Computer Science and Information Systems 17(1):161–179 https://doi.org/10.2298/CSIS181011020K

Distance Transform and Template Matching Based Methods for Localization of Barcodes and QR Codes

Melinda Katona, Péter Bodnár and László G. Nyúl

Department of Image Processing and Computer Graphics
University of Szeged
Árpád tér 2., H-6720 Szeged, Hungary
{mkatona, bodnaar, nyul}@inf.u-szeged.hu

Abstract. Visual codes play an important role in automatic identification, which became an inseparable part of industrial processes. Thanks to the revolution of smartphones and telecommunication, it also becomes more and more popular in everyday life, containing embedded web addresses or other small informative texts. While barcode reading is straightforward in images having optimal parameters (focus, illumination, code orientation, and position), localization of code regions is still challenging in many scenarios. Every setup has its own characteristics, therefore many approaches are justifiable. Industrial applications are likely to have more fixed parameters like illumination, camera type and code size, and processing speed and accuracy are the most important requirements. In everyday use, like with smartphone cameras, a wide variety of code types, sizes, noise levels and blurring can be observed, but the processing speed is often not crucial, and the image acquisition process can be repeated in order for successful detection.

In this paper, we address this problem with two novel methods for localization of 1D barcodes based on template matching and distance transformation, and a third method to detect QR codes. Our proposed approaches can simultaneously localize several different types of codes. We compare the effectiveness of the proposed methods with several approaches from the literature using public databases and a large set of synthetic images as a benchmark. The evaluation shows that the proposed methods are efficient, having 84.3 % Jaccard accuracy, superior to other approaches. One of the presented approaches is an improvement on our previous work. Our template matching based method is computationally more complex, however, it can be adapted to specific code types providing high accuracy. The other method uses distance transformation, which is fast and gives rough regions of interests that can contain valid visual code candidates.

Keywords: barcode localization, QR code localization, feature extraction, distance transform, template matching.

1. Introduction

Item identification using visual codes is popular in our everyday life, and there are several methods available for the process to be fast and reliable. The retrieval of the embedded data takes place in two steps. First, we have to find the visual code object within the acquired sensor data or image (localization step), then we have to use the symbology of the code and recognize the embedded data (decoding step). Decoding is widely studied, so we can use many approaches from the literature [8,10,19,23,27], or public APIs like the ZBar library[1].

It should be emphasized that decoding is far more straightforward, while the issue of localization is similar to object recognition and is still not fully solved. For localization of the code object, most algorithms use segmentation techniques with different features. Several applications simply ignore the localization step by adding a fast rotating laser that scans in many directions. Also, false positives are not acceptable, but the checksum digit (barcode) and the error correction (QR code) make false positives very unlikely in practice.

Visual codes are not meant to be readable for humans, they are decoded by specific devices. The most popular 1D barcode subtypes are the EAN-13 and UPC standards. These are widely used in commerce, like on wrapping of products, and they help quickly obtain the information on e.g. the producing country, types of entities of products. The flow of information is greatly boosted using visual codes, which provide decoding of the embedded data by electronic devices. Some types include features that also help their localization. The traditional 1D barcode structure is simple: a sequence of parallel light and dark bars of varying thickness represent information. The literature sometimes refers to 2D codes as “barcodes”, however, they do not necessarily consist of “bars”. They carry the embedded data along two axes, and their most popular types are QR code and Data Matrix. Some 1D and 2D codes are presented in Fig. 1. In addition, these classical visual codes can also be produced in a way that they become unique and thus can be used to validate originality or authenticity. For example, in our previous work [14], we focused on automatic localization of glitters used as a certain kind of Natural Feature Identifier (NFI).

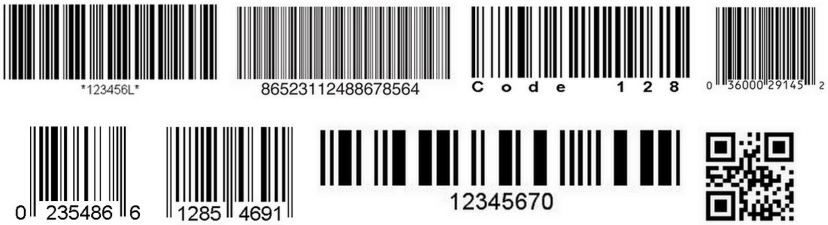


Fig. 1. Popular barcode types (from left to right). Top row (1D codes): Code39, Codabar, Code128, UPC-A; Bottom row (1D codes, 2D code): UPC-E, EAN-8, I2of5, QR code.

The use of visual codes has a reputation of more than 50 years, however, in the past, the localization process required many conditions to fulfill. The first barcodes were in a fixed position on railway trucks and were read by a fixed sensor gate. As technology progressed, PoS terminals appeared, still requiring human intervention to perform code reading. In the ’70s, new algorithms have been developed that could localize codes having various orientation and position within the image. The first approaches were very simplistic, they imitated the laser scanners of the barcode reading device. From the ’90s, machine

[1] publicly available at http://zbar.sourceforge.net/

learning provided some more sophisticated solutions for the issue. Methods providing automatic code localization are usually slower, but more accurate than their predecessors. Accuracy and processing speed are conditions that can hardly be fulfilled simultaneously, and most approaches aim to find a balance between these. Some machine learning algorithms make an exception, and they are capable of a quick evaluation after a significantly slower learning process, provided that the features can be computed efficiently and there is sufficient amount of training data available.

In industrial applications, accuracy is more crucial, since missed codes may lead to loss of profit. In those cases, speed is a second desired attribute, while in smartphone applications accuracy is not as critical, because the user handles the device interactively, and repetition of the image acquisition is possible and relatively easy.

There are numerous methods for the localization of visual codes in digital images, some imitating the classical laser scanner. Adelmann et al. [1] introduced a barcode recognition and information system to detect and read EAN-13 barcodes. This system works as a mobile application and traditional and widely used. After some preprocessing steps, Ohbuchi et al. [18] used a scanline based procedure to detect QR and EAN codes.

