Recognition and Defect Detection of Dot-matrix Text via Variation-model Based Learning

Wataru Ohyama[a], Koushi Suzuki[a] and Tetsushi Wakabayashi[a]

[a]Mie University, Tsu, Mie, Japan

ABSTRACT

An algorithm for recognition and defect detection of dot-matrix text printed on products is proposed. Extraction and recognition of dot-matrix text contains several difficulties, which are not involved in standard camera-based OCR, that the appearance of dot-matrix characters is corrupted and broken by illumination, complex texture in the background and other standard characters printed on product packages. We propose a dot-matrix text extraction and recognition method which does not require any user interaction. The method employs detected location of corner points and classification score. The result of evaluation experiment using 250 images shows that recall and precision of extraction are 78.60% and 76.03%, respectively. Recognition accuracy of correctly extracted characters is 94.43%. Detecting printing defect of dot-matrix text is also important in the production scene to avoid illegal productions. We also propose a detection method for printing defect of dot-matrix characters. The method constructs a feature vector of which elements are classification scores of each character class and employs support vector machine to classify four types of printing defect. The detection accuracy of the proposed method is 96.68 %.

Keywords: Dot-Matrix text, Variation model based learning, Industrial OCR

1. INTRODUCTION

Important information such as the consumption or expiration date is often printed on product packaging using dot-matrix characters (see Figure 1(a)). These dot-matrix characters are printed directly on the product to inform both consumers and producers about the products. Automated optical character recognition (OCR) systems for dot-matrix characters on products may reduce labor costs required, especially in processes such as quality control.

For accurate OCR of these dot-matrix characters, several methods have been proposed[1,2] However, dot-matrix characters used in production vary widely in matrix font patterns, dot size, printing quality and degradation. This makes construction of a universal automated OCR process quite difficult. Furthermore, previous methods have typically handled either single-character recognition or text extraction. An end-to-end algorithm is required to establish a complete OCR system. It is also known that there are many undocumented dot-matrix recognition products for factory automation. Most simplify the recognition task by using a controlled environment to restrict the variation in the characters to be captured. It is relatively easy to recognize controlled dot-matrix characters using training dataset compiled in the same capturing environment. However, these systems are inflexible and are restricted to the environment they were designed for.

We previously proposed a technique to improve the accuracy and robustness of dot-matrix OCR using variation model based learning.[3] We evaluated the effectiveness of the proposed learning technique with a dataset containing 38 classes (2030 character samples) captured from real products. Although the technique improved recognition performance, dot-matrix characters were not consistently extracted accurately.

In this paper, we propose an end-to-end OCR method which extracts dot-matrix characters from a product image and recognizes the extracted characters. The proposed OCR method is based on corner detection and appearance evaluation for candidate character region. Applying a corner detection algorithm to the input image, many corner points, mostly false positives, are identified. Candidate regions are created by concatenating adjoining corner points following predefined hypotheses. The appearance of each character in the candidate region is then evaluated by a modified quadratic discriminant function (MQDF) classification score.[4] Thus, dot-matrix characters can be extracted without user interaction. The process of our proposed method and output in each step are shown in Fig. 1.

Further author information: (Send correspondence to Wataru Ohyama.)Wataru Ohyama: E-mail: ohyama@hi.info.mie-u.ac.jp

Thirteenth International Conference on Quality Control by Artificial Vision 2017, edited by Hajime Nagahara, Kazunori Umeda, Atsushi Yamashita, Proc. of SPIE Vol. 10338, 103380K · © 2017 SPIE
CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2264232

(a) Dot-matrix characters on a product (b) Corner points which have characteristics of dot-matrix (c) Candidate region of dot-matrix text (d) Result of extraction and recognition

Figure 1. Extraction and recognition of dot-matrix characters.

Figure 2. Examples of printing defects in dot-matrix characters; missing dot (0 and 1), missing line (2 and 3), ink bleeding (4 and 5) and spot dust in background (6 and 7).

Dot-matrix characters sometimes contain defects or errors caused by printing facilities. Because overlooking these printing defects can cause legal problems, it is important to detect them as early in production as possible. If printing defects can be detected automatically, the results can be used to feedback immediately to the printing facilities to avoid such legal problems. We propose a method to detect printing defects in dot-matrix characters. A feature vector is constructed from elements that are MQDF scores for each character class. A support vector machine is used to classify the constructed score vector to one of four types of printing defect. Fig. 2 shows examples of printing defects that commonly appear in dot-matrix characters.

