A Method for Text Localization and Recognition in Real-World Images

Lukas Neumann and Jiri Matas

Center for Machine Perception, Czech Technical University in Prague, Czech Republic

Abstract. A general method for text localization and recognition in real-world images is presented. The proposed method is novel, as it (i) departs from a strict feed-forward pipeline and replaces it by a hypotheses-verification framework simultaneously processing multiple text line hypotheses, (ii) uses synthetic fonts to train the algorithm eliminating the need for time-consuming acquisition and labeling of real-world training data and (iii) exploits Maximally Stable Extremal Regions (MSERs) which provides robustness to geometric and illumination conditions.

The performance of the method is evaluated on two standard datasets. On the Char74k dataset, a recognition rate of 72% is achieved, 18% higher than the state-of-the-art. The paper is first to report both text detection and *recognition* results on the standard and rather challenging ICDAR 2003 dataset. The text localization works for number of alphabets and the method is easily adapted to recognition of other scripts, e.g. cyrillics.

1 Introduction

Text localization and recognition in images of real-world scenes has received significant attention in the last decade [1, 2, 3, 4]. In contrast to text recognition in documents, which is satisfactorily addressed by state-of-the-art OCR systems [5], scene text localization and recognition is still an open problem. Factors contributing to the complexity of the problem include: non-uniform background, the need for compensation of perspective effects (for documents, rotation or rotation and scaling is sufficient); real-world texts are often short snippets written in different fonts and languages; text alignment does not follow strict rules of printed documents; many words are proper names which prevents an effective use of a dictionary.

Most published methods for text localization and recognition [1, 6, 7, 8] are based on sequential pipeline processing consisting of three steps - text localization, text segmentation and processing by an OCR for printed documents. In such approaches, the overall success rate of the method is a product of success rates of each stage as there is no possibility to refine decisions made by previous stages.

Some authors have focused on subtasks of the scene text recognition problem, such as text localization [3,9,10,11,4], individual character recognition [12,13] or reading text from segmented areas of images [14]. Whilst they achieved promising

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 770–783, 2011.

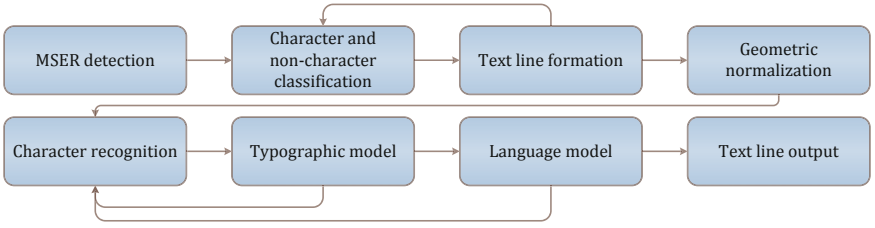


Fig. 1. Stages of the proposed method (incl. feedback loops for hypotheses verification)

results for individual subtasks, separating text localization from text recognition inevitably leads to loss of information, which results in degradation of overall text localization and recognition performance.

In this paper, we propose an end-to-end method for text localization and recognition. The technical contributions of the paper are the following. First, in the recognition part, no real-world training data are used. Learning is carried out directly on characters from fonts available in the Windows OS, with no preprocessing simulating acquisition effects, e.g. blur and deformations. Nevertheless, the proposed method achieves high recognition rates. Application of the method to other scripts, demonstrated on cyrillics in the paper, required only insertion of the relevant font sets (see Figure 2).

Second, characters are assumed to be extremal regions [15] in some scalar projection of pixel values. Character recognition is performed on a representation derived from the boundaries of extremal regions. Such a representation filters out effects of illumination, colour and texture variation in either foreground or background, or both, which is an important property for real-world text recognition (in contrast to printed document recognition, where such effects do not apply). Moreover, overlap of bounding boxes in tightly spaced text (e.g. with kerning) does not effect our method, which is not the case in methods where character detection is based on the sliding window. Extremal regions have been used for character recognition before [16], but in a very specific domain of single-font licence plate recognition rather than in a generic scene text recognition.