The toolkit of mathematical morphology has been used in many approaches in the literature. Bodnár et al. explained that using simple detectors [4] such as combination of different morphological operators and distance map to detect barcodes efficiently. Furthermore, texture analysis [3] can also achieve great efficiency. Similarly, Katona et al. [12] showed a method that relies on simple morphological bottom-hat filtering after a pre-processing step that highlights the bars of the barcode. Later in their work [13], they used simple features to localize a barcode areas. A distance map based approach was used as an extension of this process to merge split regions, which improves accuracy. Those regions, for example, arise from bad illumination, or flaws of the barcode material. Lin et al. [17] demonstrated a fast and effective method that can simultaneously detect 1D and 2D barcodes. Their method is based on a modified run length smearing algorithm. Kong [15] defines regions of interests that may contain QR codes in synthetic images with the mix of Harris corner detector and convex hull. In addition, they recommended a solution to correct for geometric distortion. Belussi et al. [2] introduced a machine learning method that is based on the locator pattern of the QR code. They proposed a cascade of weak classifiers using features from the Haar wavelet family. Although it is fast, it provides a noticeable amount of false positive code candidates. Bodnár et al. [6] proposed an improvement on that, using LBP and HoG features as an extension of the training step on the full code object. Sörös et al. [20] aim to localize 1D and 2D code using edge and corner maps, even considering the saturation channel in HSV images. Their algorithm is optimized on images suffering from heavy directional smoothing [21]. The method has high accuracy, in cases however, where the code object is surrounded by text, their approach provides oversized bounding boxes. Text filtering can help get rid of this problem, considering the surrounding text as a priori information. Szentandrási et al. [22] also work with edge and corner maps and HoG features. Their method works locally on square image cells, similarly to convolution. This approach enables parallel execution and it is also highly accurate.

Yun et al. [28] introduced an orientation histogram-based method. They used a histogram to the principal orientation components from the entire image and calculate the local entropy of the orientation to generate a saliency map. Bodnár et al. [5] presented a method based on distance transformation. The algorithm also considers local image blocks and evaluates the distance map of the edge map. It takes into account the mean and standard deviation of the distance values within each block, then makes a binary decision whether or not the block contains a barcode part. While this feature can be computed efficiently, it has weak classification power, therefore it is not sufficient for use alone for the localization step. In their work, the authors tried to overcome this attribute using morphological operations.

Many recent papers use machine learning methods to solve various image processing problems. Hansen et al. [11] used a deep learning object detection algorithm, namely You Look Only Once (YOLO) model. Their network is based on a pre-trained Darknet19 model with 6000 epochs. The most common architecture for semantic segmentation is the U-net that has different variants for each task. Ventsov et al. [26] divided the input image into 128×128 blocks, extracted statistical characteristics for each block and trained a convolution neural network. A Region-based Convolutional Neural Network (R-CNN) model was proposed by Ban et al. [25] for detecting diversified barcodes under complex scenes. They used for experiments two pre-trained model, ImageNet and VGG16.

In this paper, in Sec. 2.2, we present an improvement to this latter approach, giving a feature that can also be computed using the distance map and it also considers direction information. Also, instead of using only statistical values, we propose to use the whole distribution vector and make the final decision with SVM. The feature is computed locally, and in the final step, the accuracy is further improved by processing the feature matrix. This shows good performance for 1D codes, but on 2D codes, it is not sufficiently accurate. We also present two algorithms based on template matching, one suitable for efficiently localizing 1D barcodes (Sec. 2.1), and the other suitable for QR code detection (Sec. 2.3).

2. Methods

Although the imaging quality of recent digital cameras is high, lower quality images may be acquired as well due to various circumstances, such as dust, humidity, shaking of the camera in low-light situations. Due to this, preprocessing of the input images is usually necessary before code localization. In this section, we present three barcode detection algorithms. They use different classical operations to find the barcode in the image.

2.1. 1D barcode localization using pattern matching

In this section, we present a novel method for the localization of 1D barcodes based on pattern matching. The overview of the algorithm is presented in Fig. 2, while particular steps are illustrated in Fig. 3 and described below.

During processing, the input images can be of different sizes. We reduce the height of the image to a fixed size of 500 pixels in order to make them more uniform, easier to handle, and make processing faster. Empirical experience has shown that this is the smallest image size where smaller area code regions can still be localized. We did not use color information during the process, so the input RGB image was converted to grayscale. Input images are often blurry, therefore we use sharpening (Fig. 3(b)).

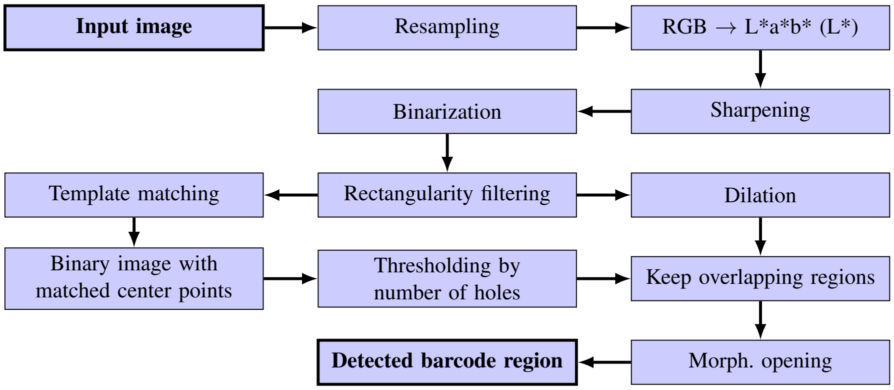


Fig. 2. The barcode localization process using template matching

The detection process is based on binary images, so the image is binarized using a global threshold. In our case, this value was 4% of the maximum intensity (Fig. 3(c)). This threshold was chosen empirically, based on the observation that white parts of the code more often fall into the gray intensity range because of dust, cheap quality labels or shapes being present because the packaging does not necessarily use white color for the bright parts of the code. This is not a robust solution, but selecting valid regions from false regions is easier than finding a missing part during a post-processing process. Obviously, a filter step is needed to reduce the number of false regions where different noise, etc., can occur. The shape of the bars of a barcode are rectangular, so we examine the shape of each object. If the shape of the object is not approximately rectangular, we do not consider it as a candidate region. The examined barcodes have a specific structure, so template matching is a possible way to detect the bars of the barcode. The input for pattern matching is illustrated in Fig. 3(d). As barcodes consist of parallel "bars", the template consists of two parallel lines. Traditional barcodes consist of a plurality of parallel lines, so a similar part of the image may be suitable for template matching. We also know the maximum and minimum distance between bars for each type of code. The template image was selected based on this information.

The process of template matching occurs in the frequency domain using Fast Fourier Transformation. The complexity of template matching in Fourier domain is $\mathcal{O}(n \log n * n_2)$, where n is the data size. We rotate the template image in every 10°, up to 170°, and compute the sum of the pointwise multiplication of the frequency representation of the original image and the rotated template images. 10° step was empirically found to be sufficient because the efficiency of this method had its maximum at around 15°. Thanks to the symmetric nature of the matched template, it is sufficient to examine only the aforementioned rotations. The summarized feature image is then thresholded with the mean of the summarized value.