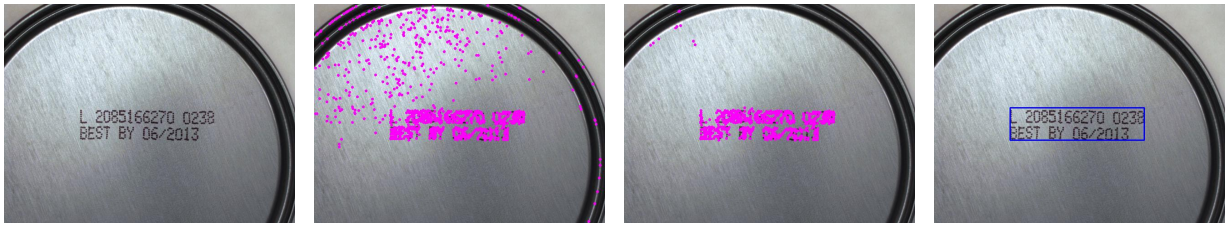
The rest of this paper is organized as follows: the detail of the proposed method is described in section 2; experiments for performance evaluation are shown in section 3; the results and discussion are given in section 4; and the conclusions are given in section 5.

2. PROPOSED METHOD

Our proposed method consists of the following steps:

1. Identification of character corner points: Corner points are detected from the entire image using the Features from Accelerated Segment Test (FAST) corner detector.[5] Corner points on dot-matrix characters are identified by appearance evaluation.
2. Identification of candidate text regions: Candidate text regions are identified by grouping character corner points.
3. Preprocessing (binarization and text-line detection): Candidate text regions are binarized and text-lines are detected for character extraction and recognition.
4. Character extraction and recognition: Dot-matrix characters are extracted from the text region and they are evaluated by the MQDF classifier trained using the variation model based learning method.
5. Elimination of false positives: False positives are eliminated using the MQDF score and positional relationships.
6. Detection of printing defect: The proposed method recognizes affected dot-matrix characters, and detects and classifies printing defects.

We detail each step in the following subsections.



(a) Original image (b) All detected corners by FAST algorithm (c) Identified candidate points (d) Grouped text regions

Figure 3. Identification of dot-matrix text candidate regions.

2.1 Identification of character corner points

Dot-matrix characters are constructed from small circular components; Therefore, corners in dot-matrix characters have a large gradient magnitude, multiple gradient directions and low saturation.

We detect corners from whole image using the FAST algorithm, a well-known corner detector. Corners detected by the FAST algorithm are expressed as $F_i = \{x_i, y_i, s_i, m_i, d_i\}$, where x_i, y_i, s_i, m_i are coordinates of corners, normalized saturation and mean gradient magnitude of 16×16 local area around the corner respectively. $d_i = \{d_{i0}, d_{i1}, \ldots, d_{i7}\}$ is direction histogram of grayscale gradient calculated in 8-direction quantization. Corners satisfying all of the following conditions are selected as candidate points c_i for constructing dot-matrix characters.

- $m_i > m_o$, where, m_o is the mean gradient of the whole image.
- $|d|_\infty < d_o + \sigma_d$, where, $|d|_\infty$ is the maximum, d_o is the mean and σ_d is the standard deviation of d_i.
- 16×16 local region around F_i contains at least one edge point detected by Canny operator.
- $s_i < 0.8$

All detected FAST corners are shown in Fig. 3(b). Candidate character corner points are shown in Fig. 3(c).

2.2 Identification of text candidate region

To identify text regions, we analyze the layout of texts on the product using candidate character points extracted in 2.1. Characters in one text are assumed to be printed in the same intensity and close to each other. Text regions are identified by classification of candidate character points using normalized histogram intersection. Candidate points on dot-matrix characters are grouped by the following steps to identify dot-matrix text candidate regions.