The proposed method is also novel in avoiding a pipeline architecture with a sequence of fixed decisions and working with multiple hypotheses at each stage

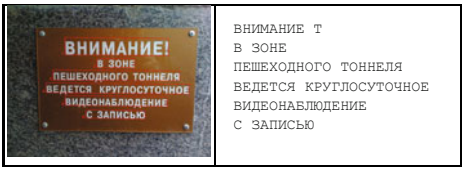


Fig. 2. Text localization and recognition output example on Russian text. Note: The only adjustment of the proposed method was a use of synthetic cyrillic fonts to train the character recognition with a Russian language model. The recognition is error free, with the exception of the exclamation mark which is not included in the training set.

of the processing (text localization, character segmentation, text line formation). Early steps are revisited in a hypothesis-verify framework and the decision about the most probable hypothesis is left to the last module, when values of all hidden parameters have been inferred.

The rest of the document is structured as follows: in Section 2, the problem of text detection and recognition is defined. Section 3 describes the proposed method. Performance evaluation of the proposed method is presented in Section 4. The paper is concluded in Section 5.

2 Problem Description

Let $\mathbf{I}$ be an input image and let $\mathcal{R}$ be a set of all contiguous regions of the image $\mathbf{I}$. Let S_m denote a set of all sequences of regions $S_m = \{(R_1, R_2, \ldots, R_m)\,; R_i \in \mathcal{R}\}$ of length m and let $\mathcal{S}$ denote a set of all sequences of all lengths $\mathcal{S} = \bigcup_{m=1\ldots n} S_m$, where n denotes the number of pixels in the image.

Text localization is defined as finding all sequences $s \in \mathcal{S}$ such that probability that the sequence represents a text $p_s(\text{text})$ has a local maximum, i.e. $\forall a \in \text{Adj}(s) : p_s(\text{text}) > p_a(\text{text})$ and $p_s(\text{text})$ is above a predefined threshold θ, where $\text{Adj}(s)$ denotes all sequences adjacent to sequence s. Two sequences are considered adjacent, if the first one differs from the second one by adding a single region at the end of the sequence. We assume that the probability $p_s(\text{text})$ is known from ground truth of training data.

Text recognition, given an alphabet $\mathcal{A}$, assigns a sequence of characters $\mathbf{y} = y_1 y_2 \ldots y_l : y_i \in \mathcal{A}$ to each sequence of regions s. Note that the length of the sequence of characters $\mathbf{y}$ may differ from the length of the sequence of regions s.

The problem of text localization and detection can be also described using notions of graph theory, which is more convenient for description of our method. Let $\mathbf{G}$ denote an undirected graph with vertices $V(\mathbf{G}) = \mathcal{R}$ and edges $E(\mathbf{G}) = \{(R_i, R_j) \in \mathcal{R} \times \mathcal{R} \mid i \neq j\}$. Each sequence $s \in \mathcal{S}$ of regions of length m is represented by a path $p = (v_1, v_2, \ldots v_n)\,; v_i \in V(\mathbf{G})$ in the graph $\mathbf{G}$ of the same length. The set of all sequences $\mathcal{S}$ then corresponds to the set of all paths $\mathcal{P}$ in the graph $\mathbf{G}$.

Because each path p has a one-to-one relation to a sequence s, the probability of a path p being a text equals to the probability of the corresponding sequence $p_s(\text{text})$. Let $h(p, v);\ p \in \mathcal{P}, v \in V(\mathbf{G})$ denote an auxiliary function such that

$$h(p, v) = \begin{cases} 1 & p_{pv}(\text{text}) > p_p(\text{text}) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where pv denotes a path which was created by extending the path p with a vertex v.

Text localization can be then equally formulated as finding all paths $p \in \mathbf{P}$ such that $\forall v \in V(\mathbf{G}) : h(p, v) = 0$ and $|p| > l_{min}$, where l_{min} denotes a predefined threshold for minimal text length. In other words, text localization is a search for all paths in the graph $\mathbf{G}$ longer than l_{min}, such that extending the path by any other vertex decreases the probability of the path being a text.

3 Text Localization and Recognition

3.1 MSER Detection

Since the original search space induced by all regions $\mathcal{R}$ of image $\mathbf{I}$ is huge, certain approximations were applied in our approach. Assuming that individual characters are detected as Extremal Regions (ER) and taking computation complexity into consideration, the search space was limited to the set $\mathcal{M}$ of Maximally Stable Extremal Regions (MSER) [15], which can be computed in linear time in number of pixels [17].

The set of MSERs detected in certain scalar image projections (intensity, red channel, blue channel, green channel) defines the set of vertices of the graph $\mathbf{G}$, i.e. $V(\mathbf{G}) = \mathcal{M}$. The edges of the graph $\mathbf{G}$ are not stored explicitly, but they are induced on the fly (see Section 3.3).