Next, we use only the center of the objects that are being obtained. Pixels belonging to a specific cluster are well-separable like the bars of the barcode that are close to each other. We used the well-known kNN clustering method to separate connected objects from each other. The set of points from template matching can be well separated, so we have

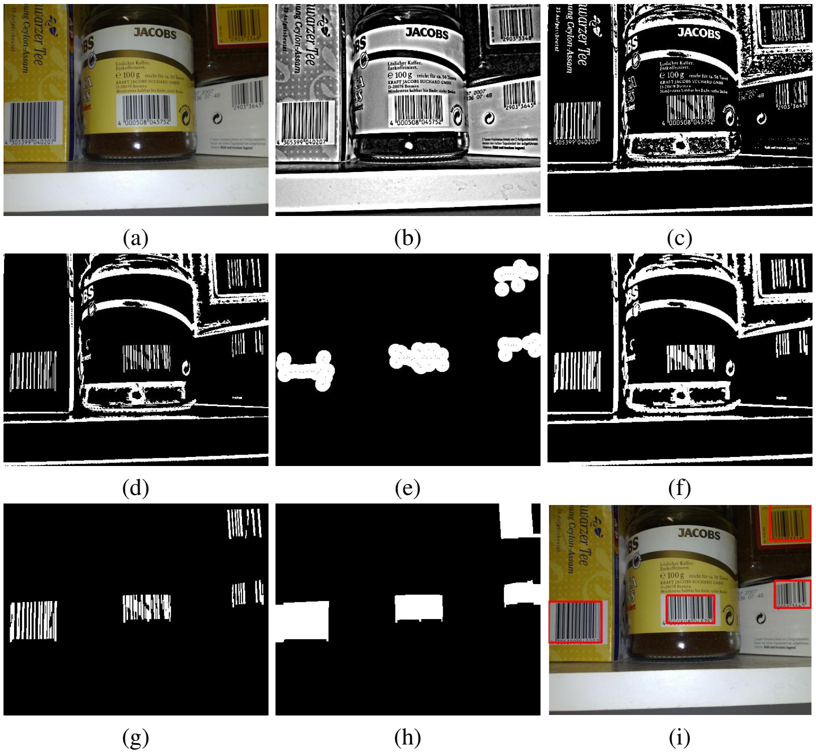


Fig. 3. The proposed method for 1D barcode localization. (a) input image, (b) deblurring, (c) binarization, (d) filtering by rectangularity, (e) thresholding using the number of cluster points, (f) result of (d) after dilation, (g) matching objects with (e) on image (f), (h) opening, (i) code candidate boxes overlaid onto (a) in red

chosen $k = 3$. To keep valid regions, we use a priori information that a barcode consists of at least 8 bars, so only those clusters are kept that have at least 8 points (3(e)). The complexity of kNN is $\mathcal{O}(ndk)n$, where n denotes the number of training points, d is the number of dimensions and k is the number of iterations.

We apply morphological dilation on the binary version of the original image, using a 3×3 structuring element (Fig. 3(f)). We investigate objects between the dilated image and filtered cluster points and we only keep overlapping regions (Fig. 3(g)). In order to determine the whole barcode region, we use morphological opening with a square-shaped structuring element. The size is defined based on the maximal distance between the stripes, which provides that every barcode will have its own connected region (Fig. 3(h)). The complexity of morphological operations depends on the used operator.

2.2. Barcode localization using distance transformation

The proposed algorithm is based on a feature derived from distance transformation. First, the Canny edge map is produced, then distance transformation is performed, where every point gets the distance value from its closest edge point. During this computation, we propose to also register the angle of each corresponding edge point, which reinforces the feature.

A priori information is needed regarding the element size of the visual code. “Bar thickness” of the barcodes may vary with respect to the distance of the code object and the camera, however, a range can be given for the expected thickness of the bars. Typically, the thinnest bar of a barcode is 1–3 px, while the thickest is 5–15 px wide in a case where the data is reasonably retrievable. With QR codes, typical element size is 5–30 px, regarding public code databases.

We propose to work on the brightness channel of the color space (V in HSV or L* in L*a*b*). As a preprocessing step, contrast stretching is performed and Gauss smoothing is applied, with a kernel size depending on the image resolution and expected bar thickness. For the aforementioned case, a 3×3 or 5×5 kernel is appropriate. After smoothing, the Canny edge map is produced as the hysteresis thresholding of the Sobel’s x and y gradients. The method greatly helps localization because it produces thin, connected edges. Those edge points are the marked points for the distance transformation. We also calculate the direction to the closest corresponding point. This approach produces similar “zones” like the Voronoi diagram (Fig. 4).

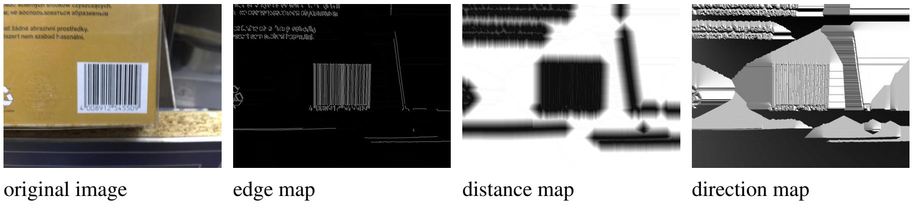
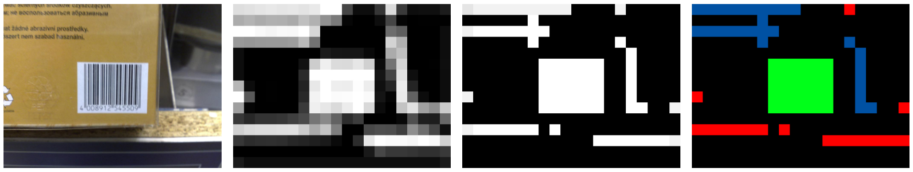


Fig. 4. Sample from the Muenster data set and its corresponding feature images

The feature image is divided into disjoint square blocks as the next step, then we calculate the distribution of the distance and angle values, aggregated in a predefined number of bins. The two vectors are then concatenated and fed into an SVM that learns a binary classification. Although we could give the raw pixel data to the SVM, it is less efficient than the aforementioned features that take advantage of spatial information. The prediction of the SVM will give an answer to the question of whether or not an image block contains part of a barcode. Such binary value is assigned to each block forming a feature matrix.