1. The input image is converted into a 16-level gray-scale image. Then, an intensity histogram $h(c_i)$ is created from a region 16×16 pixels centered on the candidate points c_i.
2. If an unclassified candidate point $c_a = (x_a, y_a)$ is found by raster scanning, a new label is attached to c_a and the process is continued from step 3, otherwise it is continued from step 5.
3. Raster scanning is continued from c_a to find another unclassified point c_b. If Eq. (1) is satisfied by c_a and c_b, c_b is given the same label as c_a, and the process is continued from step 4. If Eq. (1) is not satisfied, c_b is skipped and the raster scanning is continued. When the raster scanning reaches the bottom-right of the image, the process is repeated from step 2.

$$D_{\mathrm{h}} > D_{\mathrm{e}} \tag{1}$$

where, D_{h} and D_{e} are calculated as follows.

$$\begin{aligned} D_{\mathrm{h}} &= h(c_a) \otimes h(c_b) = \frac{1}{16 \times 16} \sum_{j=0}^{15} \min\left(h_j(c_a), h_j(c_b)\right), \\ D_{\mathrm{e}} &= \frac{1}{n} \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \end{aligned} \tag{2}$$

4. The point c_b is considered a new base point c_a and step 3 is processed recursively.

5. A candidate region of dot-matrix text is identified as the minimum rectangle that encloses the candidate points with the same label. The whole process is repeated three times, setting n in Eq(3) to 8, 16 and 32.

The text candidate regions identified from Fig. 3(c) are shown in Fig. 3(d).

2.3 Preprocessing for character extraction and recognition

Preprocessing for character extraction and recognition involves binarization of identified text regions. The text regions are usually identified as wide rectangles; that is, the width is greater than the height. Because the backgrounds contained in these wide rectangles have large variety of brightnesses, obtaining a suitable binarization result from one fixed threshold value is difficult.

Therefore, the proposed method employs a local thresholding method. The identified text region is divided into k sub-blocks and each sub-block is binarized by a threshold value determined by Otsu's discriminant method.[6] The number of sub-blocks k is determined by Eq.(4).

$$k = \max\left(\text{round}\left(\frac{w}{h}\right), 1\right) \tag{3}$$

where, h and w are the height and width of an identified text region, respectively.

Dot-matrix text sometimes appears on two or more lines. An identified text region is considered a single large rectangle surrounding all of the lines of dot-matrix text. The system must handle these multiple-line texts properly. To find the y-coordinate l dividing two lines, horizontal projection histogram $P = \{p(1), p(2), \ldots, p(h)\}$ is created and l is determined by discriminant analysis method for P. If the mean $m_P = \Sigma_{i=1}^{h} p(i)/h$ and standard deviation $\sigma_P = \sqrt{\Sigma_{i=1}^{h}(p(i) - m_P)^2/h}$ of P satisfy Eq. (5), l is considered the location to divide the text into two lines. If Eq.(5) is not satisfied, the text region is considered a single line of text.

$$p(l) < m_P - \sigma_P \tag{4}$$

Thus, by this preprocessing, text regions containing single-line or two-line text are identified, and the identified text regions are binarized.

2.4 Character extraction and recognition

In the next step, isolated dot-matrix characters are extracted for OCR. Black pixels are dilated 10 times or dilated until doing not exceed extracted l. We extract bounding box of a connected component as a character region. Let $R_i = (x_i, y_i, w_i, h_i)$ denote the i-th character region, where x_i and y_i are the coordinates of the upper left of the region, and h_i and w_i are the height and width of the region, respectively.

If multiple regions overlap by more than 20%, these regions are combined and considered one isolated character.

When w_i is greater than h_i, the method considers R_i to be separated into multiple isolated characters. The number of segmentations to make isolated characters is changed from $s+1$ to $s+5$ and the region is divided so that the width of each character is the same. In this case, divided regions become segmentation candidates and true segmentation is determined in a later step. Value s is floor of (w_i/h_i). The process of dot-matrix character extraction from identified text regions is shown in Fig. 4.

All isolated character regions are evaluated by the MQDF and target characters are classed as A–Z, 0–9 or a colon or a slash. The MQDF is expressed by:

$$\begin{aligned} g(X) &= (N + N_0 + n - 1)\ln[1 + \frac{1}{N_0\sigma^2}[||X - M||^2 - \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i + \frac{N_0}{N}\sigma^2} \left\{\Phi_i^T (X - M)\right\}^2]] \\ &\quad + \sum_{i=1}^{k} \ln\left(\lambda_i + \frac{N_0}{N}\sigma^2\right) - 2\ln P(\omega), \end{aligned} \tag{5}$$

where, X denotes a n-dimensional gradient feature vector of input dot-matrix character, M is mean vector of training samples, λ_i and Φ_i are i-th eigenvalue and corresponding eigenvector of covariance matrix of training samples, respectively. k

(a) Dot-matrix text extracted from whole image (b) Dot-matrix character candidates (c) Dot-matrix characters extracted after evaluation

Figure 4. Dot-matrix character extraction from text.

