering and other morphological operations, which guarantee high efficiency. We compare the effectiveness of the new method with several approaches from the literature and can show that, in most cases, our algorithm outperforms the others.

II. Methods

In this section, we present four barcode detection algorithms. They use different approaches to determine the barcode location. In the pseudocode we use the notation of [11]. The

methods presented in Sections II-A, II-B, and II-C were re-implemented according to the original papers. Where some details were not available in the papers, we made our best effort to fill in the gaps and this is noted in the text. Figure 2 shows intermediate stages of the processing by the four algorithms.

Nowadays, the quality of digital images are usually very good, although low-quality recordings are also made. The reason for lesser quality may be e.g. the capture device, but the environment may also cause troubles. Subsequently, often there is a need for correcting (rather improving) the image quality before starting the particular processing.

A. Method based on basic morphological operations

First, we present Tuinstra's [3] algorithm. This method relies on basic morphological operations. Most algorithms use simple operations in preprocessing, such as quantization, wherein the input image is converted into another range of intensity values (e.g. binary). In this case, the authors rely on that in the barcode region, the intensity difference between the stripes is high, so a gradient calculation highlights the bars. To estimate the gradient in the x and y directions, Sobel kernels are used. Next, the gradient image is thresholded and pixels having a high gradient value are selected.

The rest of the procedure works on the binary image. The localization is based on morphological operations. First step is a hit-or-miss transformation with a line structure element. Unfortunately, the structure element is not specified in the article [3]. In our implementation, the structure element was chosen considering the length of the longest bar in the images. We used a 10x10 SE which is sufficiently large for the objective and smaller than the expected bar length in the codes. In the course of the procedure all objects are removed that likely do not contain barcode.

After the hit-or-miss transformation, morphological dilation is performed in order to merge nearby but not necessarily connected objects to be able to compose a region. The structure element is square shaped, but in the article its exact size was not fixed. We used a 10x10 block SE matching the size used for the previous step. In practice, the image size determines this parameter. Certainly, dilation fuses some regions that do not contain barcode, but in a subsequent phase these false results are eliminated.

The next step is to perform morphological erosion to discard thin objects from the image. The structure element size is greater than that used for dilation. We used a 20x20 block SE. This step removes undesired segments which were fused by the dilation.

The final operation is a solidity test which compares the number of pixels turned on in a region to the convex hull of region. So, in this step, probably false positive objects are removed and only barcode regions remain.

B. Method based on image scanning

The procedure of Telkin and Coughlan [4] was designed for visually impaired or blind people to facilitate their everyday lives. The article gives a complete description from the quality improvement of the input image to reading the barcode. Here we just recall the part of barcode localization.

The algorithm is most suitable for the UPC-A barcode type, but is also suitable for the localization of other 1D codes like UPC-E, and the EAN standard family of codes. The procedure is based on scanning the image at different orientations.

The procedure is presented in two main phases. The quality of the discrete images obtained from the input device is not ideal, so a preprocessing phase is necessary before the particular barcode localization. The first step is Gaussian smoothing to reduce the noise level. Since the original paper did not specify the σ, we used the same value of $\sigma = 0.3$ as in our Proposed method. Then pixel gradient values are calculated for the entire image using the Sobel operator, thus yielding a kind of edge enhancement in the image.

In the next step the algorithm works with only those pixels where the gradient value is above a threshold, so a binarization is done so that pixels having a gradient above the threshold become white and the others black. We used a threshold value set to 95% of the maximal gradient value.

The detection phase is next. Firstly, the image is scanned in four directions (horizontally, vertically, and in direction of the diagonals ($\pm 45°$)). The horizontal scan comes first, wherein those edge pixels are found whose orientation is vertical. The method looks in the vicinity of each edge pixel for opposite polarity pixels. If there is a sufficient number of such pixels in an area, as if they were part of a line segment, this area will not be removed. Subsequently, vertical scan follows, when those segments remain which have almost the same beginning and end so they most likely belong to barcode areas. If there appears to be more than one such region, then the next section of operations will filter out the false positive regions.

At the final stage of the detection the entropy value is calculated for each pixel in the result image describing the disorder of intensities within a given neighbourhood around each pixel. It is recalled that the barcode lines are parallel with each other, so region entropy would be likely low, and the entropy of false positive segments are expected to be high.