In the next step, connected components of the feature image are determined. Components are filtered by size, as barcode bar width gives a range for expected minimum and maximum code size. Those code objects appear as connected components in the feature matrix. Compactness can be calculated as the proportion of the perimeter and area of a blob. The tolerance should be set according to the expected visual code type (bar width and the width-to-height ratio of the specific code). Fig. 5 shows an example for the feature image, its thresholded connected components and filtering by size and compactness.



original image feature image thresholding component classification

Fig. 5. Post-processing steps of the feature image

A rotated bounding rectangle is given for the components that meet the aforementioned conditions. After that, we look for a homography with a properly oriented rectangle having the expected code size and proportions. The decoder gets the rectangular area from the image with the inverse homography applied.

In a previous work [5] the distance transformation was performed block-wise, which means the closest marked point was only searched for within the block. It is more appropriate to do the distance transformation before the tiling because we can also find the closest corresponding points in neighboring blocks and this helps the training process of the SVM. Block size is not relevant for the distance feature itself, however, it should be selected so that most bins of the distribution contain a sufficient number of samples. The rule of thumb for binning declares that n is a proper choice for the number of bins if we have at least n^2 samples. According to that, a distribution of 16 bins defines a lower bound of 16×16 px block size. The upper bound for the block size is related to the code size. In order to successfully detect a code object, at least 15–20 blocks are needed in the feature matrix for a code candidate. Fewer blocks would mean block length being bigger than 25–35 % of the code length along its dimensions, which decreases the occurrence of blocks that are full with code pattern only.

Blocks can be overlapping, but overlapping does not significantly improve the variability (and the learnability) of the distribution, and approximating the process of convolution only increases running time. Summary of the steps can be observed in Fig. 6.

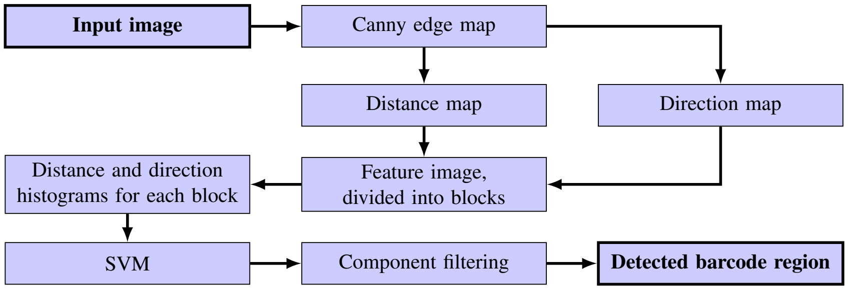


Fig. 6. Steps of the Distance Transform approach

2.3. QR code localization with template matching

In this section, we present a new method for localization of QR codes. The overview is given in Fig. 7, while the particular steps are illustrated in Fig. 8 and described below.

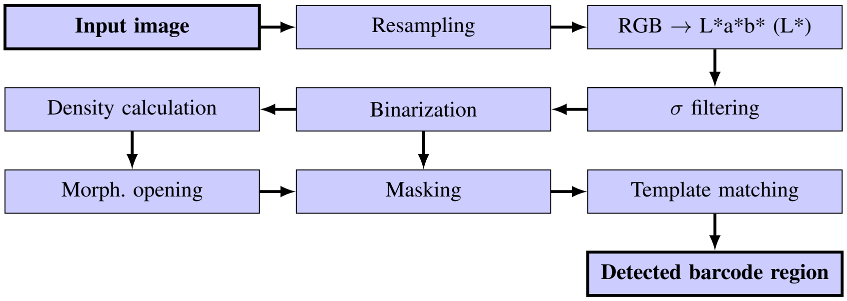
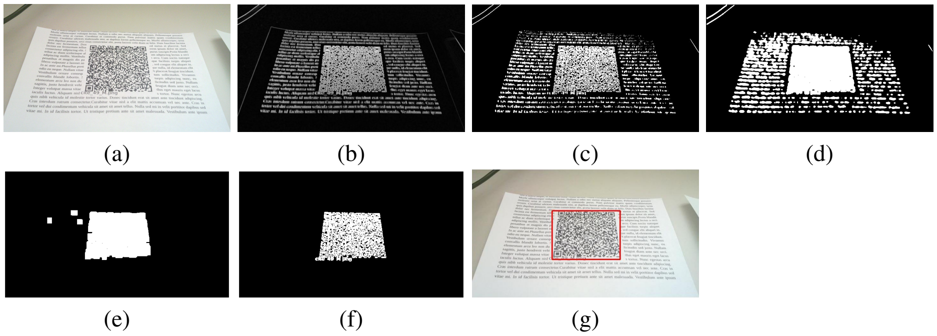


Fig. 7. The QR code localization process


(a) (b) (c) (d)

(e) (f) (g)

Fig. 8. Proposed method for QR code localization. (a) input image, (b) σ filtering, (c) binarization, (d) density calculation, (e) morphological opening, (f) post-processing, (g) detected QR code

Similarly to the procedure described in Sec. 2.2, we also work with images with specific size during the localization and convert the images to grayscale. In order to highlight the barcode areas, we use standard deviation based adaptive filtering method with 3×3 neighborhood [9]. The resulting image is heterogeneous, so we used density calculation with a fixed 7×7 kernel. We calculated the number of object points for every kernel and removed from candidate barcode regions where this value was under the half of the kernel size.

In order to remove false small regions and then merge the connected regions, we apply a morphological opening. The shape of the QR code is ideally a square, so we use a square shaped structuring element for the morphological operation. Based on empirical observations, we binarize the image obtained in the previous step with the threshold value of 7/8th of the maximum intensity. Since global thresholding is not an overly robust operation, but the bars of the barcode have low-intensity as usual (we supposed that barcode not colorful), so we can eliminate numerous false segments with a low-intensity value when determining global thresholding. In the last step, we validate the code segments by pattern matching. For this, we used a region from the inner box of a QR code as a sample. We do the template matching on the original image and we investigate overlapping with the opened binary image similarly as described in Sec. 2.2. Valid QR code regions are available after the validation step.

3. Evaluation and results

In this section, the proposed algorithms are compared against some effective ones from the literature. Several research groups [8,7,12,20,24,29,28,11,26] evaluated their algorithms on the WWU Muenster data set, therefore we also decided to use that set for evaluation, while the fine-tuning of parameters and some of our other tests were performed on our custom, synthetic image set[2].

3.1. Test suite and implementation

An artificial test set is created from the barcode examples presented in Fig. 1. One example is selected from each code type, and various distortions and levels of noise are applied. The generated images were rotated from 0° to 180° by 15°. Gaussian smoothing is applied with 3×3 kernel and 6 different σ values. Also, Gaussian noise was added from 0 % to 50 % with the step of 10 %. In total, we created 12 orientations from 8 types of barcodes, using 6 different smoothing and 6 different noise levels, with perspective distortions, counting as cca. 15 000 images. Fig. 9 illustrates some examples from our artificial data set. Furthermore, we used 1056 images of real barcodes from the WWU Muenster data set.