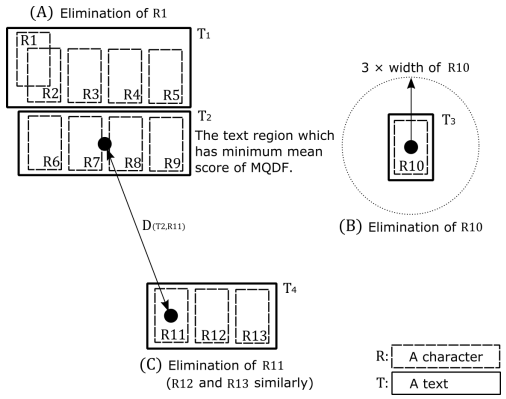


Figure 5. Elimination of false-positives using positional relationships

is a parameter which denotes the number of eigenvectors used for classification. N and $P(\omega)$ denote the number of training data and a priori probability of class ω. σ^2 is variance assuming spherical a priori distribution of X and determined by mean of all eigenvalues in all classes.

To train the MQDF classifier, we generated dot-matrix characters using the variation model based learning.[3] The training characters were generated by introducing blur, missing dots, different sized dots, different array patterns and target object rotation. These characters are used to train the OCR system to recognize a variety of character appearances.

During the recognition process, the MQDF scores are evaluated for all characters. If the region has multiple segmentation candidates, the segmenting position is selected that gives the best mean value of the MQDF score. All identified character regions are shown in Fig. 6(a).

2.5 Elimination of false positives

Several methods are used to evaluate and eliminate false positives. First, if the region is a false positive, the MQDF score increases. The MQDF assigns an input character to the class which provides the minimum MQDF score. Thus, a high MQDF score means that the confidence of classification is low. If any one of the following conditions for one character region are satisfied, the region is eliminated as a false positive.

1. The MQDF score is greater than 600.
2. The mean MQDF score in the text region where the character belongs is greater than 400.

In addition, we eliminate false positives based on the shape of extracted isolated character region.

3. The white area is greater than 90% of the bounding box.
4. The black area is greater than 90% of the bounding box.
5. The vertical size (height) of the character region is less than half the horizontal size (width) of character region.

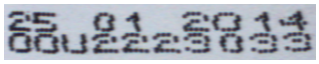
Because dot-matrix characters are typically aligned on a straight line, false positives are also eliminated based on the relative position of the characters. The relative positions of characters are shown in Fig. 5.

(a) All extracted character regions

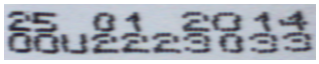
(b) Character regions which have dot-matrix character features

Figure 6. Elimination of candidate regions which do not have dot-matrix characters.

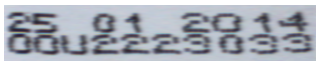
(a) Original text



(b) Text with 5% of pixels affected by smoothing



(c) Text with 10% of pixels affected by smoothing



(d) Text with 15% of pixels affected by smoothing

Figure 7. Dot-matrix text altered by motion blur simulated by smoothing in the horizontal direction.

6. Boxes bounding characters overlap and have lower MQDF score, as shown in (A).
7. No character regions are extracted within the range of three times of the character width as shown in (B).
8. $D_{(R,T)}$ is greater than three times of the character width as shown in (C).

The result of identification after elimination of false positives is shown in Fig. 6(b).

2.6 Detection of printing defects

The proposed method also detects printing defects using the MQDF classification scores for each character class. Fig. 8 shows examples of obtained MQDF scores for a normal dot-matrix character (a) and for a character missing a line (b). The signs of the MQDF values are inverted. While the corresponding class (character class 0) obtains the largest MQDF score in both cases, the score for the corresponding class is smaller for defective characters than that for normal characters.