3.2 Character and Non-character Classification

In this module, each vertex of graph $\mathbf{G}$ is labeled as a character or a non-character using a trained classifier which creates an initial hypothesis of text position, because character vertices are likely to be included in some path p representing a text.

The features used by the classifier (see Table 1) are scale invariant to detect all characters sizes, but they are not rotation invariant, which implies that characters at different rotations had to be included in the training set.

Once text lines are hypothesized (see Section 3.3), the initial character/non-character classifications are reassessed, taking hidden parameters of the text lines (character height, character spacing, etc.) into account. Thanks to the feed-back loop, the initial classification error has minimal impact on the overall performance.

A standard *Support Vector Machine* (SVM) [18] classifier with Radial Basis Function (RBF) kernel [19] was used. The classifier was trained on a set of 1227

Fig. 3. Character/non-character MSER classifier: (a) Character and (b) non-character training samples (MSER boundaries marked yellow). Initial MSER classification for (c) Armenian, (d) Russian and (e) Kannada script (character regions marked green, non-character regions marked red). Note: The training set contains only 1227 character samples of Latin script and 1396 non-character samples.

characters and 1396 non-characters obtained by manually annotating MSERs extracted from real-world images downloaded from Flickr. The classification error obtained by cross-validation was 5.6%. The training set is relatively small and certainly does not contain all possible fonts, scripts or even characters, but extending the training set with more examples did not bring any significant improvement in the classification success rate. This indicates that features used by the character classifier are insensitive to fonts and alphabets.

Table 1. Features used by the character classifier

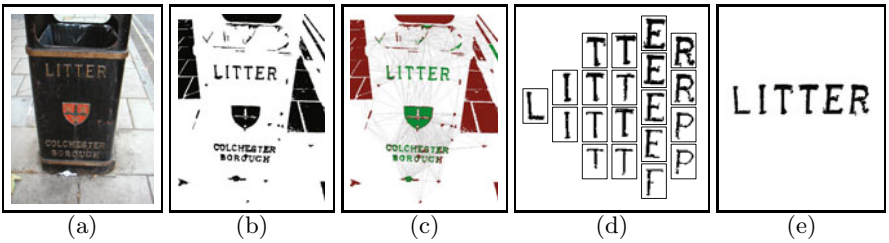
aspect ratio	relative segment height
compactness	number of holes
convex hull area to surface ratio	character color consistency
background color consistency	skeleton length to perimeter ratio

3.3 Text Line Hypothesis Formation

In real-world images a font rarely changes inside a word, which implies that certain character measurements (character height, aspect ratio, spacing between characters, stroke width, etc.) are either constant or constrained to a limited interval. Based on this observation, an approximation $\hat{h}(p, v)$ of function $h(p, v)$ (see Section 2) was implemented using a SVM classifier with polynomial kernel, whose feature vector is created by comparing average character measurements of the existing path p to the character measurements of given vertex v (see Table 2). The classifier was trained on the ICDAR 2003 Train set [20].

In our approach, only horizontal text areas which form a text line were considered. We think of a horizontal text line as a linear sequence of characters with straight or slightly curved bottom line, whose angle in the picture is in the range of ± 30 degrees.

Each path p is built in the following manner: The top-left unprocessed character vertex in the image is selected, creating an initial hypothesis of path p. The path p is then sequentially extended from left to right by all vertices $v \in V(\mathbf{G})$



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

Fig. 4. Text line hypothesis formation: (a) The source image. (b) MSERs detected in the red channel projection. (c) The induced graph (character vertices marked green, non-character marked red; edges longer than 300px omitted in the image for better readability) (d) Text line content hypotheses. (e) The selected hypothesis.

Table 2. Measurements used by the classifier in the approximation $\hat{h}(p, v)$

character width	character height
character surface	character color
aspect ratio	vertical distance from bottom line
stroke width	MSER margin [15]

such that $\hat{h}(p, v) = 1$ and distance of the vertex v in the source image is below the threshold d_{max}, which value was set experimentally to $3w_{max}$, where w_{max} denotes maximal character width in the existing path p.

If more than one vertex can be added to the path, multiple hypotheses about the path p are created and the decision about the most probable path is postponed for a later stage. If the path cannot be extended, all vertices of the path are marked as processed and next unprocessed top-left character vertex is selected to initialize a hypothesis of another independent path.