For the test set of QR codes, we used a database consisting of 1400 real images [20], and 10 000 synthetic test images. The latter set is generated similarly to the 1D barcode set. Fig. 10 shows some samples from that data set. The second public database by Dubská et al. contained two similar sets of QR code images, surrounded with text in a scene having low saturation in general. The first set has 410 high-resolution (2560 × 1440 px) images with uneven lighting conditions, high grades of distortion and minor blur. The second test set has 400 low-resolution (604 × 402 px) images with smaller grades of distortion and more even illumination, but having less light in general, thus producing darker images.

3.2. Figures of merit

To measure the efficiency of the algorithms, we compared the overlap of our segmentation output and the ground truth with the Jaccard similarity measure, defined as

$$J = \frac{TP}{TP + FP + FN},$$

[2] `http://www.inf.u-szeged.hu/~bodnaar/barcode_database/`

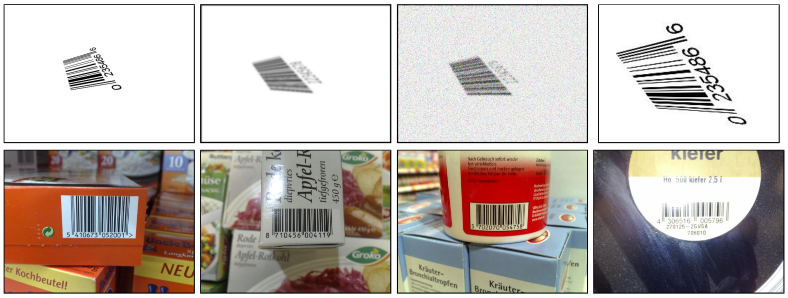


Fig. 9. 1D samples with different distortions, synthetic (first row) and real images (second row).

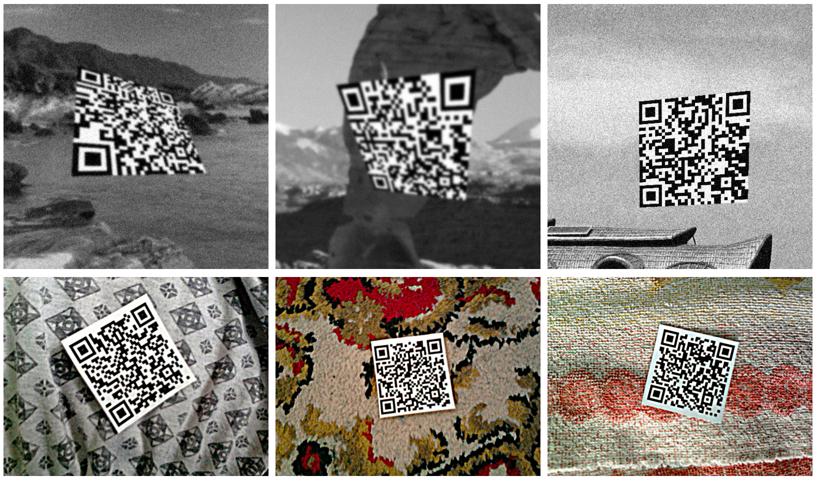


Fig. 10. Synthetic and real images with QR code

where TP denotes the correctly detected codes, FP is the number of not valid code regions and FN is the number of not localized codes. Note that, ground truth regions have a tight fitting bounding polygon showing the code object without numbers and “quiet zones” as a border. A successful detection is where $J > 0.5$, according to the work of Szentandrási et al. [22].

3.3. Parameters for the distance transformation approach

The fine-tuning of the SVM parameters were performed using a subset of the Muenster database since with 1D cases, the visual structure of the code and the distribution of the angles is more prominent.

As the first step of the evaluation, we examined how SVM accuracy is influenced by the number of bins. We performed separate trainings using only the distributions of

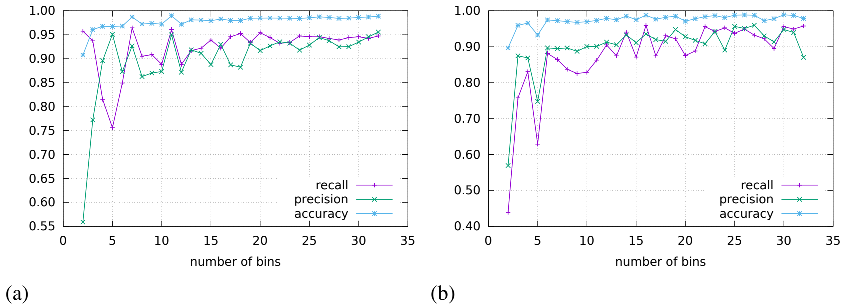


Fig. 11. Efficiency of the Distance Transformation method w.r.t. the number of distance (a) and direction (b) bins

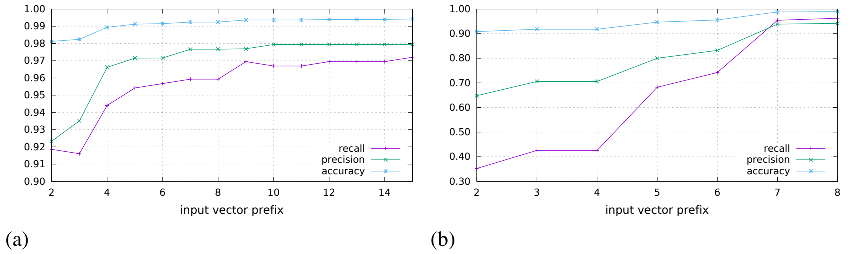


Fig. 12. SVM efficiency w.r.t. the used prefix of the distance (a) and direction (b) histograms

distance values (Fig. 11(a)) and angles (Fig. 11(b)). We can conclude that more than 16 distance classes do not significantly improve accuracy. With angles, 8 classes show the highest classification power.

Also, we examined feeding only a prefix of a distribution of distances or angles, which shows how important the individual bins are. Intuitively, we shall expect that for the distance values, the first few bins are important, because the edge points are close to each other in barcodes, and this characteristic contributes the most to the classification power. Our results confirm this assumption (Fig. 12(a)).

For the directions we shall expect that the elements of the distribution are equally important, so the number of bins is linearly correlated to accuracy (Fig. 12(b)).

We examined the accuracy considering only distance values, only angles, and both distance and angle simultaneously. Results are shown in Table 1. The highest accuracy is obtained when both directions and angles are used for the training.