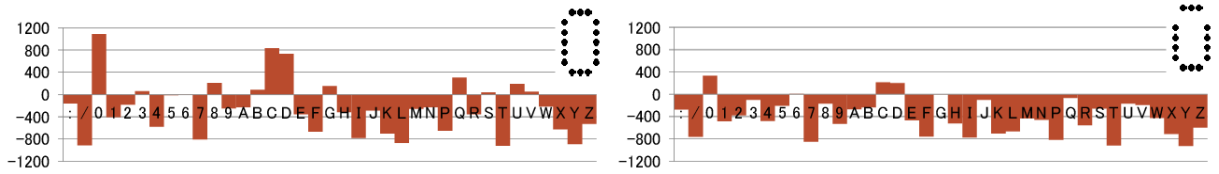
From this observation, we construct a feature vector (score vector) of which elements are inverted MQDF scores for each character class and input the feature vector by SVM to detect printing defects. The SVM is trained with a dataset consisting of normal dot-matrix characters and characters generated with printing defects.

3. EVALUATION EXPERIMENTS

3.1 Dataset

The performance of the proposed OCR method is evaluated using actual product images. The evaluation dataset consists of 250 images captured by a standard camera. The images are 640×480 pixels. Dot-matrix characters are printed on all products in dataset images.

Robustness against motion blur is required because captured images are sometimes affected by the fast motion of a production line. Therefore, we evaluated the proposed method using images containing artificial motion blur. To blur the images, we applied horizontal 1×3 smoothing to the images from the dataset. The magnitude of motion blur is b_m, where m is the percentage (%) of pixels affected by smoothing in the horizontal direction of the dot-matrix characters in the image. Images simulating small, medium and large magnitude motion blur were simulated by smoothing affecting 5%, 10% and 15% of pixels, respectively, as shown in Fig. 7. Robustness against rotation is also required because some text



(a) MQDF scores for the normal dot-matrix character (b) MQDF scores for dot-matrix characters with a missing line

Figure 8. Examples of MQDF scores (vertical axes, displayed by negative of original values) for each character class for (a) a normal dot-matrix character and (b) a dot-matrix character with a printing defect (a missing line). The horizontal axes show the character IDs.

Table 1. Result of experiment with MQDF scores

Image type	Recall	Precision	Accuracy
Original	78.60%	76.03%	94.43%
Small blur	78.20%	72.54%	93.47%
Medium blur	75.27%	70.18%	92.16%
Large blur	68.30%	63.91%	88.04%
15 deg. rot.	80.53%	73.96%	93.64%
30 deg. rot.	77.67%	76.03%	91.22%
45 deg. rot.	80.26%	77.47%	93.82%

Table 2. Result of experiment without MQDF scores

Image type	Recall	Precision	Accuracy
Original	85.12%	26.14%	94.43%
Small blur	83.37%	25.29%	93.64%
Medium blur	82.80%	22.90%	91.22%
Large blur	76.50%	22.52%	88.02%
15 deg. rot.	86.09%	23.44%	93.43%
30 deg. rot.	83.95%	24.56%	90.95%
45 deg. rot.	85.83%	27.18%	92.80%

is captured in a rotated position. Therefore, we evaluated the proposed method using images rotated by 15, 30 and 45 degrees.

To evaluate the performance of printing defect detection, we collected 2600 images of dot-matrix character containing four types of printing defect; missing dot, missing line, bleeding and spot dust.

3.2 Experiment outline

We evaluated recall, precision and F-measure of dot-matrix characters extraction and recognition accuracy for extracted characters. To verify the effectiveness of false positive elimination using MQDF values, results are compared with the results of processing without the elimination process. Accuracy, recall and precision are defined by the following equations,

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}, \quad \text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \quad \text{Accuracy} = \frac{\text{RR}}{\text{TP}},$$

where TP, FN, FP and RR denote the numbers of extracted dot-matrix characters at the true position, dot-matrix characters that are not extracted, extracted non-dot-matrix character regions and correctly recognized characters at the correct position, respectively.