Every time a path p is extended by a new vertex, a bottom line approximation is calculated by Least-Median Squares (LMS) fitting of bottom points of individual regions in the text line; the approximation is then used to calculate the vertical distance of a vertex in the $\hat{h}(P, M)$ function. If the path is shorter than 5 vertices, only straight bottom line is allowed; if the path is longer, the bottom line is allowed to be slightly curved by fitting a parabola (see Figure 11, bottom-left).

3.4 Geometric Normalization

Perspective distortion is rectified prior to character recognition as all characters are trained in the frontoparallel view. The orientation of a camera to a plane with text in 3D space is modelled as a homography with a transformation matrix $\mathbf{H}$, which is decomposed as

$$\mathbf{H} = \begin{pmatrix} s\cos\theta & s\sin\theta & t_x \\ -s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/b & -\sigma/b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \ell_x & \ell_y & 1 \end{pmatrix} \tag{2}$$

The transformation has 8 degrees of freedom. However, only 3 of them are important for character recognition: the perspective foreshortening parameters ℓ_x, ℓ_y and the shear σ. Rotation θ can be easily calculated from the text line approximation and the scale parameters s and b, as well as the translation parameters t_x, t_y are not important thanks to the normalization, which is applied before the character recognition.

The sought parameters are estimated by the method of Myers et al. [21]. In this method, the perspective foreshortening parameters ℓ_x, ℓ_y are calculated from the horizontal vanishing point $\mathbf{V}_H$, which is located by finding top and bottom line of the text block and calculating its intersection.

Following Myers, the text block is rotated in the range of ± 3 degrees by 0.2 degree increments from detected text line orientation in order to find the top line.

For each rotation, the peak value of number of column top-most pixels in each row of the text block bitmap is calculated and the top points in the rotation with highest peak value are then considered a top line. The same process is repeated for the bottom line, here the number of column bottom-most pixels is calculated.

The shear σ is found by first rotating the text block so that the bottom line is horizontal and then iteratively applying a shear transformation in range of -45 to 45 degrees and measuring sum of squares of count of pixels in each column. The shear with the highest value is taken as a result.

(a) (b) (c)

Fig. 5. Geometrical normalization. (a) Text area in source image with detected top and bottom line. (b) Normalization input. (c) Normalization result.

3.5 Character Recognition

The character recognition starts by normalizing the MSER to a fixed-sized matrix of 35×35 pixel, while retaining the centroid of the region and aspect ratio [22]. Next, boundary pixels are inserted into separate bitmaps according to their orientation. After Gaussian blurring each bitmap is sub-sampled to a matrix of 5×5 pixels to generate 25 features. In total, 25 features $\times$ 8 directions generate 200 features for each MSER mask.

The 200-dimensional feature vectors are classified by a SVM classifier with Radial Basis Function (RBF) kernel. Based on the assumption that fonts in real-world images are very similar to standard synthetic fonts, the set of 40 synthetic fonts which are installed as part of Microsoft Windows OS was used to train the classifier using one-against-one strategy.

If the width of the region is bigger than threshold c_{min}, which was experimentally set to $c_{min} = 1.5w_{max}$, it is possible that the region actually corresponds to more than one character and thus an attempt to split the region is made. Candidate points for splitting are detected as the local minimum of the distance between the top and bottom pixels in a column (see Figure 8). Each combination of splitting is then evaluated in the context of surrounding letters using a feedback loop (see Section 3.7) and the hypothesis with the highest score according to the language model is selected.

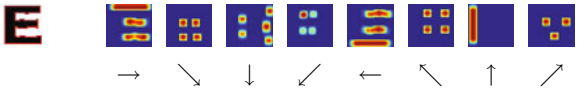
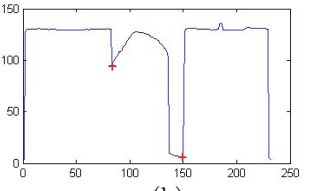


Fig. 6. Character recognition features: Input character (left). Features of the chain-code bitmap for each direction (right).

Fig. 7. Synthetic training samples of the character classifier. No "real world" training samples were used.

(a)



(b)

Split regions	Recognized text
	ItI
	Iu
	RI
	N

(c)

Fig. 8. Region splitting. (a) Source region. (b) Region column heights. (c) Resulting hypothesis.