We also evaluated the training accuracy using a small number of training samples. Results are shown in Fig. 13. The whole Muenster database with a given block size of 50×50 px contains cca. 300 000 input vectors, about 10 % of them labeled as positive. The SVM classes should be weighted according to that proportion. We used 10-fold cross-

Table 1. SVM training performance for various features

input	recall	precision	accuracy
directions only	0.9625	0.9416	0.9893
distances only	0.8169	0.8978	0.9698
both dir. & dist.	0.9597	0.9871	0.9942

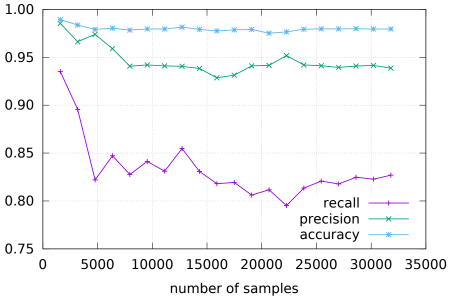


Fig. 13. SVM efficiency w.r.t. the number of used training samples

validation for validation purposes. The decrease in accuracy at the beginning of the graph indicates that a small number of samples are more separable.

We implemented the algorithm using OpenCV, with the automatic SVM optimizer option. Optimal parameter set means that the error is minimal during the cross-validation. Additionally, it shall be noted that considering the directions along with the distances does not increase running time significantly, because, during the 2-pass calculation of distance values, the directions can also be recorded. Regarding memory usage, the space needed to store the angles is similar to that for the original image.

3.4. Comparison

Our approaches were compared against other algorithms of the literature using the Muenster database as a benchmark. Results are shown in Table 2. Considering the proposed method with distance transformation (PROP-DT, Sec. 2.2), it shall be noted that the earlier method based only on distance values can only be used as a weak classifier, but this improvement with the directions and the SVM trained on the whole distribution makes it a usable state-of-the-art solution. Only the algorithm of Tekin el al. [24] has better mean value, however, with higher variance. For the comparison, we selected algorithms based on various features, like edge and corner maps [20], or deformable templates [8]. Zamberletti et al. [29] work with the popular localization method that is also implemented in ZXing barcode reading framework, while Creusot et al. [7] have an approach that uses MSER. Our template matching based approach (PROP-TM, Sec. 2.1) is even more specific to the barcode localization issue, therefore it shows even better performance than PROP-DT. Hansen et al. [11] reported 0.87 Jaccard value to result in their article, but we did not use this result in our comparison, because there is some missing information about training parameters, so we were unable to re-implement and evaluate the procedure.

Table 2. Comparison of various localization algorithms on the Muenster data set. (Mean and standard deviation of Jaccard index.) Best performing method is typeset in bold.

Algorithm	$\overline{J}$	st.dev.
Zamberletti et al. [29]	0.6950	N/A
Creusot et al. [7]	0.7990	N/A
Gallo et al. [8]	0.7089	0.3542
Tekin et al. [24]	0.8122	0.2562
Katona et al. [12]	0.5200	0.2967
Sörös et al. [20]	0.6647	0.2277
Yun et al. [28]	0.4716	0.2240
PROP-DT (Sec. 2.2)	0.8104	0.1944
PROP-TM (Sec. 2.1)	**0.8430**	**0.1876**

Most of the presented methods have the main goal as to highlight the bars of the barcode (gradiens calculation, bottom-hat filtering, etc.). The algorithms that are based on simple image operations and logic are examined on the synthetic image database under different conditions. We investigated and found that the procedures are not sensitive to rotation. We also examined the results with different noise and blur levels. In both cases the generated images had 6 different levels for those parameters, modifying the σ value of Gaussian smoothing and a γ weight for weighted addition of uniform noise to the original image. We present the obtained Jaccard indexes in Table 3. It shows that the efficiency of the procedures is hardly reduced by increasing the level of noise and smoothing. However, in all cases, the methods do not behave significantly differently on ideal cases. We used 8 different codes to generate the images. The procedures were less successful in localization task for these three types (Code-39, I2of5, UPC-E) as shown in Table 4. We also present some qualitative results of the aforementioned approaches on challenging images. See 14 for details.

In the 2D case, we evaluated PROP-DT on the two data sets of Dubská et al. [22], and obtained $\overline{J}_{set2} = 0.7588$ and $\overline{J}_{set1} = 0.5592$ as shown in Table 5. This means that distance transformation cannot be used for 2D code localization because of the low level of separability of distance and angle values. However, our template matching approach (PROP-TMQR, Sec. 2.3) with 2D QR codes performed well on the Dubská data sets, with $\overline{J}_{set1} = 0.8315$ and $\overline{J}_{set2} = 0.8102$. Hansen et al. [11] published 0.73 average Jaccard value on Dubská sets.

We also compared the results on synthetic images along different blur and noise levels (Table 6).

Table 3. Accuracy of the methods for different blur (σ value) and noise levels (in percent) on 1D synthetic images (mean Jaccard index).

Blur	Noise	Gallo [8]	Yun [28]	Sörös [20]	PROP-DT	PROP-TM
0	0	0.74	0.64	0.75	0.83	0.82
	30	0.75	0.66	0.76	0.83	0.82
	50	0.75	0.65	0.77	0.83	0.81
3	0	0.77	0.64	0.74	0.75	0.82
	30	0.75	0.67	0.76	0.83	0.81
	50	0.73	0.63	0.76	0.77	0.78
5	0	0.76	0.65	0.74	0.73	0.82
	30	0.74	0.66	0.76	0.83	0.81
	50	0.73	0.63	0.75	0.77	0.78

Table 4. Accuracy of the algorithms for various types of code on 1D synthetic images (mean Jaccard index).

	Gallo [8]	Yun [28]	Sörös [20]	PROP-DT	PROP-TM
Codabar	0.81	0.71	0.8	0.73	0.87
Code-128	0.82	0.73	0.83	0.78	0.92
Code-39	0.57	0.52	0.59	0.84	0.72
Ean-13	0.82	0.69	0.81	0.85	0.83
EAN-8	0.81	0.70	0.81	0.91	0.83
I2of5	0.69	0.6	0.69	0.77	0.72
UPC-A	0.78	0.66	0.78	0.84	0.80
UPC-E	0.69	0.64	0.74	0.74	0.73

Table 5. Comparison of various localization algorithms on the QR synthetic database (mean Jaccard index).

input	Ohbuchi [18]	Lin [16]	PROP-DT	PROP-TMQR
Dubska set1	0.79	0.81	0.56	0.83
Dubska set2	0.77	0.79	0.76	0.81

Table 6. Accuracy of the methods for different blur (σ value) and noise levels (in percent) on QR synthetic images (mean Jaccard index).