C. Method based on bottom-hat filtering

Next, we describe an algorithm by Juett and Qui [5], that is based on bottom-hat filtering.

If the input is not a grayscale image, the algorithm will convert it to that by quantization. In preprocessing, the method corrects the non-ideal image with simple contrast stretching in order to highlight differences between light and dark areas. When we converted the color image to grayscale, the image intensity range scales into a given interval. The next step is bottom-hat filtering, when the closing of the stretched image is subtracted from the original stretched image. [12] The structure element size depends on the widest bar in the barcode to be detected. The article specified 25x25 block SE for images of size 720x480 pixels. In less complex images, after bottom-hat filtering the false areas are less than after

Fig. 2. Intermediate stages of the processing by the four algorithms. Columns: TT, ETJC, JJXQ, and Proposed. First row: original image, last row: final output.

gradient calculation. The rest of this algorithm works with binary images, so the next step is the binary conversion.

After this, the contour is defined. The binary image is eroded using a 5x5 structure element and subtracted from the original binary image.

This is followed by the step to determine the orientation, which is performed by directional image openings using a relatively large linear structure element. These openings are performed at 16 different orientations, with a step of $11.25°$. Selecting one suitable orientation is difficult, because the barcodes could be in any orientation between $0°$ and $180°$. Thus several orientation is probed which significantly increases the execution time of the algorithm. The directional opening images are summarized and a low resolution density image is calculated. Then, this image is converted back into binary and each region represents a potentially barcode region.

In the last phase, objects whose area is smaller than a threshold are eliminated. Using the centroids of the remaining objects lines that are next to each other are found and the potential corner points of the object are computed.

D. The proposed method

In this section, we present our novel method for barcode localization. The algorithm is based on morphological operations. Our aim was to devise an algorithm with sufficiently high speed and accuracy.

Similarly to the previously described methods [3]–[5], our algorithm consists of two main phases. In the preprocessing phase, the input image is converted to grayscale, because although barcodes may be printed in various colors, the pattern of dark bars on a light background is equivalent.

To reduce the image noise, we use smoothing with a Gaussian kernel where $\sigma = 0.3$. Edge enhancement follows, which in most barcode localization method is done by calculating gradient values. Instead, we use bottom-hat filtering that is also based on intensity differences. Although, bottom-hat filtering is less attractive regarding operation time than other non-direction edge enhancement operations, its accuracy is higher. For the further steps of the procedure the grayscale images are converted to binary, using a standard thresholding technique. To select the correct threshold value, we considered that the barcode has positive intensity but not necessarily very high and set the threshold to 95% of the maxmimal value.

After preprocessing the digital image is suited to finding segments which contain barcode. The result image so far contains a lot of false regions. We take advantage of the structure of the barcode, the fact that it consists of parallel lines at about same distance. We analyze the entire image using a horizontal scan whether the white pixels are located at appropriate distances at the given direction. The maximal distance parameter depends on the image resolution.

After scanning, there are many small connected components which satisfy the criteria but are not barcode regions. These false regions are eliminated in the next step using an area threshold. We set the threshold to half the size of the largest component. A too high threshold value could remove small barcodes from high-resolution images, therefore in this step we only remove the smaller false regions.

Our observations showed, that for the final detection steps mostly dense text regions remain along with the supposedly barcode areas. Since a barcode consists of a sequence of parallel bars and the bars are located at varying distance from each other, they do not compose a connected component. Therefore in this phase we use dilation, to merge these patterns. We use a square shaped structure element for dilation whose size is defined as

$$S = \max(40, \text{width of the widest bar} * 3)$$

Nevertheless, this dilation may also thicken and merge unwanted non-barcode locations as well. To cope with this problem, we also use the dual operation, i.e. erosion.

Here, the structure element is also square shaped, consistent with the size of the 1D barcodes. The structure element size is less than (about 1/3 of) the one used for dilation, that is matching the width of the widest bar. After erosion, those areas that possibly contain barcode areas can be found. Of course, there may still be false-positive objects which are removed in the last step on the basis of their size and proportions. Similarly to the previous step, objects smaller than half the size of the largest object are removed.