To verify the effectiveness of our proposed OCR method, we compare the extraction results with those from a MSER[7,8]-based character extraction method and from another well-known corner detectors, the eigen-corner detector[9] and the Harris detector.[10]

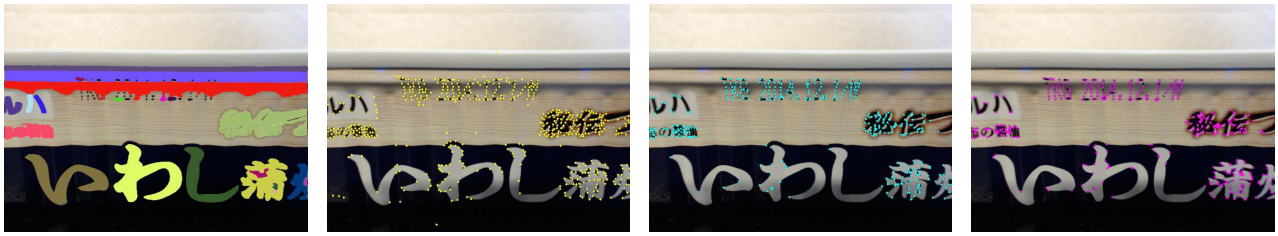
4. RESULTS

Table.1 shows the result of the experiment. For the original images, recall and precision of extraction are 78.60% and 76.03%. The accuracy of the extracted characters is 94.43%.

Fig. 9 and Table.3 show examples of actual corner detection results and quantitative performance by comparing methods. Quantitative evaluation for MSER is omitted because MSER does not work properly for dot-matrix OCR. We can observe that the three corner-detection methods provide similar detection results in Fig. 9; however, the FAST corner detector outperforms other methods in recall, precision and computation time.

Table.2 shows the results of the experiment without the MQDF score. By false-positive elimination using the MQDF score, precision is dramatically improved without worsening recall.

The accuracy of printing defect detection by SVM with RBF-kernel is 96.68%



(a) MSER (b) Eigen-corners (c) Harris (d) FAST

Figure 9. Examples of actual dot-matrix text extraction by (a) MSER, (b) Eigen-corner detector, (c) Harris detector and (d) FAST corners for visual comparison.

Table 3. Quantitative comparison of performance of dot-matrix OCR methods.

Method	Recall(%)	Precision(%)	Time (ms)
Eigen-corner	69.86	69.56	15.44
Harris	77.37	73.45	14.14
FAST	78.60	76.03	6.76

5. CONCLUSION

In this paper, we propose an OCR method for dot-matrix text. Candidate regions of dot-matrix text are identified extracted by clustering using intensity histograms of FAST corner points. Identified text candidates are segmented into character candidates by connected components after dilation and aspect ratio of text region. Characters in extracted text regions are segmented and recognized, and false positives are eliminated. The variation model based learning method, gray-scale gradient features and MQDF classifications are used for OCR. Corner detection and recognition evaluation using the MQDF score are effective for identifying extracting and recognizing dot-matrix characters printed on products in camera-captured images.

The proposed method can be applied to medium motion blurred images but further improvements to robustness are still required. The proposed method is robust against rotation because we have no limited process for rotation. Future development will focus on better model generation and more effective binarization.

REFERENCES

[1] Namane A., Soubari A.H., Meyrueis P., “Degraded dot matrix character recognition using csm-based feature extraction.” Proc. 10th ACM Symp. Doc. Eng., 207-210 (2010)

[2] Du Y., Ai H., Lao S., “Dot text detection based on fast points” Proc. ICDAR2011, 435-439 (2011)

[3] Endo K., Ohyama W., Wakabayashi T., Kimura F., “Performance Improvement of Dot-Matrix Character Recognition by Variation Model Based Learning”, Proc IWRR2014, O2-1 (2014)

[4] Kimura F., Takashina K., Tsuruoka S., Miyake Y., “Modified quadratic discriminant functions and the application to Chinese character recognition” IEEE Trans. PAMI 9, 149-153 (1987)

[5] Rosten E., Drummond T., “Machine leaning for high-speed corner extraction”, Proc. ECCV2006 430-443 (2006)

[6] Otsu N., “A threshold selection method from gray-level histograms” IEEE Trans. SMC, 9 (1) 62-66 (1979)

[7] Yin X.C., Yin X., Huang K., Hao H.W., “Robust text detection in natural scene images”, IEEE Trans. PAMI, 36 (5) 970-983, (2013)

[8] Neumann L., Matas K. “Real-time scene text localization and recognition”, Proc. CVPR2012 (2012)

[9] Shi J., Tomasi C., “Good Features to Track”, Proc. CVPR94, 593-600 (1994)

[10] Harris C., Stephens M., “A Combined Corner and Edge Detector”, Proc. Alvey Vision Conf., 147-151 (1988)