Blur	Noise	Ohbuchi [18]	Lin [16]	PROP-DT	PROP-TM
0	0	0.96	0.50	0.77	0.99
	50	0.96	0.50	0.77	0.99
	100	0.96	0.50	0.77	0.99
1.5	0	0.57	0.51	0.67	0.77
	50	0.52	0.49	0.67	0.78
	100	0.49	0.40	0.70	0.75
3	0	0.47	0.42	0.65	0.83
	50	0.48	0.45	0.65	0.80
	100	0.45	0.37	0.64	0.76

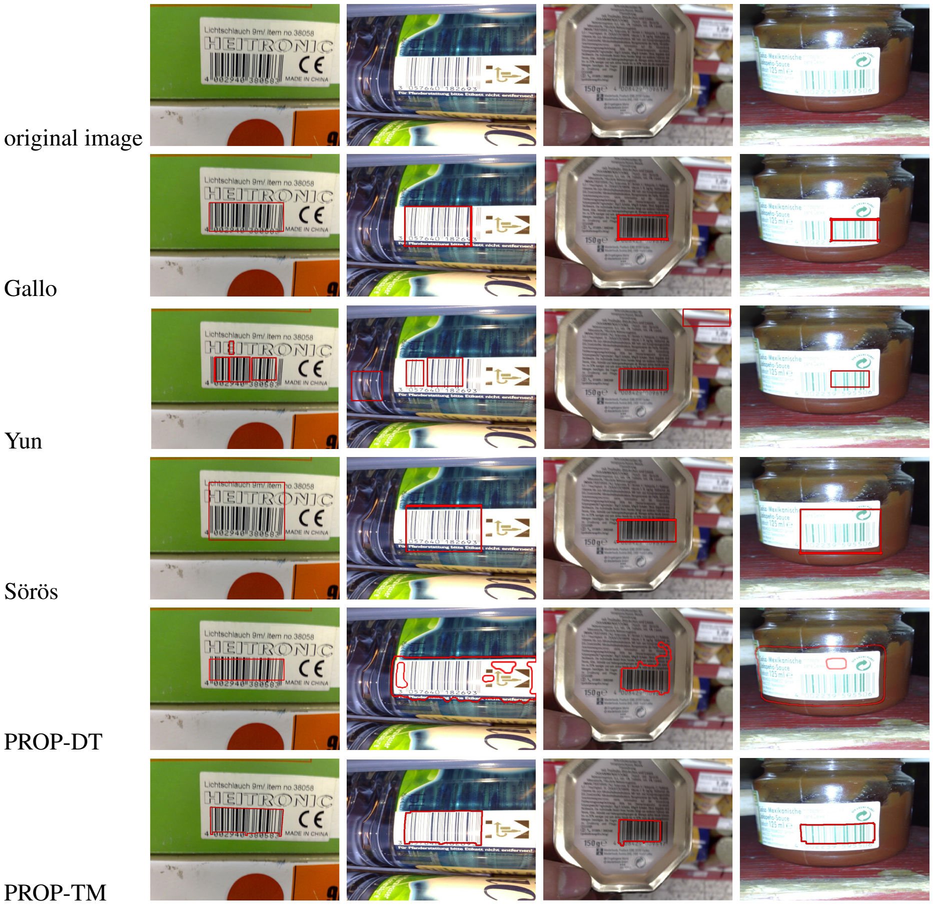


Fig. 14. Qualitative results for the compared methods on some challenging 1D images.

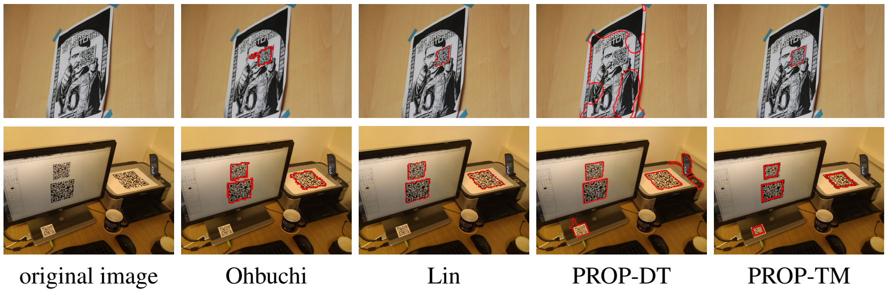


Fig. 15. Qualitative results for the compared methods on some challenging QR images.

4. Conclusion

Three novel approaches were presented, two for 1D barcode localization, and one for 2D QR codes. They are compared to algorithms from the literature, some of them being universal (working on both 1D and 2D codes), others specialized for either 1D or 2D. For the evaluation of efficiency, we generated data sets containing a large number of synthetic images. Results indicate that the proposed algorithms are efficient, even in cases where the visual codes suffer from perspective distortion.

Distance transformation was used for barcode localization in [5], however, it only could be considered as a weak classifier. Adding angle information to the feature, accuracy improves significantly. Distance transformation can be used as a standalone feature for the problem, with various code types. The approach also has a disadvantage, namely, it cannot be tuned to consider sophisticated structural details of different code types, like template matching.

In specific cases, when we can define more assumptions on the expected code type, size, or orientation, template matching can outperform general purpose solutions. However, in most applications, we cannot make very specific assumptions. Nevertheless, for some industrial applications, where the code properties fall in narrow ranges, the concept of template matching can be useful. In more general cases, the method based on distance transformation performs well, and it only contains computationally simple steps. Also, template matching might need different templates depending on the code type to find.

The proposed methods can be implemented to run in real time. The methods that use SVM and distance transformation take cca. 250 ms for an image of size 1024x768 px. The template matching based method's running time is 680 ms for an image in the case of 1D barcodes, and 419 ms for QR codes.

Acknowledgments.

This work was supported by the project “Integrated program for training new generation of scientists in the fields of computer science”, No. EFOP-3.6.3-VEKOP-16-2017-0002. The project has been supported by the European Union and co-funded by the European Social Fund. Melinda Katona was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4.A/2-11-1-2012-0001 'National Excellence Program'. The authors would like to thank Dr. Attila Tanács for his detailed and helpful comments and suggestions.