INITIALIZATION:

N = image width	Se_2 = dilation struct. el.
M = image height	i_t = minimum threshold
f = grayscale image	Se_1 = bottom-hat struct. el.
$\bullet$ = morph. closing	max_d = maximum distance
min_a = minimum area	$\ominus$ = morph. erosion
$\oplus$ = morph. dilation	Se_3 = erosion struct. el.

ALGORITHM:

$f := (f \bullet Se_1) - f$

for $i := 0, \ldots, N$ **do**
  for $j := 0, \ldots, M$ **do**
    if $f_{i,j} > i_t$ **then**
      if $f_{i,j} < max_d$ **then**
        $f_{i,j} \leftarrow 1$
      else if
        then $f_{i,j} \leftarrow 0$ *// no edge seen within specified maximum distance*
      end if
    else if
      then $f_{i,j} \leftarrow 0$
    end if
  end for
end for

$f := f \oplus Se_2$
$f := f \ominus Se_3$

if object area $> min_a$ **then**
  Record this object as a barcode segment
else
  Discard this object
end if

Fig. 3. Pseudocode of the proposed algorithm

The proposed algorithm is summarized in Figure 3.

III. Evaluation

In this section, we compare the discussed methods' effectiveness under specific characteristics. We use the following acronyms for referring to the algorithms: TT (Timothy R. Tuinstra's method [3]), ETJC (Ender Telkin és James M. Coughlan's method [4]), JJXQ (James Juett és Xiaojun's method [5]), and Proposed, for our new proposed algorithm.

A. Test suite, test environment, and implementation

We generated barcodes digitally with the types shown in Figures 1. Only one base image was chosen for each code

type. Test images contained one or three barcodes from each types, respectively, and images were affected by distortions. For each base image, we generated all combinations of the following properties: rotation in every 15° from 0° to 180°, Gaussian blur filter with 3x3 kernel with 6 different σ, additive noise from 0% to 50% with a step of 10%, and 15% shear. In summary, the test set contained images containing either 1 or 3 codes of 10 different barcode types, with 12 orientations, 6 different blur filters, and 6 different rates of additive noise, and with or without shear, with a total of 17280 images.

Figure 4 shows a set of generated test images with various blur and noise level. For clarity, shear and rotation are not illustrated here, which are, of course, applied to individual codes in the distorted images.

Another 100 images containing barcodes were collected from real-life examples without any modifications. These images presented scratches, blur, minor light reflections and distortions also.

We implemented the methods in MATLAB, with the help of the Image Processing Toolbox. Evaulation was performed on a computer with Pentium(R) Dual-Core 2.30 GHz CPU.

B. Results and discussion

In this section, we show how effective the implemented algorithms are on images with various characteristics, and compare their running time w.r.t each other. For calculating accuracy we used the Jaccard the coefficient of similarity, which measures the overlap of the bounding boxes of the real and the detected barcode region. $J(A,B) = \dfrac{|A \cap B|}{|A \cup B|}$, where A denotes the real barcode and B is the bounding box of the detected code. This not only measures the success of detection, but also considers the delination accuracy of the methods.

We can conclude that all four algorithms are capable of effectively detecting barcodes. The ETJC method is fast, however, when there are several barcodes in the image and noise is also present, it loses efficiency. As already mentioned, we also applied stretching when generating the images. The efficiency of the algorithms do not seem to depend on the streching and also seem to be insensitive to rotation as well.

TT is slower than the other algorithms, which can most probably be attributed to the fact that here the image is scanned in four different orientations.

JJXQ, the method based on bottom-hat filtering falls off to the discussed fast algorithms with respect to running time. The process is slower than the others because in the detection phase the image is scanned in 16 different orientations. This brings greater accuracy to the process but at the expense of more computation.

Our proposed algorithm is also very fast. In most cases, faster procedures locate barcodes less accurately or not at all, as if loss of accuracy were the price for the speed gain. However, in our case, speed is not coupled with a high cost in accuracy. From the evaluation output we can conclude that the proposed procedure has the best running efficiency for all test images, and in many cases it also shows best accuracy.