3.6 The Typographic Model

A feed-back loop for character recognition was introduced as it is virtually impossible for the character classifier (see Section 3.5) to correctly differentiate between upper-case and lower-case variant of certain letters (such as "C" and "c") without knowing the heights of other letters in the text line. In order to correctly recognize the interchangeable letters, the height of unambiguously recognizable big and small letters is measured and then compared to actual height of the classified letter (see Figure 9).

Horizontal spacing between individual characters is measured and spaces between words are inserted at appropriate positions using a heuristics based on the analysis of the histogram of text line spacings.

Big	ABbDdEFfGHhKkLMNQRTtY012345678
Small	aegmnqry
Interchangeable	CcIiJjlOoPpSsUuVvWwXxZz

Fig. 9. The typographic model. Letter categories (left). Text line measurements (right) - (1) big and (2) small letters height, (3) base-line. Interchangeable letters marked gray.

3.7 The Language Model

The method treats each text line hypothesis individually, but in reality some of the hypotheses are mutually exclusive, either because their corresponding paths P in graph $\mathbf{G}$ have to be disjoint (one region can only be present in one text line) or due to their actual position in the image (a given area in an image can contain only one text line).

Given an alphabet $\mathcal{A}$, word $w = a_1 a_2 a_3 \dots a_n, a_i \in \mathcal{A}$ and a set of words in a dictionary $\mathcal{W}$ a word score $s(w)$ is defined

$$s(w) = \begin{cases} 1 & w \in \mathcal{W} \\ \sqrt[n]{\prod_{i=1}^{n-1} P(a_i, a_{i+1})} & w \notin \mathcal{W} \end{cases} \tag{3}$$

The probability $P(r, s)$ is estimated using relative frequency of the sequence in the dictionary $\mathcal{W}$.

Given a text line $t = w_1, w_2, \dots w_n$, the text line score $S(t)$ is then defined

$$S(t) = \sqrt[n]{\prod_{i=1}^{n} s(w_i)} \tag{4}$$

Given a set $\mathcal{T}$ of mutually exclusive hypotheses, the hypothesis with the highest score $S(t);\ t \in \mathcal{T}$ is selected.

Table 3. The set of mutually exclusive hypotheses and their score $S(t)$ in the English language model. The selected hypothesis is in bold.

Text line hypothesis	Recognized text	Score
LITTEP	LITTEP	0.0528
LITTER	LITTFR	0.0356
LITTER	LITTER	**0.0814**
LITTFP	LITTFP	0.0168

4 Experiments

4.1 Chars74K Dataset

The performance of the proposed method was evaluated on the *Chars74K*[1] dataset using the protocol proposed in the method of de Campos et. al [12]. In total, the *GoodImg* dataset used by the method of de Campos et. al contains 636 images with 7705 annotated characters of Latin alphabet. The SVM classifiers used in the method and their parameters were trained on an independent training set. The language model was created using a dictionary of approx. 10000 most frequent English words.

For each character in the ground truth, one of the three situations can occur: a letter is localized and recognized correctly (*matched*), a letter is localized correctly but not recognized correctly (*mismatched*) or a letter is not localized at all (*not found*). Since the dataset does not contain full annotations for words, it is not possible to obtain word recognition statistics.

[1] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

Fig. 10. Text localization and recognition examples on the Chars74K dataset. Kannada letters output marked red.

Table 4. Individual character recognition results on Chars74K dataset

	matched	mismatched	not found
proposed method	**71.6%**	**12.1%**	**16.3%**
de Campos et al.	54.3%	45.7%	N/A

The results show that the proposed method outperforms the results of de Campos et al. [12], where the best result achieved on the English *GoodImg* dataset is 54.30% correctly recognized letters. Note that the method of de Campos et al. works with manually located letters and thus there is no need for text localization. In our method, characters are detected automatically and the failure of detection is 16.3%, more than half of the total error rate of 28.4% (see Table 4).

Kannada letters in the Chars74k dataset were also successfully localized, but since the character classifier was not trained to support Kannada alphabet, the method outputs random strings for such texts (see Figure 10); since the method was evaluated only on English ground truth, the detected Kannada letters did not have any impact on the results.