References

1. Adelmann, R., Langheinrich, M., Flörkemeier, C.: A toolkit for bar-code-recognition and -resolving on camera phones – jump starting the internet of things. In: In Workshop Mobile and Embedded Interactive Systems (MEIS'06) at Informatik (2006)
2. Belussi, L.F.F., Hirata, N.S.T.: Fast QR Code Detection in Arbitrarily Acquired Images. In: Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on. pp. 281–288 (2011)
3. Bodnár, P., Nyúl, L.G.: Barcode Detection with Morphological Operations and Clustering. In: Signal Processing, Pattern Recognition, and Applications, Proceedings of the Ninth IASTED International Conference on. pp. 51–57 (2012)
4. Bodnár, P., Nyúl, L.G.: Improving Barcode Detection with Combination of Simple Detectors. In: 8th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2012. pp. 300–306 (2012)
5. Bodnár, P., Nyúl, L.G.: Barcode detection with uniform partitioning and distance transformation. IASTED International Conference on Computer Graphics and Imaging pp. 48–53 (2013)
6. Bodnár, P., Nyúl, L.G.: QR Code Localization Using Boosted Cascade of Weak Classifiers. In: Image Analysis and Recognition, pp. 338–345. Springer International Publishing (2014)
7. Creusot, C., Munawar, A.: Real-Time Barcode Detection in the Wild. In: 2015 IEEE Winter Conference on Applications of Computer Vision. pp. 239–245 (Jan 2015)
8. Gallo, O., Manduchi, R.: Reading 1D Barcodes with Mobile Phones Using Deformable Templates. IEEE Trans. Pattern Anal. Mach. Intell. 33(9), 1834–1843 (2011)
9. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using MATLAB. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2003)
10. Ha, J.E.: A new method for detecting data matrix under similarity transform for machine vision applications. International Journal of Control, Automation and Systems 9, 737–741 (2011)
11. Hansen, D.K., Nasrollahi, K., Rasmussen, C.B., Moeslund, T.B.: Real-Time Barcode Detection and Classification using Deep Learning. In: IJCCI. pp. 321–327 (2017)
12. Katona, M., Nyúl, L.G.: 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. pp. 307–314 (2012)
13. Katona, M., Nyúl, L.G.: Efficient 1D and 2D Barcode Detection Using Mathematical Morphology. In: Mathematical Morphology and Its Applications to Signal and Image Processing, Lecture Notes in Computer Science, vol. 7883, pp. 464–475. Springer Berlin Heidelberg (2013)
14. Katona, M., Nyúl, L.G.: Fast Recognition of Natural Feature Identifiers by a Mobile Phone. Acta Cybernetica 22(1), 101–116 (2015)
15. Kong, S.: QR Code Image Correction based on Corner Detection and Convex Hull Algorithm. Journal of Multimedia 8, 662–668 (2013)
16. Lin, D.T., Lin, C.L.: Multi-symbology and Multiple 1D/2D Barcodes Extraction Framework. In: Proceedings of the 17th International Conference on Advances in Multimedia Modeling - Volume Part II. pp. 401–410 (2011)
17. Lin, D.T., Lin, C.L.: Automatic location for multi-symbology and multiple 1D and 2D barcodes. Journal of Marine Science and Technology 21, 663–668 (2013)
18. Ohbuchi, E., Hanaizumi, H., Hock, L.A.: Barcode Readers Using the Camera Device in Mobile Phones. In: Proceedings of the 2004 International Conference on Cyberworlds. pp. 260–265. CW '04 (2004)
19. Shams, R., Sadeghi, P.: Bar Code Recognition in Highly Distorted and Low Resolution Images. In: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. vol. 1, pp. I–737 –I–740 (2007)
20. Sörös, G., Flörkemeier, C.: Blur-resistant Joint 1D and 2D Barcode Localization for Smartphones. In: Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia. pp. 11:1 – 11:8 (2013)

21. Sörös, G., Semmler, S., Humair, L., Hilliges, O.: Fast Blur Removal for Wearable QR Code Scanners. In: Proceedings of the 2015 ACM International Symposium on Wearable Computers. pp. 117–124. ISWC '15, ACM (2015)
22. Szentandrási, I., Herout, A., Dubská, M.: Fast Detection and Recognition of QR Codes in High-resolution Images. In: Proceedings of the 28th Spring Conference on Computer Graphics. pp. 129–136. SCCG '12 (2013)
23. Tekin, E., Coughlan, J.: A Bayesian Algorithm for Reading 1D Barcodes. In: Proceedings of the 2009 Canadian Conference on Computer and Robot Vision. pp. 61–67. CRV '09 (2009)
24. Tekin, E., Coughlan, J.: BLaDE: Barcode localization and decoding engine. Tech. rep., Technical Report 2012-RERC (2012)
25. Tian, Y., Che, Z., Zhai, G., Gao, Z.: BAN, A Barcode Accurate Detection Network. In: 2018 IEEE Visual Communications and Image Processing (VCIP). pp. 1–5 (2018)
26. Ventsov, N.N., Podkolzina, L.A.: Localization of Barcodes Using Artificial Neural Network. 2018 IEEE East-West Design & Test Symposium (EWDTS) pp. 1–6 (2018)
27. Wang, K., Zou, Y., Wang, H.: Bar code reading from images captured by camera phones. In: 2005 2nd International Conference on Mobile Technology, Applications and Systems. p. 6 (2005)
28. Yun, I., Kim, J.: Vision-based 1D barcode localization method for scale and rotation invariant. In: TENCON 2017 - 2017 IEEE Region 10 Conference. pp. 2204–2208 (2017)
29. Zamberletti, A., Gallo, I., Carullo, M., Binaghi, E.: Neural Image Restoration for Decoding 1-D Barcodes using Common Camera Phones. In: Computer Vision, Imaging and Computer Graphics. Theory and Applications. pp. 5–11 (2010)

Melinda Katona received the M.S. degree in Computer Science in 2014 at University of Szeged, Hungary. He is currently an PhD candidate at Institute of Informatics, University of Szeged. Her research interests include pattern recognition and medical imaging.

Péter Bodnár graduated at University of Szeged in 2011 as technical informatician. After that, he was admitted the PhD fellowship of the Doctoral School in Informatics. After the three years of PhD studies, he was awarded the predoctoral fellowship in 2014. He wrote his thesis about localization of visual codes. He participates in various projects of the Department of Image Processing and Computer Graphics at University of Szeged since 2011. Besides that, he has participated in the teaching activity of the Institute of Informatics.

László G. Nyúl, received the M.S. degree in Computer Science in 1994, and the Ph.D degree in 2003 from the University of Szeged, Hungary. He has been a visiting research associate at University College London, University of Pennsylvania, Universität Erlangen-Nürnberg, Medical University Graz, and Uppsala Universitet. He is currently an associate professor and the Head of Department of Image Processing and Computer Graphics, and the Head of the Institute of Informatics at the University of Szeged, Hungary. His research interests include image processing, medical imaging, and pattern recognition.

Received: August 29, 2018; Accepted: September 10, 2019.