TABLE I
ACCURACY OF THE ALGORITHMS FOR VARIOUS COMBINATIONS OF DISTORTIONS (BLUR (B), NOISE (N), SHEAR (S)). MEAN VALUES (EXPRESSED IN PERCENT) FOR ALL IMAGES WITH 1D BARCODES (TOP), AND FOR ALL IMAGES WITH 2D CODES (BOTTOM).

	B N S	TT	ETJC	JJXQ	Proposed
1D	− − −	100.0	100.0	100.0	100.0
	+ − −	83.1	79.9	76.8	96.1
	+ + −	84.5	84.5	79.4	96.4
	− + −	82.1	81.9	75.1	95.9
	− + +	91.7	83.0	81.1	96.2
	+ + +	80.8	88.6	83.7	91.2
	+ − +	90.6	83.5	90.0	97.1
	− − +	82.8	84.0	77.2	96.1
2D	− − −	100.0	100.0	100.0	100.0
	+ − −	83.5	98.0	76.9	91.0
	+ + −	88.2	99.8	76.1	92.6
	− + −	82.4	99.8	76.0	90.0
	− + +	90.1	98.0	80.2	90.9
	+ + +	84.3	97.8	74.8	89.9
	+ − +	88.7	100.0	82.7	96.1
	− − +	87.5	99.9	77.1	91.0

TABLE II
RUNNING TIME OF THE ALGORITHMS. IN IMAGES WITH 1D (ABOVE), RESPECTIVELY 2D (BELOW) BARCODE MEAN VALUES (± STANDARD DEVIATION) (EXPRESSED IN PERCENT) FOR ALL IMAGES WITH 1D BARCODES (TOP), AND FOR ALL IMAGES WITH 2D CODES (BOTTOM).

		TT	ETJC	JJXQ	Proposed
1 code	1D	0.23±0.13	0.67±0.22	0.81±0.43	0.13±0.10
	2D	0.21±0.14	0.77±0.47	0.73±0.44	0.12±0.12
3 codes	1D	0.54±0.35	1.73±0.66	1.86±0.78	0.18±0.09
	2D	0.46±0.21	1.70±0.54	1.68±0.71	0.18±0.10

The weakness of the method appears when the images is very noisy or when there are such image areas which are similar to a barcode.

We mentioned earlier that the generated test images contain various amount of added noise and smoothing, and they contain one or several barcodes. We show the effect of these properties on the behaviour of each described method. In Table I the overall accuracy of the algorithms is displayed with respect to the different distortions. The efficiency of the methods, i.e. processing time is also an important aspect. Table II presents the execution time of the detection methods for images with 1 or 3 barcodes. Here, one can easily appreciate the significant differences between the different approaches.

Subsequent tables show, how the algorithms behave on images which contain either one or three code pieces of various 1D and 2D barcodes types. The structure of barcode types varies which also has an effect on how well the algorithms can perform. Accuracy of the detection methods for images containing a single piece of code for various code types is presented in Table III. TT, ETJC, and JJXQ has very bad performance for Code128, however the Proposed algorithm handles this variable-length code as well as the fixed-length types. For the 2D codes, ETJC shows exceptionally good accuracy, but the other three methods also perform well on these stacked barcodes. As to images with three barcodes, results are also shown in the bottom part. There is more