4.2 ICDAR 2003 Dataset

The proposed method was also evaluated on ICDAR 2003 Robust Reading Competition Test dataset[2], which contains 5370 letters and 1106 words in 249 pictures. The same parameter setting as in the previous experiment (see Section 4.1) was used. The dictionaries supplied with the ICDAR 2003 dataset were not

[2] http://algoval.essex.ac.uk/icdar/Datasets.html

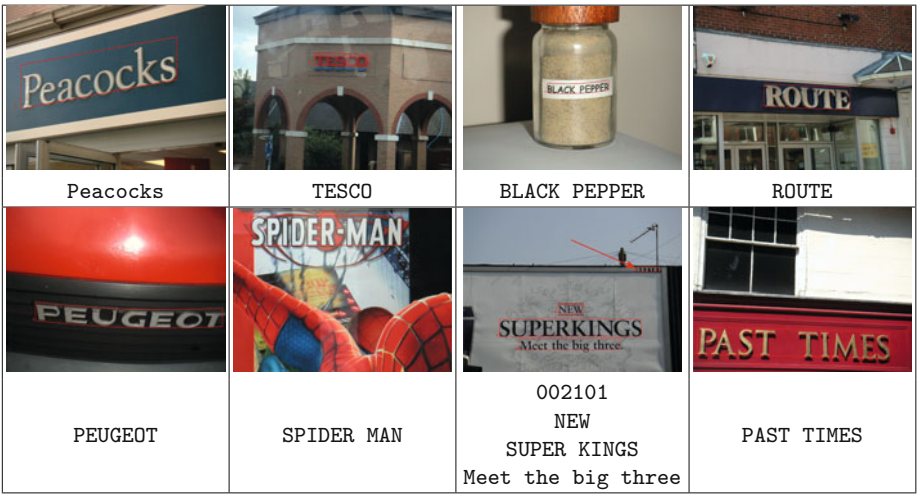


Fig. 11. Text localization and recognition examples on the ICDAR 2003 dataset

(a) (b) (c)

Fig. 12. Problems of the ICDAR 2003 ground truth. (a-b) Text detected by the proposed method but missed by the annotator (marked with a red arrow). (c) Interpretation as text is controversial (the cross is marked as "X" in the ground truth).

Fig. 13. Examples of ICDAR 2003 images where the proposed method fails to localize the text

Table 5. Results on the ICDAR 2003 dataset

(a) Text localization

method	precision	recall	f
Pen et. al [11]	0.67	0.71	0.69
Epshtein et. al [4]	0.73	0.60	0.66
Hinnerk Becker [23]	0.62	0.67	0.62
Alex Chen [23]	0.60	0.60	0.58
proposed method	**0.59**	**0.55**	**0.57**
Ashida [20]	0.55	0.46	0.50
HWDavid [20]	0.44	0.46	0.45
Wolf [20]	0.30	0.44	0.35
Qiang Zhu [23]	0.33	0.40	0.33
Jisoo Kim [23]	0.22	0.28	0.22
Nobuo Ezaki [23]	0.18	0.36	0.22
Todoran [20]	0.19	0.18	0.18

(b) Robust reading

method	precision	recall	f	t
proposed method	0.42	0.39	0.40	89s

(c) Individual character recognition (total numbers in parentheses)

	matched	mismatched	not found
proposed method	67.0% (3598)	12.9% (695)	20.1% (1077)

used in order to evaluate generic performance of the method. The standard definitions of word precision and recall defined in ICDAR 2003 Text Locating and Robust Reading competitions were used [20].

The results show that in terms of text localization, the proposed method achieves worse results than the winner algorithm of ICDAR 2005 [23] or the method proposed by Pen et al. [11], but is still competitive. In text recognition evaluation, we are not able to compare the proposed method with any existing method because there were no entries for ICDAR 2003/2005 Robust Reading competitions. We are not aware of any method with results on the complete ICDAR 2003 dataset.

5 Conclusions

An end-to-end method for scene text localization and recognition was proposed. The proposed method introduces a number of novel features, mainly: a departure from a strict feed-forward pipeline that is replaced by a hypotheses-verification framework simultaneously processing multiple text line hypotheses; the use of synthetic fonts to train the algorithm eliminating the need for time-consuming acquisition and labeling of real-world training data and the use of MSERs which provides robustness to geometric and illumination conditions.

The performance of the method was evaluated on two standard datasets. On the de Campos et al. Char74k dataset [12], a highly significant increase in recognition rate from 53% [12] to 72% was achieved. The text recognition results on the ICDAR 2003 dataset ($f = 0.40$, 67.0% correctly recognized letters) establishes a new baseline as no results in Robust Reading on a complete ICDAR 2003 dataset have been published.