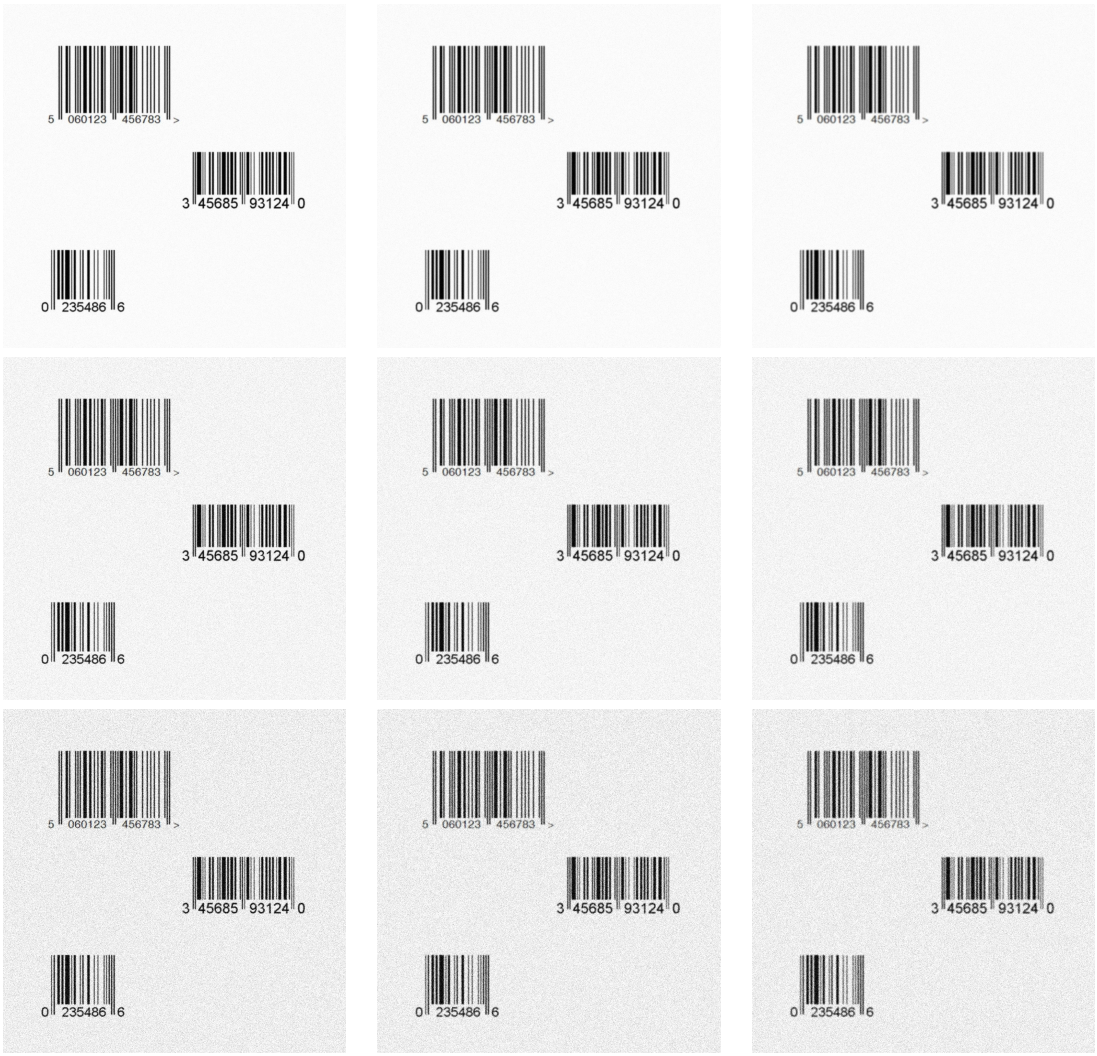


Fig. 4. A set of generated test images with various blur and noise level. Rows: 10%, 30%, and 50% noise, Columns: Blur with $\sigma = 0.5, 1.5, 2.5$.

variance in performance and some methods (TT, Proposed) show a solid performance while the others very much depend on the type of code present in the image.

The accuracy of the algorithms degrades on distorted images. Next, we analyze the effect of distortions on efficiency. Table IV shows the methods' behavior with respect to the level of blur applied to the images. Image smoothing does not change the performance considerably. Although there is a noticeable inverse relation between σ and accuracy, this is not significant. It is interesting to notice that in some cases the barcode detection accuracy is higher in images with three barcodes than for images with a single code present. This can be due to the fact that relative size (w.r.t. image size) of individual codes are smaller in the 3-code examples. ETJC and the Proposed method show outstanding accuracy in images which contain 2D barcodes and there are not considerable difference between the precision values.

In a similar manner, we analyzed how the algorithms perform on images with different levels of noise. We can see from Table V that each algorithm is somewhat sensitive to noise. The studied literature methods lose much of their accuracy as the noise level increases, however, the Proposed method degrades to a lesser extent.

We tested the methods on a set of 100 real-life images, too. These images contain 1D barcodes each. Table VI demonstrates the accuracy of the algorithms.

In Figure 5 we show a visual comparison. Using a single image example, the intermediate and final output of the four tested algoritms are shown at their key phases. The chosen real image (column (a)) is noisy and non-ideal. Column (b) shows the result after preprocessing. In column (c), the segmentation output before the last step is displayed. Here, ETJC and TT results contain the barcode location and also some other false regions. The Proposed method only shows the barcode

TABLE III

ACCURACY OF THE ALGORITHMS FOR VARIOUS TYPES OF CODES, FOR IMAGES CONTAINING A SINGLE CODE AND FOR THOSE CONTAINING 3 CODES. MEAN VALUES (EXPRESSED IN PERCENT) FOR ALL DISTORTED IMAGES WITH 1D BARCODES (TOP), AND FOR ALL IMAGES WITH 2D CODES (BOTTOM).

1 code

		TT	ETJC	JJXQ	Proposed
1D	Codabar	99.0	98.9	99.2	99.9
	Code128	50.4	63.6	70.9	95.7
	Code39	98.7	94.8	86.3	99.4
	EAN-13	98.8	94.7	91.7	98.9
	EAN-8	98.7	99.6	84.6	99.6
	I2of5	98.6	84.6	84.4	93.1
	UPC-A	99.1	98.2	94.9	86.4
	UPC-E	82.7	92.6	76.0	99.6
	All 1D	91.8	86.4	78.9	96.6
2D	Codablock	87.0	99.8	83.3	87.7
	PDF417	81.7	99.6	79.7	93.9
	All 2D	85.0	99.7	82.5	90.8

3 codes

		TT	ETJC	JJXQ	Proposed
1D	Codabar	83.8	56.2	52.7	92.8
	Code128	83.6	95.0	79.6	80.1
	Code39	83.3	90.5	67.3	94.7
	EAN-13	83.3	91.4	76.6	99.3
	EAN-8	83.3	97.2	81.8	99.4
	I2of5	78.8	35.5	93.8	99.2
	UPC-A	74.6	65.5	92.6	99.9
	UPC-E	98.7	77.5	96.7	99.3
	All 1D	74.0	74.7	77.4	95.6
2D	Codablock	86.6	100.0	80.0	90.7
	PDF417	83.3	100.0	83.9	91.7
	All 2D	85.0	100.0	82.4	91.2

segment at this stage. From the images in column (d), one can have an impression of how accurately each method locates the barcodes.

For the digitally generated images maximal size was 800x800 px. At this resolution TT and JJXQ cannot compete with the other two methods in speed. The Proposed method is not far better on 800x800 images than ETJC, but when executing them on 2500x1722 px images, ETJC runs for 3.2 sec, while our Proposed algorithm finishes under 1.3 sec. We can state, that for larger images there are considerable execution time differences between the algorithms. The chart in Figure 6 well illustrates that the Proposed algorithm outperforms the other three both in terms of accuracy an speed. Figure 6 displays each algorithm's average running time, and their accuracy for all digitally generated test images. Note, that methods which use bottom-hat filtering in the preprocessing phase, have higher accuracy than the other two, but their running times are quite different from each other. JJXQ loses its accuracy proportional to image degradation (see Table I), while our Proposed algorithm maintains higher accuracy for those cases as well. Although the diagrams do not show, the tests also demonstrated that in all cases the methods' running time significantly grows with increasing image size.

In the test images the barcodes are in different orientations, but tests showed that the algorithms are not sensitive to code orientation.

TABLE IV

ACCURACY OF THE ALGORITHMS FOR DIFFERENT BLUR LEVELS, FOR IMAGES CONTAINING A SINGLE CODE AND FOR THOSE CONTAINING 3 CODES. MEAN VALUES (EXPRESSED IN PERCENT) FOR ALL DISTORTED IMAGES WITH 1D BARCODES (TOP), AND FOR ALL IMAGES WITH 2D CODES (BOTTOM).