The text localization results on the ICDAR 2003 dataset ($f = 0.57$) are worse than the method proposed by Pen et al. [11] ($f = 0.69$). Most frequent

problems of the proposed method in text localization are individual letters not being detected as MSERs in the projections used, invalid text line formation or invalid word breaking. However, the result has to be interpreted carefully as we noticed that there are problems with the ICDAR 2003 evaluation protocol, e.g. not all text in the image is marked as such and vice versa (see Figure 12).

Acknowledgement. The authors were supported by EC project FP7-ICT-247022 MASH, by Czech Government reseach program MSM6840770038 and by Grant Agency of the CTU Prague project SGS10/069/OHK3/1T/13.

References

1. Wu, V., Manmatha, R., Riseman Sr., E.M.: Textfinder: An automatic system to detect and recognize text in images. IEEE Trans. Pattern Anal. Mach. Intell. (1999)
2. Chen, X., Yang, J., Zhang, J., Waibel, A.: Automatic Detection and Recognition of Signs From Natural Scenes. IEEE Trans. on Image Processing 13, 87–99 (2004)
3. Ezaki, N.: Text detection from natural scene images: towards a system for visually impaired persons. In: Int. Conf. on Pattern Recognition, pp. 683–686 (2004)
4. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: CVPR 2010: Proc. of the 2010 Conference on Computer Vision and Pattern Recognition (2010)
5. Lin, X.: Reliable OCR solution for digital content re-mastering. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series (2001)
6. Chen, X., Yuille, A.L.: Detecting and reading text in natural scenes. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 366–373 (2004)
7. Gao, J., Yang, J.: An adaptive algorithm for text detection from natural scenes. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, p. 84 (2001)
8. Jain, A.K., Yu, B.: Automatic text location in images and video frames. In: International Conference on Pattern Recognition, vol. 2, p. 1497 (1998)
9. Pan, Y.F., Hou, X., Liu, C.L.: A robust system to detect and localize texts in natural scene images. In: IAPR International Workshop on Document Analysis Systems, pp. 35–42 (2008)
10. Kim, E., Lee, S., Kim, J.: Scene text extraction using focus of mobile camera. In: International Conference on Document Analysis and Recognition, pp. 166–170 (2009)
11. Pan, Y.F., Hou, X., Liu, C.L.: Text localization in natural scene images based on conditional random field. In: ICDAR 2009: Proc. of the 2009 10th International Conference on Document Analysis and Recognition, pp. 6–10 (2009)
12. de Campos, T.E., Babu, B.R., Varma, M.: Character recognition in natural images. In: VISAPP, February 05-08 (2009)
13. Yokobayashi, M., Wakahara, T.: Segmentation and recognition of characters in scene images using selective binarization in color space and gat correlation. In: Proc. of the 8th International Conference on Document Analysis and Recognition, pp. 167–171 (2005)
14. Weinman, J.J., Learned-Miller, E., Hanson, A.R.: Scene text recognition using similarity and a lexicon with sparse belief propagation. IEEE Trans. Pattern Anal. Mach. Intell. 31, 1733–1746 (2009)

15. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22, 761–767 (2004)
16. Matas, J(G.), Zimmermann, K.: A new class of learnable detectors for categorisation. In: Kalviainen, H., Parkkinen, J., Kaarna, A. (eds.) SCIA 2005. LNCS, vol. 3540, pp. 541–550. Springer, Heidelberg (2005)
17. Nistér, D., Stewénius, H.: Linear time maximally stable extremal regions. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 183–196. Springer, Heidelberg (2008)
18. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
19. Muller, K.R., Mika, S., Ratsch, G., Tsuda, K., Scholkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. on Neural Networks 12, 181–201 (2001)
20. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: Icdar 2003 robust reading competitions. In: ICDAR 2003: Proc. of the 7th International Conference on Document Analysis and Recognition, p. 682 (2003)
21. Myers, G.K., Bolles, R.C., Luong, Q.T., Herson, J.A., Aradhye, H.: Rectification and recognition of text in 3-d scenes. IJDAR 7, 147–158 (2005)
22. Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognition 37, 265–279 (2004)
23. Lucas, S.M.: Text locating competition results. In: International Conference on Document Analysis and Recognition, pp. 80–85 (2005)