1 code

		TT	ETJC	JJXQ	Proposed
1D	0.0	100.0	100.0	100.0	100.0
	0.5	91.2	88.4	91.3	96.1
	1.0	90.4	88.5	89.4	97.1
	1.5	90.4	88.5	89.4	96.8
	2.0	90.3	90.4	87.5	97.1
	2.5	90.4	90.4	87.5	97.1
	All 1D	90.5	89.1	89.6	96.7
2D	0.0	100.0	100.0	100.0	100.0
	0.5	88.8	100.0	83.1	95.8
	1.0	88.4	100.0	82.6	95.7
	1.5	87.7	100.0	82.6	95.7
	2.0	83.6	100.0	82.6	95.7
	2.5	83.6	100.0	82.6	95.7
	All 2D	87.5	100.0	83.3	95.8

3 codes

		TT	ETJC	JJXQ	Proposed
1D	0.0	100.0	100.0	100.0	100.0
	0.5	93.8	94.4	92.4	98.0
	1.0	92.4	86.7	89.1	98.0
	1.5	91.7	86.7	89.1	97.0
	2.0	86.3	82.9	90.2	97.0
	2.5	82.7	82.0	90.2	97.0
	All 1D	82.7	77.9	90.6	97.5
2D	0.0	100.0	100.0	100.0	100.0
	0.5	89.1	100.0	83.2	100.0
	1.0	87.7	100.0	82.5	100.0
	1.5	87.0	100.0	82.5	100.0
	2.0	85.9	100.0	74.5	100.0
	2.5	85.0	100.0	74.5	100.0
	All 2D	87.0	100.0	82.9	100.0

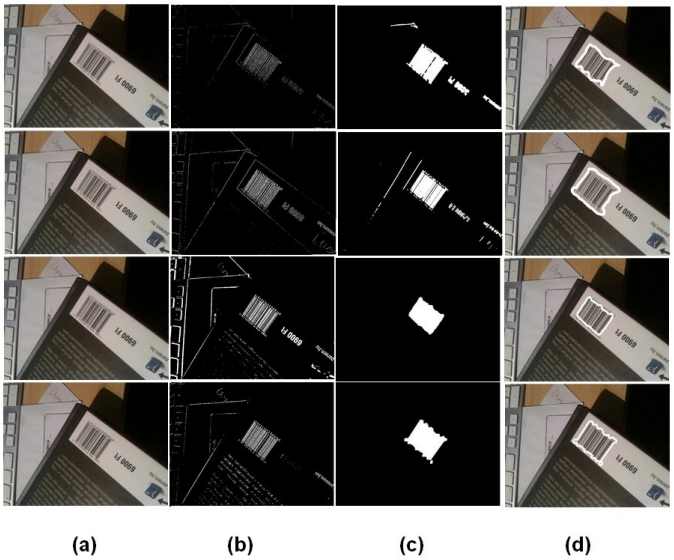


Fig. 5. Intermediate and final outputs of the algorithms at key phases. From top to bottom: TT, ETJC, JJXQ, Proposed. From left to right: original image (a), after preprocessing (b), before the last step (c), final output (d).

TABLE V
ACCURACY OF THE ALGORITHMS FOR DIFFERENT NOISE LEVELS, FOR IMAGES CONTAINING A SINGLE CODE AND FOR THOSE CONTAINING 3 CODES. MEAN VALUES (EXPRESSED IN PERCENT) FOR ALL DISTORTED IMAGES WITH 1D BARCODES (TOP), AND FOR ALL IMAGES WITH 2D CODES (BOTTOM).

1 code

		TT	ETJC	JJXQ	Proposed
1D	0	100.0	100.0	100.0	100.0
	10	90.2	89.5	93.9	97.4
	20	89.7	86.9	87.5	97.4
	30	89.4	86.2	79.2	97.4
	40	90.1	90.1	78.5	97.8
	50	93.3	84.2	69.9	94.9
	All 1D	90.5	88.6	83.5	96.8
2D	0	100.0	100.0	100.0	100.0
	10	87.7	100.0	84.7	94.4
	20	87.7	98.6	81.9	94.2
	30	87.7	100.0	79.3	94.2
	40	85.7	100.0	76.7	94.2
	50	77.8	98.6	69.6	92.3
	All 2D	85.8	99.5	77.3	94.8

3 codes

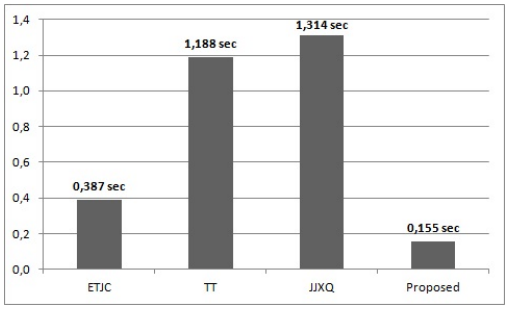
		TT	ETJC	JJXQ	Proposed
1D	0	100.0	100.0	100.0	100.0
	10	94.8	97.5	92.6	96.0
	20	89.4	87.6	89.5	96.3
	30	84.6	86.3	82.7	96.7
	40	87.2	84.9	79.9	96.7
	50	73.8	82.7	64.4	90.6
	All 1D	87.6	87.3	84.7	95.6
2D	0	100.0	100.0	100.0	100.0
	10	86.3	100.0	84.0	100.0
	20	85.8	100.0	78.9	100.0
	30	90.4	100.0	73.0	100.0
	40	86.3	100.0	72.6	100.0
	50	84.9	100.0	71.4	85.7
	All 2D	86.4	100.0	73.3	95.6

TABLE VI
ACCURACY OF THE ALGORITHMS FOR REAL-LIFE IMAGES CONTAINING A SINGLE 1D BARCODE. MEAN VALUES (EXPRESSED IN PERCENT).

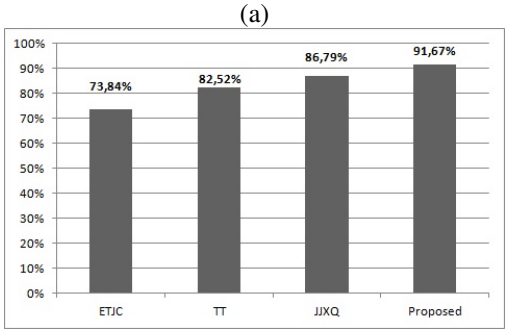
TT	ETJC	JJXQ	Proposed
92.0	91.4	92.0	94.0

IV. CONCLUSION

We have presented a novel barcode detection algorithm and compared its performance (in terms of accuracy and speed) with three methods from the literature. We concluded that these algorithms do not specialize for individual barcode types, they can efficiently detect various types of 1D and stacked 2D barcodes. For the evaluation, we created a test database containing 17280 synthetic images representing various degradations (blur, noise, shear) as well as 100 real images. We demonstrated that the algorithms selected from the literature are rather sensitive to noise, while the proposed new method is less sensitive. The proposed method outperforms the other three in detecting 1D codes both in terms of accuracy and speed.



(a)



(b)

Fig. 6. Running time and accuracy of the algorithms. Mean values for all generated and distorted images.

V. ACKNOWLEDGMENTS

The work of the second author is supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

REFERENCES

[1] R. C. Palmer, *The Bar Code Book: Reading, Printing, and Specification of Bar Code Symbols*. Helmers Pub, 1995.

[2] R. Adelmann, "Toolkit for bar code recognition and resolving on camera," in *Phones - Jump Starting the Internet of Things. In: Informatik 2006 workshop on Mobile and Embedded Interactive Systems*, 2006.

[3] T. R. Tuinstra, "Reading barcodes from digital imagery," Ph.D. dissertation, Cedarville University, 2006.

[4] E. Tekin and J. M. Coughlan, "An algorithm enabling blind users to find and read barcodes," in *Applications of Computer Vision (WACV), 2009 Workshop on*, 2009, pp. 1–8.

[5] X. Q. James Juett, "Barcode localization using bottom-hat filter," *NSF Research Experience for Undergraduates*, 2005.

[6] P. Bodnár and L. G. Nyúl, "Efficient barcode detection with texture analysis," in *Signal Processing, Pattern Recognition, and Applications, Proceedings of the Ninth IASTED International Conference on*, 2012, pp. 51–57.

[7] ——, "Improving barcode detection with combination of simple detectors," in *The 8th International Conference on Signal Image Technology (SITIS 2012)*, 2012, accepted for publication.

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

[9] D.-T. Lin, M.-C. Lin, and K.-Y. Huang, "Real-time automatic recognition of omnidirectional multiple barcodes and dsp implementation," *Machine Vision and Applications*, vol. 22, pp. 409–419, 2011.

[10] 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*, 2008, pp. 1–4.

[11] F. Y. Shih, *Image Processing and Mathematical Morphology: Fundamentals and Applications*. CRC Press, 2009.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2007.